Sonatype Nexus Lifecycle Review From A Customer

IT Central Station
Unbiased reviews from the tech community

Sonatype Nexus Lifecycle

Review by a Real User

Verified by IT Central Station

Java Development Manager at a government with 10,001+ employees

JavaDevef0ca

WHAT IS OUR PRIMARY USE CASE?

We use it as a repository or manager. We store all our software application artifacts. We also use it for the vulnerabilities.

HOW HAS IT HELPED MY ORGANIZATION?

Before, we had open-source Nexus Repository, but with Lifecycle we have Nexus RM and IQ Server as well and we can scan .jars. In addition, we have the plugins for individual developers, which benefits us and the developers when they introduce a new artifact into their applications. It helps them identify what are potential risks and defects. They can resolve them right there and proceed there with their development. It also brings intelligence to the open-source artifacts, because intelligent servers scan all the vulnerabilities, identify the problems, and then we can ask the individual teams to fix them. That is a plus. The solution blocks undesirable open-source components from entering our development lifecycle. There are certain .jars which we can block. In terms of open-source governance, the tool tells us all the threats that are out there in the public sector repositories, threats which, potentially, no one knows. We get to know them and we can use the tool to let other people know which direction to go in. 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 plugin which is included for the IDE. It gives a report of the vulnerabilities. It does save time in figuring out the right open-source versions that we need to use. It has helped improve the productivity of the developers by about ten percent.

WHAT IS MOST VALUABLE?

The way we can define policies and apply those policies selectively across the different applications is valuable. We can define a separate policy for public-facing applications and a separate policy for the internal applications. That is cool. Since we have public-facing applications, they are more vulnerable, because anyone from anywhere can access them: for example, Excel and Java scripting. We can detect if we potentially have any .jar open-source product that can become vulnerable. We can define stricter policies for the public-facing applications, versus internal where we are protected by the firewall. We already have a more secure way of accessing those internal applications, so we can limit the strictness of the internal policies a little bit. We can relax some of the rules there defining the different levels, from a security perspective. That is useful. In addition, we like the way, when the product has found a vulnerability, that it also recommends the version in which that particular vulnerability was fixed. It generates a report with all the different types of vulnerabilities that were found. We can then go to individual vulnerabilities and look into the historical information: When, and in what version of the .jar, it was introduced, when it was fixed, and what the usage in the market is for that particular open-source component. That is very useful information to us. The solution's data quality shows in the way that it recommends the correct artifact that we should use and the different versions that are available. Based on that data we can make better decisions. It also integrates well with the IDEs. Instead of discovering a problem during deployment, we can identify the problem right at the development phase. That is a cool feature of Lifecycle. We use Bamboo for our builds and the

Nexus IQ plugin is compatible with Bamboo. We can scan the vulnerabilities at build time.

WHAT NEEDS IMPROVEMENT?

It doesn't provide real-time notifications from the scans. We have to re-scan every time, whenever a build happens. Also, since Nexus Repository just keeps on adding the .jar artifacts whenever there is a build, whenever an application is going up, there is always a space issue on the server. That is one of the things that we are looking for Nexus to notify us about: if it is running out of space.

FOR HOW LONG HAVE I USED THE SOLUTION?

We have been using the product for more than six months now.

WHAT DO I THINK ABOUT THE STABILITY OF THE SOLUTION?

It is stable as of now, the version we are using. We hope that it continues to work as we expect.

WHAT DO I THINK ABOUT THE SCALABILITY OF THE SOLUTION?

We haven't actually explored scalability. But in terms of scalability, if there is anything that we need to add, like CPU, memory, or any extra RAM, that can be added dynamically. But we are not sure if Nexus would need downtime for things like that.

HOW ARE CUSTOMER SERVICE AND TECHNICAL SUPPORT?

Technical support has been really prompt.

IF YOU PREVIOUSLY USED A DIFFERENT SOLUTION, WHICH ONE DID YOU USE AND WHY DID YOU SWITCH?

We used the open-source version before moving to the licensed version of Sonatype.

HOW WAS THE INITIAL SETUP?

The initial setup was okay. It was pretty straightforward. We had some hiccups in the migration itself when we migrated from open-source to licensed Nexus. At that time we faced some issues with the configuration and we had that resolved. But the deployment took only an hour. Because we had an existing, open-source Nexus RM, we had to migrate it to the new, licensed Nexus Pro version. So we had to coordinate with other teams, come up with a plan, and then execute accordingly.

WHAT WAS OUR ROI?

We have only been using the licensed version for six months. But long-term, we definitely see it saving time and that will be our long-term return on investment.

WHAT'S MY EXPERIENCE WITH PRICING, SETUP COST, AND LICENSING?

Pricing is comparable with some of the other products. We are happy with the pricing.

WHICH OTHER SOLUTIONS DID I EVALUATE?

We didn't look at any other options. We have been using Nexus for years. We had some initial sessions with them, we did a PoC and we liked the product. We went ahead with it.

WHAT OTHER ADVICE DO I HAVE?

Their support is good. They help with understanding the environment. They helped us with the initial PoC work. Their product is configurable. We can customize the policies. We had some hiccups, but it was pretty self-explanatory once we understood all the different parts. It was easy to set up and get going. From an implementation perspective, it's not a complex setup, which is a good thing. We have ten people using the solution, which includes developers, some of our managers, and architects. For deployment and maintenance of Nexus, we need just one person, a developer. We have pretty much scanned all of our applications. We have around 30-plus Java applications. Based on the current set of applications and the number of users who are using this product, there are no plans to increase usage at this time.

Read 10 reviews of Sonatype Nexus Lifecycle