PeerSpot

# Case Study

Veracode

**Michael Calabrese**

Vice President of Engineering at Avant Assessment

Review by a Real User

Verified by PeerSpot

## What is our primary use case?

We use it for security validation. As a company, we need to make sure that our code is secure. Not only do we need and want to do this for ourselves, but we also need to do it because of our security obligations to our clients.

## How has it helped my organization?

It has been helping us capture security vulnerabilities that we would not catch otherwise.

When it comes to our ability to fix flaws, Veracode has given us more visibility into certain flaws that could show up, flaws that can be subtle and not seen in the code. For example, though it was not obvious, there was a case where a developer naively added the authentication into the code, which we're not supposed to do, obviously. It was not seen by our review process, and Veracode caught it and we were able to eliminate it.

It has also helped us to save time. The example, and where I see the most benefits of that, is in the Security Labs, where I have the developers training and constantly improving their security, and remembering their security techniques. That way, they are more proactive and make sure things are correct. They're faster because they're doing it in the first place.

Overall, in terms of our security posture, Veracode has made us more reliable. We're finding those flaws and our clients trust us more because of it.

And when considering whether it has reduced the cost of development, security, and

operations for us, the short answer is no. But the long answer is yes. It clearly has added more procedures in place, which we needed to have, and that has definitely increased the cost of development. But in the long-term, how much have we saved from the intangible of a flaw not being exposed?

## What is most valuable?

The Security Labs feature, in particular, is valuable, and I have been using the static code analysis as well.

## What needs improvement?

I do have two pet peeves with the platform.

The user interface is slow as a dog; really slow. You go to any modern interface and it's a lot more snappy. Even though I understand a lot of what they're doing and why it might be slow, it is really slow. You click on something and it takes two to three seconds. That doesn't sound long, but it just feels super clunky. There are many times when their product goes to check my code and it dies, and I don't know why. I've contacted support and they're not really helpful with this particular problem. I go to the logs and I look at what I can but I can't tell why the check process has essentially just died in the middle of checking.  Other than those two complaints, I still find it very strong and powerful.

In terms of additional features, the big one I would like to see is that, right now, I have to click through too many things to get to the triage report, which is the main thing I want to see for anything. I have to click through this one screen that doesn't give me any information and I really just want to get to the mitigation review screen quickly. Anything that would save me going through clicks and four or five different screens, because the interface is slow, would be fantastic. I want to get to that mitigation screen because the summary screens are not all that interesting to me. I need to know, "Is this mitigated? Is it not?" and get it checked off and reviewed.

## For how long have I used the solution?

I've been using Veracode for two years.

## What do I think about the stability of the solution?

It has been a very stable product. I don't think the issues that we're having are related to its stability.

## What do I think about the scalability of the solution?

The scalability is "medium" because one of the things I've been having to do now is scale out more of the microservices by tier so that I can verify that the code is correct per tier. For me to

scale up like that seems to be taking a lot of effort. I might be doing something wrong. Maybe it could be solved in a different way. But the scalability is average. On a scale of one to 10, I would put it at about five.

We do have plans to use more of Veracode. We are expanding into the SCA, where it is scanning the containers, and we've also just contracted with Veracode to do penetration testing.

## How are customer service and support?

The one time I had to use their technical support for the bug where a code check dies, I found them a little off-putting. They have never really fully answered the question. I got tired of asking because they didn't understand what I was saying.

During installation, their support was fantastic, a 10 out of 10. But in dealing with this one issue, I would give them a two.

## How would you rate customer service and support?

Neutral

## Which solution did I use previously and why did I switch?

We haven't used another solution. Veracode is the first solution of this kind that we have worked with.

## How was the initial setup?

The initial deployment was pretty straightforward. We ran into some issues, but honestly, nothing out of the ordinary. I would definitely put it toward the easy side. I found the documentation to be appropriate.

The deployment time was days.

We are using Jenkins as our CI/CD. We're using Amazon Cloud K8 deployments.

We integrated it in two different ways. The original way was with AWS CodePipeline. For that, we used Veracode's Docker service. Once we had it hooked up and could send the file, that was pretty easy to use. The second way is we now actually use Jenkins for our code build. We do the same thing although we're going to change to the Jenkins plugin here shortly. But it was still the same, with the ability to use Docker to send the file to Veracode. Once we wrote it, it was really easy, which is why we did it that way on Jenkins. Through both of them, the implementations worked easily.

From the time of deployment, we saw the benefits within one to two months, which was fairly immediate.

There is maintenance required because, sometimes, the pipelines for our code review essentially stop. I have to go and check that, as I mentioned earlier. The second piece of maintenance is that if there are any flaws or

false positives, you have to mitigate those results. We have two people involved in the maintenance.

## What about the implementation team?

I did the original Amazon CodePipeline implementation by myself and got it hooked up. As we went to more complex things, with Jenkins, that was done through an integrator DevOps team. On our side, it was just me involved.

## What was our ROI?

I'm sure we have seen ROI, but I do not have a direct metric on it. There are a lot of intangibles in that. For example, what would be the cost of a particular flaw that we caught with Veracode, if it had gone live?

## What's my experience with pricing, setup cost, and licensing?

When I looked at the pricing, it was definitely a value. In terms of the service and what it's checking, the cost was very reasonable, particularly because we could have multiple code bases as part of a project.

Make sure that you're comparing apples to apples if you're concerned about the price of Veracode versus what you're reviewing. Some of the stuff that Veracode does and applies is not the same for other services. When I really compared apples to apples, I found Veracode to be rightly priced.

There were no costs in addition to the standard licensing fees, although we just signed up for a couple of other products.

## Which other solutions did I evaluate?

We looked at other solutions but one of the big things that made a huge difference with Veracode had to do with pricing. Because we're moving more and more toward a microservices architecture, and we have about six code bases that make up our entire product, they made it clear that as long as something was a part of our product, it was the same price. That was amazing to us because competitors charged per code base. It was definitely a more economical solution and the one that made more sense, and is more in line, with our product. That really simplified the thought process for us and was a huge competitive advantage.

## What other advice do I have?

Veracode is a valuable tool to have in the toolbox to prevent vulnerable code from going into production. Veracode's false positive rate has been very good. It's reasonable. False positives take more time, but I have not noticed

that time to be a significant burden. Its policy reporting for ensuring compliance with industry standards and regulations is adequate.

In terms of having visibility into application status at every phase of deployment, Veracode doesn't provide that. It doesn't control the whole deployment cycle, so there's no way it can report on all of it.

The platform's interfaces look slightly antiquated but don't let that stop you from using it, because it has been a good solution for us.

The biggest lesson I have learned using it is that it's really nice to have these security checks in a single place in your code pipeline. We have multiple security companies at this point, but having the code review and product review security in one place helps us know that that part is "containerized." Having everything dealing with code review in one place is nice.