CASE STUDY

ROOKOUT

Rookout Slashes Otonomo's Debugging Time

> Rookout has saved our developer team a lot of time when investigating errors. Our team is now able to do so by simply adding a breakpoint.

otonomo

> Rookout gives us a sense of security. At anytime, we can see what is going on in our system, even as it is running live. This allows us to address issues when and if they arise.

ROOKOUT otonomo

Rookout Slashes Otonomo's Debugging Time

The ability to investigate issues in real-time production environments drastically reduces the time developers waste debugging.

The Otonomo Debugging Challenge

Before adding Rookout to their workflow, Otonomo had been struggling with long investigations of errors in all development environments, including production, staging, and dev. The R&D team had been struggling to solve these issues using "the old fashioned way". This was done by adding log lines, measuring additional metrics, and attaching the PDB/GDB debugger to running processes, and attempting to debug on a developers' local machine. This classic method was an excruciatingly long process and Otonomo's developers felt that they were spending too much time in doing so.

In an attempt to modernize the time-consuming debugging process, the developers at Otonomo decided to try a new method that would allow them to quickly get to the root cause of bugs and customer issues. They were looking for a solution in which they wouldn't need to add log lines or go through the full CI/CD cycle again, and instead be able to fix what was needed while the software is running live.

Adopting A Live Debugger

Otonomo's adoption of Rookout into their debugging workflows has been predominantly for use in their staging and dev environments. Their R&D team has been able to save a significant amount of development time, as they are now able to turn to Rookout when investigating an issue, instead of needing to subject themselves to yet another iteration. According to Nir Nahum, Software Team Lead at Otonomo, "Rookout is a useful tool that helps minimize the need to go through long deployment cycles."

> *Rookout gives us the sense of security that when something goes wrong, we have a strong tool that allows us to see what is going on in our system while it's running live.*

Most often, when investigating what is going on in their code, Otonomo's developers first attempt to reproduce the issue in production to try and understand what is happening. "Rookout gives us the sense of security that when something goes wrong, we have a strong tool that allows us to see what is going on in our system while it's running live. We try to add Rookout to anything we can in the system, whether it's the onboarding of a new microservice or something else.", said Nir.

Not only is their debugging time cut significantly shorter, and the troubleshooting process is made much easier and more efficient, but it also helps speed up client integrations, especially when doing so remotely. They are able to easily set a breakpoint or two - or a dozen breakpoints - to make sure the client is working where they should and gives them the data they need to understand what is working or isn't, even when doing so over a video chat. No longer do their developers have to go and add another metric, another logline, or go and search for the data every time a specific action happens. They are able to get all of that information easily and efficiently by using Rookout. It gives them the confidence and empowerment their developers need to make things run smoothly.

Results: How Rookout Helped

As is with all companies, time is always of the essence. With Rookout, Otonomo has found that bugs can be investigated quickly in all development environments (production, staging, and dev). According to Nir, "Rookout has saved our developer team a lot of time when investigating errors. Our team is now able to do so by simply adding a breakpoint and as such are able to avoid having to go through guessing the root cause, the uncertainty of an issue, or having to add additional logs and roll to production."

The developers at Otonomo have experienced a rise in productivity by being able to avoid deployments of "only-logs-and-metrics" code. Rookout has become their on-demand solution for dealing with issues when they appear.

"The biggest difference is that you can discover problems while your system is running live. We are able to do this in just a few minutes instead of a few hours. Our developers can easily set the right breakpoint, instead of having to go and do a binary search throughout all of the options that exist in the system.", said Nir.

Nir went on to say that, "In case you are spending too much time trying to investigate an error in your environment and you are trying to understand whether a specific line was processed, what were the variable's values and the stack trace, Rookout is a great fit for your needs."

> *Rookout is a useful tool that helps minimize the need to go through long deployment cycles.*

A Few Words About Otonomo

Otonomo is the premier one-stop-shop for vehicle data. Since its founding in 2015, Otonomo has built a vehicle data platform and marketplace that now fuels an ecosystem of OEMs, fleets, and more than 100 service providers. The platform ingests more than 4 billion data points per day from over 40 million global connected vehicles, then reshapes and enriches them, in order to accelerate the time to market for new services that improve the in-and-around the car experience. Otonomo's platform allows automotive OEMs the opportunity to create new revenue streams by enabling the utilization of the vast amounts of data vehicles generate on a daily basis and that OEMs are required to store and maintain.

In addition to its proprietary data platform, Otonomo has developed a robust suite of SaaS offerings that provide data consumers with additional capabilities and vertically specific applications. Privacy by design and neutrality are at the core of Otonomo's platform, which enables GDPR, CCPA, and other privacy-regulation-compliant solutions using both personal and aggregate data.