CASE STUDY

ROOKOUT

Rookout Improves OptimalQ's Developer Workflow

> *OptimalQ's developers have fallen in love with Rookout, having found that Rookout saves them about an hour or so on each bug.*

> *The introduction of Rookout into OptimalQ's software significantly reduced the time their developers spend on production maintenance tasks, thus instead allowing their developers to focus on value creating tasks.*

OptimalQ
Science of Availability

CASE STUDY

ROOKOUT

OptimalQ
Science of Availability

Rookout Improves OptimalQ's Developer Workflow

Debugging in production eradicates wasted time, by getting engineers the data they need to optimize their workflows

The OptimalQ Debugging Challenge

Before adopting Rookout into their workflow, OptimalQ found that a significant amount of their developer's time was spent researching production issues. OptimalQ found that the majority of bugs they were dealing with were very data relative and therefore difficult to reproduce and keep track of. Their developers would turn to copious logging as a solution. They would add a large amount of logs and then use various systems to aggregate them. Then the logs would be analyzed to get the relevant data for the specific bug they were working on - hoping they caught what was needed.

The Adoption Process

"It was an amazing experience for us to onboard", said **Yechiel Levi, CTO and Co-Founder of OptimalQ**. "All it took was to add a records import into our own code, and that was just a one-liner."
OptimalQ's main concerns in the beginning were security-based, with their greatest concern being for their client's privacy and how Rookout, as a sub-processor for them, would be able to maintain that high level of security.
"From the start, security compliance was at the top, which was specifically a concern in regards to the data that is being transmitted. And, of course, our source code is never shared with Rookout."

> *OptimalQ's developers have fallen in love with Rookout, having found that Rookout saves them about an hour or so on each bug.*

Results: How Rookout Helped

Rookout enabled OptimalQ to quickly and easily identify and pinpoint bugs in production, thereby significantly reducing both the amount of time their developers spent debugging and their stress doing so. Yechiel said that, "while skeptical at first, OptimalQ's developers have fallen in love with Rookout, having found that Rookout saves them about an hour or so on each bug".

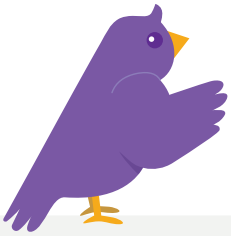
ROOKOUT

OptimalQ
Science of Availability

The introduction of Rookout into OptimalQ's software significantly reduced the time their developers spend on production maintenance tasks, thus instead allowing their developers to focus on value creating tasks. According to Yechiel, "my developers said that it saved them about an hour or so on each bug", as well as mentioning that, "the cost alone is worth it for even just 3 bugs that you're maintaining in production".

Yechiel also added that, "They (the developers) said, this saves us a bunch of time when we need to solve production issues. We can get the customers issues and complaints fixed immediately. Being able to do so really reduced the amount of time that our developers have had to research and reproduce."

The introduction of Rookout into OptimalQ's software significantly reduced the time their developers spend on production maintenance tasks, thus instead allowing their developers to focus on value creating tasks.



A few words about OptimalQ

As the pioneers of the science of availability, OptimalQ handles production systems for large companies, helping them to engage with their customers when the customer is both physically and mentally available. Using AI and machine learning algorithms, OptimalQ identifies and analyzes user behavior, and translates it into availability insights that can predict the most effective times to engage with leads and customers.

Their stack

OptimalQ's system is a distributed system built of multiple microservices. There's multiple instances of each microservice that communicate with each other. When an issue occurs, OptimalQ has to trace through all the different microservices in order to find the data on the bug they are trying to fix. Their application is Python based and deployed in Kubernetes.