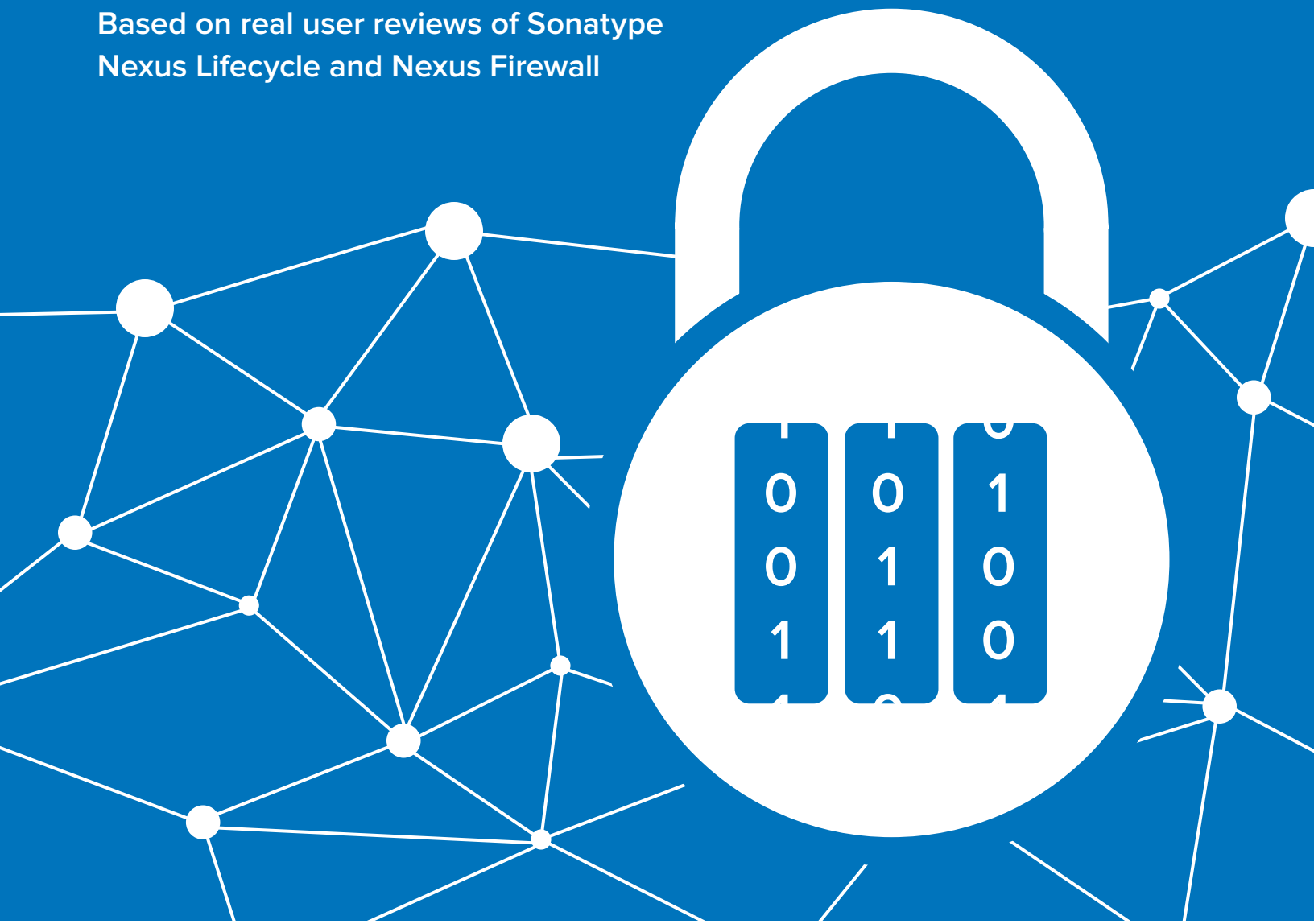
PeerPaper™ Report 2021

Connecting the Dots with Developers and Software Composition Analysis (SCA)

Based on real user reviews of Sonatype Nexus Lifecycle and Nexus Firewall



IT Central Station
Unbiased reviews from the tech community

InfoWorld
FROM IDG

ABSTRACT

Software Composition Analysis (SCA) tools are typically seen as security solutions, but this view ignores the needs of the developers who build the software itself. SCA solutions are addressing this issue by adapting to developers' needs. This paper looks at what it takes to make SCA work for developers. Based on real user experiences with Sonatype Nexus Lifecycle and Nexus Firewall on IT Central Station, it explores how next-generation SCA solutions enable greater developer productivity.

CONTENTS

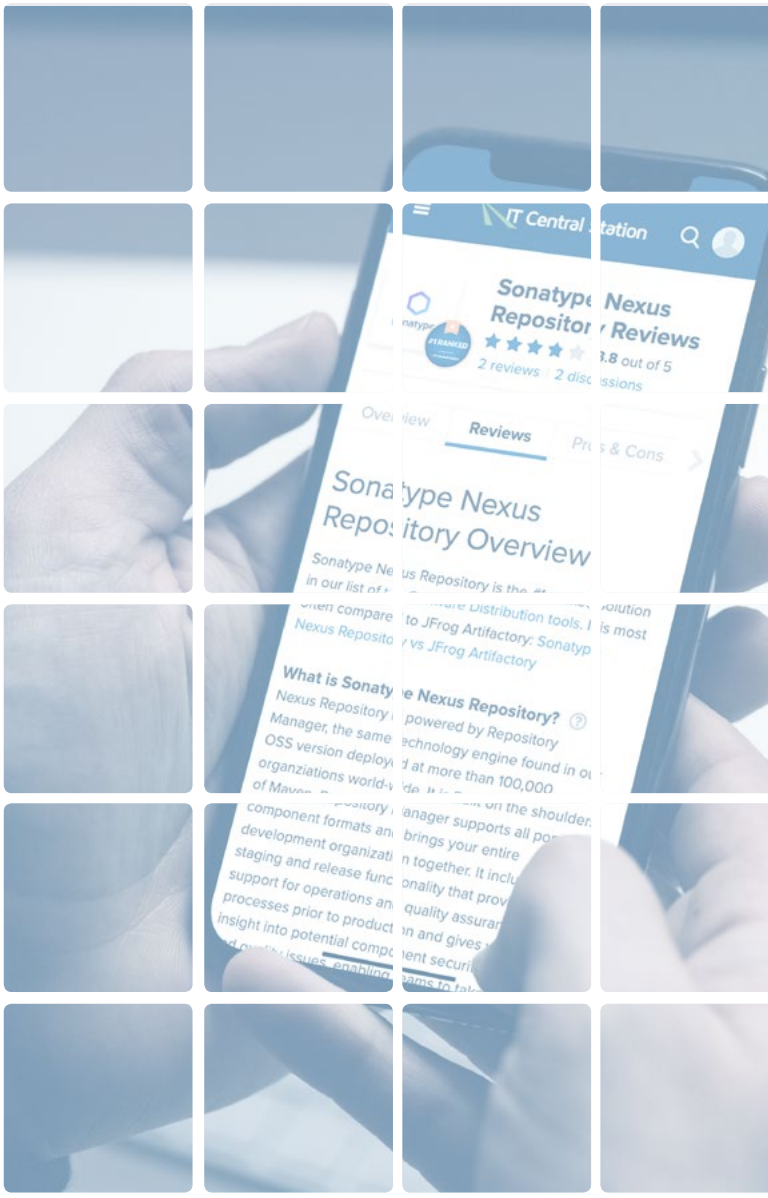
INTRODUCTION

It's long been the norm to view Software Composition Analysis (SCA) tools as security solutions. This makes sense, because SCA's intent is to avoid introducing risk into software by way of open source components. But, what about the developers who are building the software itself? To close this gap, SCA solutions are beginning to adapt to developers' needs.

This paper takes on the challenge of connecting the dots between SCA and developers. It is based on real user experiences with Sonatype Nexus Lifecycle and Nexus Firewall, as described in reviews on IT Central Station. It probes the ways SCA tools can drive improvements in developer productivity, along with greater developer inclusion in the SCA process. Drivers of better developer outcomes include seamless integration with developer tooling, improved data accuracy, and a low rate of false positives. A policy engine helps to ensure that developers use only the highest quality open source components.

SCA and Developers

Software developers face a host of conflicting pressures in their work. Told to keep things moving quickly, developers are also tasked with fixing security issues in the code. Open source vulnerability issues drop even more tasks into their “to do” lists. This is often a frustrating situation for many developers who may think, “Isn’t open source supposed to speed things up? After all, it eliminates the need to code from scratch. Now, it’s slowing things down? That doesn’t seem right.” The optimal practice is to include developers in the remediation of open source licensing and security issues in a way that keeps pace with their work. Intuitive and developer-friendly SCA tools enable a sensible approach to this conflict.



Making SCA Work for Developers

It is possible to make SCA developer-focused. One solution that improves productivity is to integrate SCA with commonly used development environments and DevOps tools. Functions that impede progress can prevent engagement or even create active resistance, such as over-flagging software as unverified (false positives) or not keeping up to date with licenses.

Because developers are often tasked with remediating vulnerabilities and licensing problems found by SCA solutions, SCA tools should block undesirable components and enable effective policy management from the start. Figure 1 below illustrates the developer's perspective, showing how SCA fits into the Software Development Lifecycle (SDLC).

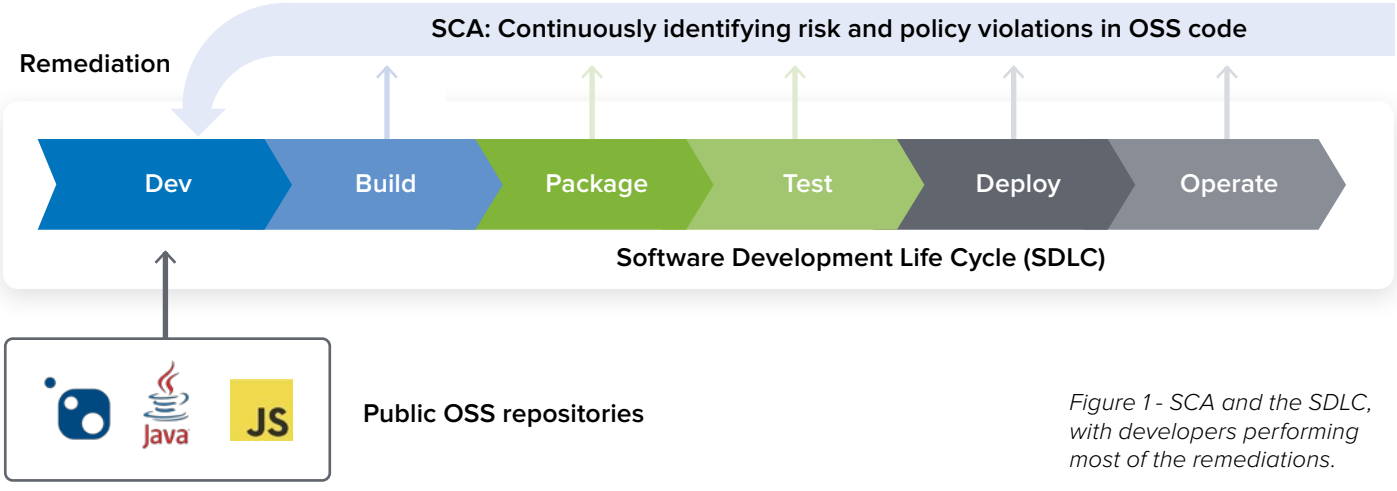


Figure 1 - SCA and the SDLC, with developers performing most of the remediations.

Increasing Developer Productivity

Businesses are seeing ongoing requirements by their customers for more tools and new features in software. Without effective development strategies to help keep up, there is an inevitable competitive price to pay. Furthermore, studies show that developers prefer environments where they're more productive. As such, it's essential to address SCA concerns while not interfering with progress. A DevSecOps Manager at a large financial services firm spoke to this need:

"[Nexus Lifecycle] has helped developer productivity. It's like working in the dark and all of a sudden you've got visibility. You can see exactly what you're using and you have suggestions so that, if you can't use something, you've got alternatives. That is huge."

> **The solution has improved the time it takes us to release secure apps to market by at least 50 percent.**

A Java Development Manager at a large government agency similarly explained how Sonatype Nexus Lifecycle drives better developer productivity: "The solution has improved the time it takes us to release secure apps to market by at least 50 percent. It has also increased developer productivity to some extent because of the IDE [Integrated Development Environment] plugin." He also noted how Nexus gives his developers a vulnerability report, meaning they no longer have to track down the right open source versions to use. He added, "We saw a 10% gain in developer productivity with this capability."

A Software Architect for a small tech vendor sees efficiencies coming from catching security vulnerabilities before a code library is implemented:

"Busy developers will usually prefer to spend the majority of their time implementing features and fixing bugs to meet customer timelines, rather than indefinitely researching possible vulnerabilities in a library they want to use."

"Because it is easy to have this information available (via Nexus Lifecycle), it saves us the hassle of having to refactor later."

A Solutions Delivery Lead at a small financial services firm added:

"The solution has increased developer productivity when remediating issues, as the issues are clearly laid out. We are saving 5 to 10 percent in developer productivity. Nexus [Lifecycle] has improved the time it takes us to release secure apps to market by saving us weeks of rework."

Integrating with DevOps Tooling and More

Getting better with SCA means integrating the SCA tools with DevOps (merging developers with IT Operations) and the IDEs that power that process. IT Central Station members acknowledge the advantage of such integration, as an Enterprise Infrastructure Architect at a small tech services company (Qrypt) explained:

> **It's very seamless for our users. They don't even have to think about it until they have a violation.**

"The Nexus Lifecycle plugin for Azure DevOps allows us to just include the [Nexus Lifecycle]

scan as part of the pipeline deployment. It's very seamless for our users. They don't even have to think about it until they have a violation. [Lifecycle] informs them or stops the build, and the developers have to resolve it."

For a Software Architect at a small tech vendor, "these live updates are a huge improvement to what we were using before." In contrast to their previous solution, Palamida, the Nexus Lifecycle integrates well with their ecosystem. Nexus, running on Amazon Web Services (AWS), connects to Sonatype's service for updates. An Architect at SV Informatik GmbH, a small IT services provider, also compared Nexus Lifecycle to another competitor:

"We also evaluated Black Duck. We selected the Nexus platform because of the data quality and the ability to integrate it into our build process."

A Senior DevOps Engineer who uses Nexus Lifecycle at Primerica, a mid-sized insurance company, also strongly recommends it, saying: "get it implemented into your environment as quickly as you can because it's going to help. Once you get it, get your devs on it because they're going to thank you for it."

Delivering Data Accuracy and a Low Rate of False Positives

Solutions that classify too many programs as suspicious distract developers and may even affect morale. Serious SCA solutions that wish to collaborate with developers need to secure the process, while also minimizing false positives via accurate, up-to-date information. As a security Team Lead at Tyro Payments Limited, a small fintech firm, observed: "while the other products were flagging stuff too, they were flagging things that were incorrect. Nexus has low false-positive results, which give us a high confidence factor. [That] helped us roll out our security policies across the development cycle and ensure that our deployments to production are as secure as possible."

This capability helps avoid critical vulnerabilities from being exposed onsite.

"It saves us time in any remediation activities that we may have had after deployment. If we had discovered security issues after the application was completely developed and deployed, it would be more difficult to go back and make changes or put it back into a cycle."

> "[That] helped us roll out our security policies across the development cycle and ensure that our deployments to production are as secure as possible.

A VP and Senior Manager who uses Nexus Lifecycle at a midsized financial services firm explained:

"The data quality is really good. They've got some of the best in the industry as far as that is concerned. As a result, it helps us to resolve problems faster. The visibility of the data, as well as their features that allow us to query and search – and even use it in the development IDE – allows us to remediate and find things faster."

Similarly, a Senior Enterprise Architect who uses Nexus Lifecycle at the MIB Group, a small insurance company, noted:

"The data quality helps us solve problems faster. Previously, we wouldn't have seen that vulnerability without a painstaking process."

Staying on Top of License Information

Software license compliance is an ongoing, serious problem for developers. To make an SCA solution fit with developer needs, strong license tracking capabilities are key.

A small tech vendor's Engineering Manager put it this way:

"Nexus [Lifecycle] helped us a lot with the management of our OSS licenses and with our knowledge of the licenses and vulnerabilities. It also helped us with knowledge of the libraries that are embedded in our products, and to build a software bill of materials for our projects."

A Configuration Manager at a large wellness and fitness company concurred:

"One of the ways it has improved the way our organization functions is that it created awareness of unlicensed, third-party dependencies and insecure vulnerabilities. You can click on a certain vulnerability and it will give you a recommendation. For example, if you're using something that's not licensed or has a certain license type, it will recommend to you, 'you should go onto this license,' or, 'go to this version, which covers this vulnerability.'"

> "...it created awareness of unlicensed, third-party dependencies and insecure vulnerabilities.

More feedback about Nexus Lifecycle licensing capabilities:

- "The product team has seen some return on investment because they have avoided some vulnerabilities thanks to Nexus [Lifecycle]. They have avoided legal problems around the licenses that are embedded in our products by raising policy violations during scans." - Engineering Manager at a large tech vendor
- "With our leaders across our different organizations, we set policies that govern what types of libraries can be used and what types of licenses can be used. We input those as settings in the tool and [then] the tool manages that throughout the lifecycle, automatically." - VP and Senior Manager at a midsized financial services firm
- "Every time a new build is created in our Continuous Integration (CI) server, Nexus [Lifecycle] will check exactly what libraries we're using. It does this for our Java libraries, JavaScript, and other things that it finds. Then, it checks a number of things for each of those libraries. For example, it checks the license that is being used. Sometimes with open source software, the license is a bit more restrictive than might be convenient for what you are doing." - Software Architect at a small tech vendor

Blocking Undesirable Components

The best approach to SCA is to avoid installing the wrong module in the first place. This is not always possible, but it's definitely helpful when the capability is available. A Senior Cyber Security Architect and Engineer at a large software company explained how this works in their environment:

"At the moment, any developer who needs to download anything from the open source world must do so through Sonatype. All other access is

blocked on the servers themselves. The servers cannot directly go through to PyPI, for example."

Figure 2 below depicts this process in operation.

"The solution blocks undesirable open source components from entering our development lifecycle," said a government agency Java Development Manager.

Nexus Firewall also prevents undesirable open source components from entering the SDLC at Qrypt. As their Enterprise Infrastructure Architect explained:

"We've agreed on the governance of our policies for blocking builds automatically."

Blocking undesirable components can mean forcing contributors to the SDLC to only use the proper and allowed libraries at the correct time in the development lifecycle. This is the approach taken by a VP and Senior Manager who uses Nexus Lifecycle at a mid-sized financial services firm. He said:

"The solution blocks undesirable open source components from entering our development lifecycle. That's its whole point and it does it very well."

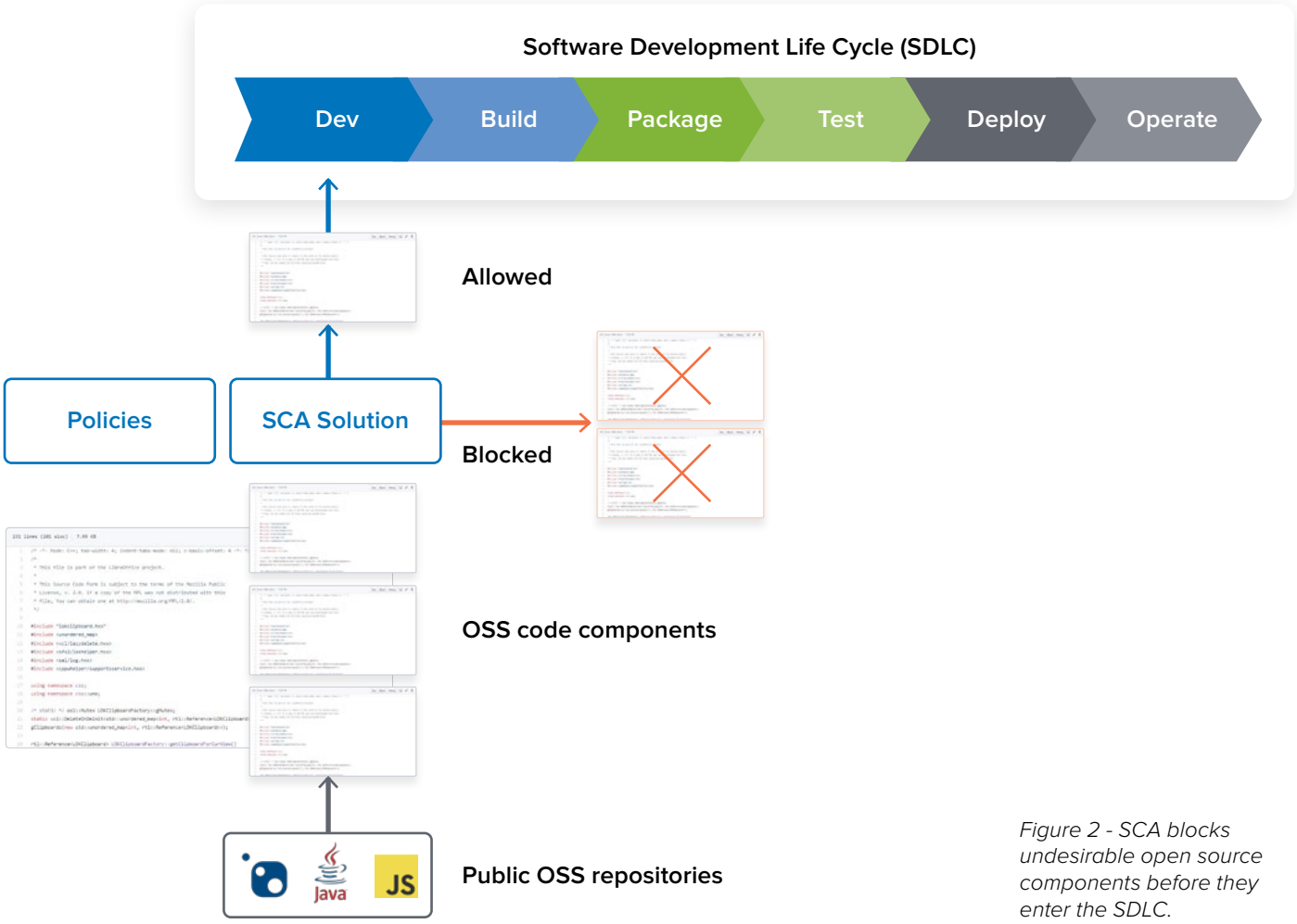


Figure 2 - SCA blocks undesirable open source components before they enter the SDLC.

Providing a Strong Enforcement Engine, Including Default Policies

The rules that affect which open source components are usable and which are not is at the heart of SCA. Developers need to enforce these policies, but they generally don't want to have to know too much about them, nor get bogged down in the minutia of policy definition and enforcement. For this reason, any effective SCA solution for developers must implement both a strong policy engine and a workable set of default guidelines.

As the tech vendor's Engineering Manager described:

"In terms of open source intelligence and policy enforcement across our SDLC, before using Nexus [Lifecycle] in particular, we were struggling to provide a software bill of materials for our products."

Their environment had previously left maintenance of a dependency list up to the development team.

"We know that, with the human factor, sometimes some libraries were forgotten in the list. We also had some problems identifying the licenses of the different embedded libraries that were in our products. That could have resulted in legal problems when we deployed."

The government agency Java Development Manager valued how the solution helped his team define policies, which are selectively applied. He related,

"We can define a separate policy for public-facing applications and one for the internal applications. That is cool."

The MIB Group's Senior Enterprise Architect shared:

"The default policies and the policy engine provide the flexibility we need. The default policy was good enough for us. We didn't really mess with it. We left it alone because [it] pretty much works for our use cases." Policy enforcement was also a critical capability.

> **"We can define a separate policy for public-facing applications and one for the internal applications. That is cool.**

"We have defined policies about certain things at various levels, and what risks we're willing to expose ourselves to."

He then cited the example of setting up a proxy for a library from Maven Central. If Nexus Lifecycle says it has a security-critical vulnerability, it's listed as "security high" or "component unknown." From there, his team can set different actions to trigger. For example, it can warn the developer with an alert or let the QA stage of the development process turn the component into a failure action.

CONCLUSION

SCA processes need to align with developers' workflows and work styles. Without such alignment, the entire SCA effort risks devolving into a frustrating, oppositional dynamic between developers and other SCA stakeholders. The software and the business will suffer the consequences. To avoid this outcome, it is necessary to adapt SCA solutions to developers' needs. This means determining how SCA processes and solutions can improve developer productivity. Success also involves integrating SCA into developer tools.

As IT Central Station members point out in their reviews of Nexus solutions, an effective policy engine allows developers to select only properly licensed, high-quality open source components. This method allows the teams involved in software development to work together to avoid security risks and license compliance problems.

ABOUT IT CENTRAL STATION

User reviews, candid discussions, and more for enterprise technology professionals.

The Internet has completely changed the way we make buying decisions. We now use ratings and review sites to see what other real users think before we buy electronics, book a hotel, visit a doctor or choose a restaurant. But in the world of enterprise technology, most of the information online and in your inbox comes from vendors. What you really want is objective information from other users. IT Central Station provides technology professionals with a community platform to share information about enterprise solutions.

IT Central Station is committed to offering user-contributed information that is valuable, objective, and relevant. We validate all reviewers with a triple authentication process, and protect your privacy by providing an environment where you can post anonymously and freely express your views. As a result, the community becomes a valuable resource, ensuring you get access to the right information and connect to the right people, whenever you need it.

www.itcentralstation.com

IT Central Station does not endorse or recommend any products or services. The views and opinions of reviewers quoted in this document, IT Central Station websites, and IT Central Station materials do not reflect the opinions of IT Central Station.

ABOUT SONATYPE

Sonatype is the leader in developer-friendly, full-spectrum software supply chain management providing organizations total control of their cloud-native development lifecycles, including third-party open source code, first-party source code, infrastructure as code, and containerized code. The company supports 70% of the Fortune 100 and its commercial and open source tools are trusted by 15 million developers around the world. With a vision to transform the way the world innovates, Sonatype helps organizations of all sizes build higher quality software that's more aligned with business needs, more maintainable, and more secure.