Lyft utilizes Bugsnag's two-way sync integrations to ship new product features 40% faster

bugsnag
A SMARTBEAR COMPANY

lyft

CHALLENGE

Lyft realized their mobile crash reporting tool was hindering their ability to deliver a consistently positive experience to their users. Unreliable crash reports and inefficient debugging workflows caused bugs to slip through, despite their team's best efforts to thoroughly test and debug prior to releasing new versions of their apps. In the end, they decided to move away from Crashlytics.

RESULTS

Lyft has now standardized on Bugsnag for crash reporting across all mobile applications — iOS and Android. Bugsnag is used throughout development, testing, and post-release, allowing Lyft engineers to reliably find and resolve crashes before they can impact their users. They have even decreased the time spent troubleshooting errors by 30%.

INDUSTRY
Transportation Networking and Consumer Mobile App

PLATFORMS
iOS, Android

RELEASE CYCLE
Weekly

DAILY ACTIVE USERS
~1 million

In a highly competitive industry, Lyft needed a trustworthy crash reporting tool to ensure the quality of application releases.

Lyft is a consumer application company that facilitates peer-to-peer ridesharing, with a focus on fostering community for the drivers and passengers on their platform. For a consumer mobile application in a highly competitive industry, reliability and precision in engineering can make or break the customer experience. For this reason, the main priority for Lyft's engineering team is reliability with the goal of "making sure new versions of our apps are fully stable and bug free so our users have a positive experience," says Billy Pham, Technical Program Manager for Passenger Applications.

However, their team lacked trust in their previous tool because, despite their well built-out processes for testing Alpha and Beta versions of their applications prior to release, bugs would still surface in major releases.

They realized that missing data and error reports were causing issues in production. "We would not catch every single event, and so crashes and bugs were slipping through, and of course, this caused a lot of pain for our users. It wasn't reliable," says Billy.

Lyft's engineers also had difficulty assessing the impact of individual errors because there was a lack of visibility into how frequently certain errors were occurring. This was especially problematic given their fast release cycles of one week. They needed a way to understand error impact to help with decision making around release blockers.

These issues, coupled with a plan to triple the size of their engineering team, led them to search for a more reliable tool that would support their hyper-growth.

Observability and prioritization for high-quality releases

Lyft has now implemented Bugsnag across all stages of the mobile development cycle which helps them detect and fix bugs before they impact their users. Since they can deliver on their main priority of reliability, they have even made gains in productivity, shipping new releases faster than before.

"Bugsnag is a great way of logging crashes reliably, efficiently assigning those issues to your teams, and resolving them in a timely manner in order to focus on new features and provide a better experience for your users."

— Billy Pham, Technical Program Manager

Since implementing Bugsnag, most crashes are now caught during Alpha and Beta releases, and the time to resolution for bugs has gone down by 30%. When monitoring a release, Lyft uses filters to find errors by version, a powerful functionality that gives them visibility into each release without much effort. Dave King, Engineering Manager for Mobile Infrastructure, explains that their team relies on the quantitative data Bugsnag provides on bugs in a particular version. This visibility allows them to triage and fix all high impact bugs in a particular release without hassle. "Bugsnag has great functionality to sort and group all crashes, so we're able to know which are the most impactful issues based on the number of events and number of users that experienced a crash," says Billy.

Efficient debugging process saves time for innovating Lyft's platform

Lyft has improved their debugging process by integrating Bugsnag with JIRA and Slack, and have decreased the time spent managing bugs in their tools by 30%. Lyft uses JIRA to manage bugs and has streamlined the process by integrating with Bugsnag. The two-way sync integration allows them to avoid duplicate work because creating tickets and closing bugs across both tools happens automatically with one click. This has made prioritizing bug fixes more efficient for their team, and is one of the features they feel they could not live without. Billy explains that "this saves us a lot of time since we don't have to copy paste and manually write descriptions. A one click button creates a ticket, and we can just pass it along to the developer. It has made a big impact on our speed."

To improve visibility on errors throughout their team, they use Bugsnag's API to send the top 3 errors in each platform to Slack. Receiving alerts in Slack helps the team stay informed on the status of bugs, and allows them to collaborate on fixes. It also encourages more ownership over their work since everyone is able to see incoming bugs. Another improvement to their debugging process has been adding engineers to on-call duty as they are the best placed to quickly understand and fix a bug in Bugsnag, with the added bonus of increasing ownership of their work.

All these gains in efficiency taken together have in turn made it possible for their team to release product enhancements and new features 20-40% faster. Overall, Lyft "considers Bugsnag to be a better tool in terms of functionality and reliability."