The Dropbox mobile team aims for three nines of user availability with Bugsnag

bugsnag
A SMARTBEAR COMPANY

Dropbox

GOAL

Dropbox's mobile team set the goal to achieve three nines of user availability by the end of 2021, which would bring it in line with the team's session stability score.

RESULTS

With Bugsnag, mobile engineers have the stability management features required to monitor release stability and address errors with speed in order to improve the user stability score.

INDUSTRY
File hosting service

APP & PLATFORMS
Android, iOS, Web

EMPLOYEES
2,500+

LOCATION
San Francisco, CA

Dropbox values product health and developer productivity

Since its first product release in September 2008, Dropbox has focused on delivering a modern workspace where users can store, access, and collaborate on content. Today, Dropbox's global collaboration service provides a wide range of business and personal offerings around cloud storage, document collaboration, and file synchronization that are accessible on web, desktop, and Android and iOS mobile applications.

"Product health is one of our highest focuses," states Angella Derington, Dropbox Mobile Staff Engineer. "One of the things we want to achieve for all of our mobile apps is to hit the three-nines on user availability. To do that, we need tools that any engineer can utilize to troubleshoot where crashes come from and identify what's happening in each release."

Moving from homegrown to Crashlytics to Bugsnag

Years ago, Dropbox built a homegrown crash reporting system. However, staffing and resourcing dropped off over time, which led to a stagnant system that did not provide the features required to monitor application health.

Approximately two years ago, the observability team, which owned all tooling to observe behaviors on client applications, recognized the need to adopt a new error monitoring solution that would evolve faster and enable Dropbox to benefit from the expertise of a third-party provider. Evaluation of external crash reporting solutions began, with the objective to find a solution that could be implemented company-wide.

To bridge this interim period, the mobile team decided to try a free solution. "We moved to Crashlytics for about a year, which was an improvement from our homegrown solution but lacked a lot of things, especially breadcrumbs and query tools," states Angella.

"The mobile engineers were easy to convince because there's so much functionality with Bugsnag, and it's easy."*

When Dropbox selected Bugsnag as its stability monitoring solution, Angella's team became the first internal customers. "The mobile engineers were easy to convince because there's so much functionality with Bugsnag, and it's easy," explains Angella. "It was a very smooth transition, and I think the full migration took us only about six weeks. We'd already abstracted our crash reporting library, so it was pretty straightforward to plug in Bugsnag."

Bugsnag provides full visibility during error investigation and experiments with breadcrumbs, automatic grouping, and custom filtering

Today, Bugsnag enables the Dropbox mobile team to monitor stability by release and prioritize and fix errors with speed, precision, and ease. Breadcrumbs is one of the critical Bugsnag features that the team relies upon as a best practice for debugging why a crash happened and for resolving customer complaints.

"When we moved to Bugsnag — especially with its breadcrumbs — engineers were suddenly saying, 'I can look at a crash and know how to fix it!'"*

"Bugsnag removes uncertainty and anxiety around the time it takes to troubleshoot a crash and any concerns about reproducibility. Having information that pinpoints where to start looking in the code is groundbreaking for engineers," states Angella.

In addition to breadcrumbs, the mobile team benefits from indexing of custom events, which is particularly useful with its weekly beta releases. "I had never experienced or seen indexing of custom events in a solution before," states Angella. "We do a lot of A/B testing and feature gates, and it's extremely helpful to be able to index so you can create queries and bookmarks around them."

Historically, identifying the impact on a class of crashes proved hard to accomplish. "We've used Bugsnag a lot to identify impact and be able to fix classes of crashes a lot faster than we would have otherwise," explains Angella.

Out-of-the-box ANR detection and workflow integrations make debugging precise and actionable, while bookmarking and custom alerts help teams stay focused

The team also uses bookmarks on queries: "While Bugsnag does a pretty good job of identifying similar crashes, sometimes exceptions are chained. It's helped a lot to be able to set up a bookmark that's looking specifically for the exception message and be able to find all of those errors."

 The Android team especially appreciates Bugsnag's ability to handle Application Not Responding (ANR) errors. "I love that ANRs are in the tool. On previous teams, we'd forget about ANRs until customer complaints start coming in because we weren't monitoring them in our tool," states Angella. "Bugsnag is the first crash reporting tool where it's actually being sucked in from Google Play, which is perfect."

Dropbox uses Bugsnag's integrations with Jira, Slack, and PagerDuty to automate tickets and provide targeted, customized alerting. "In the past, we've turned off alerting because this fragmentation amounts to so much noise that no one pays attention to," states Angella. "The ability to be specific about what you want to be alerted about from your crash reporting tool is very helpful."

With a passion to deliver high quality apps, Dropbox expands Bugsnag usage, allocating the time-saved ROI towards building new features
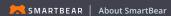
Dropbox's mobile teams use breadcrumbs for application health, and this best practice is spreading to the rest of the organization.

"With newer Dropbox apps like Passwords, it's a simpler task to build out breadcrumbs and handled exceptions in our code base," states Angella. "The original Dropbox app, which has evolved from the very beginning, is around half a million lines of code right now and at varying stages of age, so it will take a little longer before it's as end-to-end as apps like Passwords."

Today, the teams working on this desktop client application are putting in the work to add breadcrumbs throughout the large code base. This critical step will create consistency with logging and enable other engineering teams to fully benefit from Bugsnag in the same ways that the mobile teams are today.

"It's a passion of mine to deliver high-quality applications, especially ones that I work on. Of course, engineers don't love fixing crashes; they like building new code. Minimizing that time with Bugsnag is magical."

* — Angella Derington, Mobile Staff Engineer