



# Sonatype Nexus Repository Review From A Customer



From IT Central Station, the leading review site for enterprise technology solutions.

---

## Review by a Real User

Verified by IT Central Station



**Hagen Rahn**

Senior Software Engineer at SYSTEMA  
Systementwicklung Dipl.-Inf. Manfred Austen  
GmbH

### **WHAT IS OUR PRIMARY USE CASE?**

The primary use case is to store good artifacts our company has produced and proxy external artifacts to help reduce the outgoing traffic and to filter specific components which are known to be vulnerable.

### **HOW HAS IT HELPED MY ORGANIZATION?**

First of all, we now have a well-documented process on where to find any build result produced within the last two years. This documentation has been made available to the whole company. Previously, we had several platforms. The results were sitting in some continuous integration system while build results for other teams were hosted on a shared drive, etc. It has helped us reduce the effort in maintaining several systems. That is a huge benefit. We only have to maintain one system and we can use this system concisely across several locations. It has saved time tremendously. As a rough approximation, I would say we only use 20 percent of the administration time we used previously, so it's saving us 80 percent. And storage management has been simplified so we have been able to reduce hardware resources and backup processes. On the hardware side, as a guess, we have saved 30 to 40 percent compared to what we used previously. It's very substantial.

### **WHAT IS MOST VALUABLE?**

The primary feature is that I now have the ability to provide a central platform for storing build artifacts; a concise way for any project team to store its build with us.

### **WHAT NEEDS IMPROVEMENT?**

I'm looking forward to getting things like automatic governance done, but the bigger priority I'm waiting for is a feature to have hot publication between several Nexus instances. That's more important for me right now because in our company we have several locations distributed all over the world, and each location is producing its own artifacts, sometimes for the same project. I really would appreciate a scenario where the developers could provide their data to the local repository and it would be hot-replicated to the other repository instances. That would be the most important feature for me right now. As far as I know, it's not available, but it's on the roadmap. There are also some minor usability features which are changing from version to version, but that's always progress in the correct direction. They recently added the group artifact version (GAV) search. That was something my users really requested for some time. The next big feature my users request is a remote search so if you have a proxy repository the search can be performed within the local Nexus instance. That would be a major improvement. I think these requests are already known to the Sonatype and already on the roadmap. Also, the code snippets for integrating different artifacts: Currently, they are available for Maven dependencies. We really would appreciate it if they were available for other build systems. That was available in Nexus 2 and it is already on the roadmap, but I'm not sure what the priority is.



### **FOR HOW LONG HAVE I USED THE SOLUTION?**

One to three years.

### **WHAT DO I THINK ABOUT THE STABILITY OF THE SOLUTION?**

I haven't had any negative impact so far, so I'm very confident using Nexus in terms of its reliability. Everything that has been a system issue has been some misconfiguration on our side. Nexus is a very robust system in my opinion.

### **WHAT DO I THINK ABOUT THE SCALABILITY OF THE SOLUTION?**

Currently we run with a hot failover system in each location, which is provided by a load balancer. We only have single-node instances running in each location with the one hot failover node, in case of hardware issues. But we haven't had any problems in the last two years we used Nexus Pro. It's just for our peace of mind, to give some robustness to the system, some additional availability. Initially, we started off with about two gigabytes of RAM. Based on the size of our system we had to increase that to the current configuration of eight. But it's really not a system that consumes high resources. That's one thing I like about the Nexus installation. I used to evaluate Nexus versus Artifactory and, in my opinion, Artifactory consumed much more resources.

### **HOW ARE CUSTOMER SERVICE AND TECHNICAL SUPPORT?**

Technical support is very prompt. If we have a question we get the answer within a few hours. Within less than two hours I usually have a thorough answer. If it's a case where we need to do a conference call, they are very polite and helpful at that point as well. The last issue I had was an authentication issue. It was a basic question on how to perform some login strategies for our customer accounts. I wrote to my technical contact there and, within a few hours, I got a reply. We scheduled a phone call just to reduce the amount of time writing emails. I like the way they get to the root cause of any issue customers have and solve the issue with their best efforts.

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

We started off with several, local, self-deployed strategies like shared devices and web servers hosting some results. We even stored build results on a source code management system.

### **HOW WAS THE INITIAL SETUP?**

When I started with Nexus, I thought the setup was complex, but nowadays I think it's quite straightforward. When I first implemented it, I really needed to gather the basic idea of the repository server, and the guys at Sonatype were quite supportive, getting the system set up initially. Now that I know the mechanics, it was quite easy to first perform the migration from version 2 to 3, and I'm really confident that it wouldn't be that hard to perform another setup or to even get any other operator here in our company. I'm really confident in the use and setup of the system. The initial deployment started off with reading the documentation and then getting things done. It took about two to three days to get to production-ready deployment, including getting all the machines running here our company and IT requests, which were simply waiting time. When we upgraded from 2 to 3, it was a journey of three hours or so. For the initial deployment, our strategy was to first have it in our main location. We wanted to provide a demo setup with one instance and this became the de facto leading, master repository. Later on, we recognized that due to some inefficiencies and VPN problems, it would be better to create local instances of the Nexus installation. We then ported the leading system to the other locations and provided a proxy between all those instances. Right now we have three separate instances in different locations and they all proxy each other, so if one of the VPN connection fails, the other two should still be available. The deployment required me, as the application owner and, for some operational stuff, like repository creation and artifact maintenance, about four backup persons who are capable of replacing me. The maintenance takes about two hours a month. Upgrading from one version to another takes a few minutes. I really would appreciate a live update with zero downtime, but that's currently not on the roadmap, as far as I know. It will be there eventually.



## **WHAT ABOUT THE IMPLEMENTATION TEAM?**

We had a contact at Sonatype who answered questions when they occurred. I had a talk with him for about 15 minutes when I thought about the transition process to the other locations. We basically operate the system on our own but get instantaneous support from Sonatype if needed.

## **WHAT WAS OUR ROI?**

We haven't really evaluated ROI, but we recognize that the time for maintaining systems and providing solutions for the use case of storing and retrieving artifacts has been reduced by about 80 percent, as I said earlier. And we have reduced several hardware systems for storage of artifacts. In my opinion, the most important part is that we can use the setup of this Nexus repository server to get our developers to learn new build systems. There has been an educational purpose as well which, in the end, empowers them to have knowledge of new technologies, and to get started quite quickly. If you think of migrating some thousand lines of Ant code to a Maven project, once this is done the project will run for several years. Maintaining several thousand lines of Ant releasing might become a burden eventually.

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

In my opinion, the pricing is very fair and very customer-oriented. It's much better than any other tool I have used so far.

## **WHICH OTHER SOLUTIONS DID I EVALUATE?**

We began to evaluate different Maven-based repository managers, so we had Artifactory, Nexus, Archiva from Apache, and IBH SYSTEMS. After an evaluation period of three to four months, we decided to go for Nexus, as it is one of the most important players in the field. We also had a vision of incorporating Docker images and the like, which was only supported in Nexus at that point of time. Another difference between them is that Nexus provides Enterprise Support. That is available with Artifactory, but the biggest argument for Nexus was that you could use stuff like Docker proxying, Docker repositories, proxying NPM repositories, etc. We needed a system that's not only for the Java toolchain, but also for C#.

## **WHAT OTHER ADVICE DO I HAVE?**

Our company is about 25 years old. When we started developing stuff in Java, we didn't have much tool support and, as we are a company that integrates other systems, we basically built our own tools. That's quite nice if you can do it, but it is always a burden to maintain. That was another aspect we had in mind. We wanted to reduce the maintenance of self-created systems and to get our administrators to use a standardized toolchain. That really helped to reduce their efforts and keeps us up to date with current developments in the business. As you know, software development is something that is changing every now and then, so every month you have to be up to date with some new framework. It's really hard to keep that up if you try to build your own build-and-development chain. So it's very nice to have several things that are already done by other companies. The storing and distributing of artifacts was really one burden that didn't provide any practical business value but required loads of effort. This is one thing that the Nexus Repository Manager addresses very well. We use Nexus Repository to manage binaries, build artifacts, and release candidates across the pipeline. We used to do it with Nexus Repository Management 2, using their staging system. Nexus Repository 3 didn't introduce staging until one of the most recent releases, so we started off with a staging repository where we would publish our "to be released" artifacts, and then provide a release repository so that we could push them using some continuous integration pipeline configured in Jenkins CI system. I'm going to be using the tagging feature they added recently to provide the staging functionality we used to have in Nexus 2. We don't use it right now for open-source governance because that would require us to rely on the IQ Server and we do not have any licenses for the IQ Server. I do some manual filtering - based on CVE advisories. I exclude some packages from being downloaded, ones I'm aware of that are vulnerable, but that's more or less a manual effort to configure those right now. The governance part is not fully automated, right now, but we are working on that. We currently run with about 240 accounts on Nexus. There are several developers and several system accounts, but we have a daily login rate of about 160 users. We are currently trying to force more teams to use the system. There are still some older systems

relying on very old infrastructure, which are currently not connected to the CI pipeline. This should change within the next one to two years, and then we will have no system that's not connected when using any dependencies for Maven or NuGet or NPM or Docker. They are the four repository types we are currently running. There is a business rule that no system shall get its artifacts directly from the web. Everything shall be proxied by Nexus. Eventually, everybody in the company will be forced to use the Nexus. Overall I would rate it at nine-plus out of ten. I'm very pleased using Nexus Repository Manager. And I'm also looking forward, based on the experience that I have had with them and their technical and business department, to adding to our toolchain regarding IQ Server and lifecycle products. That is something I have to sell internally, but I think this will come eventually.

[Read 8 reviews of Sonatype Nexus Repository](#)