Rollbar

Case Study

# Rollbar Enables Continuous Deployment Strategy to Innovate Faster and Improve Customer Experience

Rollbar’s Continuous Code Improvement Platform helped one customer to spend less time testing while catching, and fixing, escaped errors before users do.

## Goal:

# Catch Bugs Faster to Support Continuous Deployment Strategy

A large online retail platform with over 100 million transactions over thousands of types of items and with hundreds of thousands of customers needed to innovate fast to stay competitive. The Retail-as-a-Service (RaaS) sector is highly competitive and user experience is critical. The platform also works with many other large platforms, so efficient and smooth operation is key.

The customer explained that they wanted to innovate faster and move faster with less time for new features in staging environments and a faster push to production. To do that they were going to need a solution to alert them to production errors in real-time and also provide enough context to fix the issue or roll back the right part of the deployment. The goal was to catch and resolve errors before they impacted customers.

In the past, the customer was unable to use this approach because it took them too much time to investigate issues. They had to delve through logs, ferret out details, find the correct URLs in the APM and look at HAProxy logs. It took significant time and effort and meant they were unable to implement a faster push out of staging. In addition, their infrastructure was getting very complex with nearly 100 microservices at play, so pinpointing issues was getting tougher with causes coming from potential sources as disparate as storage, recent deployments, Kubernetes, cascading events across microservices and corrupted data. Their existing monitoring solution, Datadog, could surface issues such as ‘5% of shopping carts are down’, but it was noisy with too many incident alerts and didn’t provide sufficient context to track the root cause or even which segment of new code should be rolled back while investigation continued.

### Customer at a Glance

**Tech Stack:**

Ruby on Rails, Java, Node.js, Jenkins, Teleport, Atlassian Opsgenie

**Monitoring Stack:**

Datadog, AWS, Redis, RabbitMQ, Cloudflare, and MongoDB

**Solutions:**

Low Risk Releases, Production Code Quality, Customer Support

Strategy:

# Use Rollbar for Real-Time Code Error Notifications & Rich Contextual Information To Fix Quickly

The customer deployed Rollbar seven years ago and it is now used by the entire 150-person engineering team. It is the trusted tool for monitoring and resolving code issues and allows the customer to make hundreds of changes a day because of their confidence in their ability to quickly resolve or roll back anything causing an issue.

## Real-time Production Error Monitoring

The customer has said that they are investing in reacting fast to things that happen in production. Although they still perform QA in staging environments, they spend less time there with an end-to-end QA pipeline under ten minutes to run thousands of tests and then push to production. They expect to go from initial QA to production in under ten minutes because they no longer fear errors in production code.

Rollbar gives this customer the confidence they need to support their new continuous release schedule because they know instantly when there is an issue. Rollbar is the fastest monitoring tool they have used. The customer still uses Datadog for tracing and logs along with other tools like AWS, Redis, RabbitMQ, CLoudflare and MongoDB but it considers Rollbar the first line of defense.

“Rollbar is not the only monitoring tool that we use…but it’s the fastest,” the customer’s head of infrastructure engineering said. “Just set up those Slack notifications and [see] what’s going on. If it cannot be addressed right away, then it’s shared with the platform’s site reliability teams or in the wider engineering channel.”

Sometimes the solution is immediate and other times the alert and associated information is shared to the wider team to get a more difficult issue resolved, especially if it is a site-wide reliability issue, in which case they are able to put a much larger group to solving the problem.

> Rollbar is not the only monitoring tool that we use…but it’s the fastest.
>
> — **Head of Infrastructure Engineering**

> There may be so many changes that we are performing and I think one of the benefits that Rollbar is giving is the rich context. We see right away from Rollbar, which change, what stack, what application, what's the exception. We get the detailed information of what's broken, and developers can act fast and fix that.
>
> —— **Head of Infrastructure Engineering**

## Detailed Contextual Information to Quickly Fix Or Roll Back

The customer needed more than just real-time notifications. Pinpointing the cause of errors is at least as important as knowing about them promptly. Rollbar provides detailed contextual information to identify root causes immediately without lots of time triaging and digging through logs. The customer uses the rich context that Rollbar provides to see right away which change, which code stack, which application is involved and what is the exception that occurred, even in complex microservices environments. One of the reasons the customer chose Rollbar is the amount of information about the context of exceptions, including not just what's broken but all the other related information. That allowed the customer to act fast to fix exceptions.

"There may be so many changes that we are performing and I think one of the benefits that Rollbar is giving is the rich context. We see right away from Rollbar, which change, what stack, what application, what's the exception. We get the detailed information of what's broken, and developers can act fast and fix that," said the customer.

Results:

# Business Benefits for the Customer Include:

### Increased Developer Productivity

First, Rollbar's Continuous Code Improvement Platform isn't a tool that needs to be constantly overseen by the development team. It's operational overhead is very low. That allowed the customer to spend time on writing code rather than setting up and running a monitoring system.

### Faster Time to Awareness and Resolution

Plus, once an issue is detected,the customer didn't need to spend hours, or days, triaging the issue by digging through logs to find the cause of the issue. Or even figuring out what person or team should be troubleshooting it in the first place. For example, if they got an alert in Slack that the shopping cart was down they can see immediately that the alert includes 'cannot connect to the database, connection refused'. That allowed the customer to see when the error occurred to the second and specifically what the error was, making it much easier to track down the offending code.

### More Frequent and More Confident Releases

Since the customer is now focused on more frequent deployments they are investing less in testing. They have made the decision to deploy changes to production right away - to make changes fast and get alerted fast. Their philosophy is to make the change, potentially get a Rollbar notification and revert the change. "We would rather deploy the change to production right away, and we have decided that the key to our innovation is to make changes fast and get alerted fast," said the customer's head of infrastructure engineering.

### Better Customer Experience for Increased Revenue

Since Rollbar is powering their goal to innovate fast, the customer can get updates and changes out to customers on a daily basis. And because they're confident they can catch and resolve any error in production, they're pushing changes almost immediately into production. "So developers are spending time writing the code, not setting up the monitors and alerts, and the entire monitoring system," said the customer's head of infrastructure engineering. This allows them, in a sense, to perform real-time A/B tests with customers. Now they can test changes with real customers and get the valuable insights they need to deliver a great experience that gets them to convert.

> We would rather deploy the change to production right away, and we have decided that the key to our innovation is to make changes fast and get alerted fast.
>
> —— **Head of Infrastructure Engineering**