SENTRY

Save time. Don’t let errors throw you off.

We get it. *Errors suck.* And you don’t want to spend too much of your time fixing them, dealing with them, investigating them, etc. We should work toward creating the right infrastructure and culture to handle these unanticipated issues and address the destruction that these errors have on the team’s efficiency.

What you should know about Sentry:

- Get a more efficient workflow, without added levels of complexity.
- Get alerts only on the issues you want to know about with configurable alerts.
- Alerts lead you directly to the information you need.
- Suspect commits pinpoint (probably) the commit and engineer responsible.
- Resolve errors via a commit and get back to coding, quickly!

Errors hurt business

Errors are inevitable (in life and in code), causing a lost trust from customers, lost revenue, and decreased team productivity. Instead of trying to eliminate them altogether, we should minimize the impact that these errors have on the software development process.

- Crashes increase churn by as much as 534%, representing a 6 increase from "average" churn rates[1], based on research from tens of thousands of mobile apps representing hundreds of millions of application launches.
- Users are up to 8 times less likely to use a product the next day after a crash[2].
- Acquiring a new customer is anywhere from five to twenty-five times more expensive than retaining an existing one[3].
- Fred Brooks, engineer and author, explains that "we usually expect the number of bugs to be smaller than it turns out to be. Therefore testing is the most mis-scheduled part of programming. Coding is generally ⅙ of time."[4]

Without careful attention to error management and code performance, developers can spend more time handling code issues than they do writing new functional code. This can have detrimental effects.

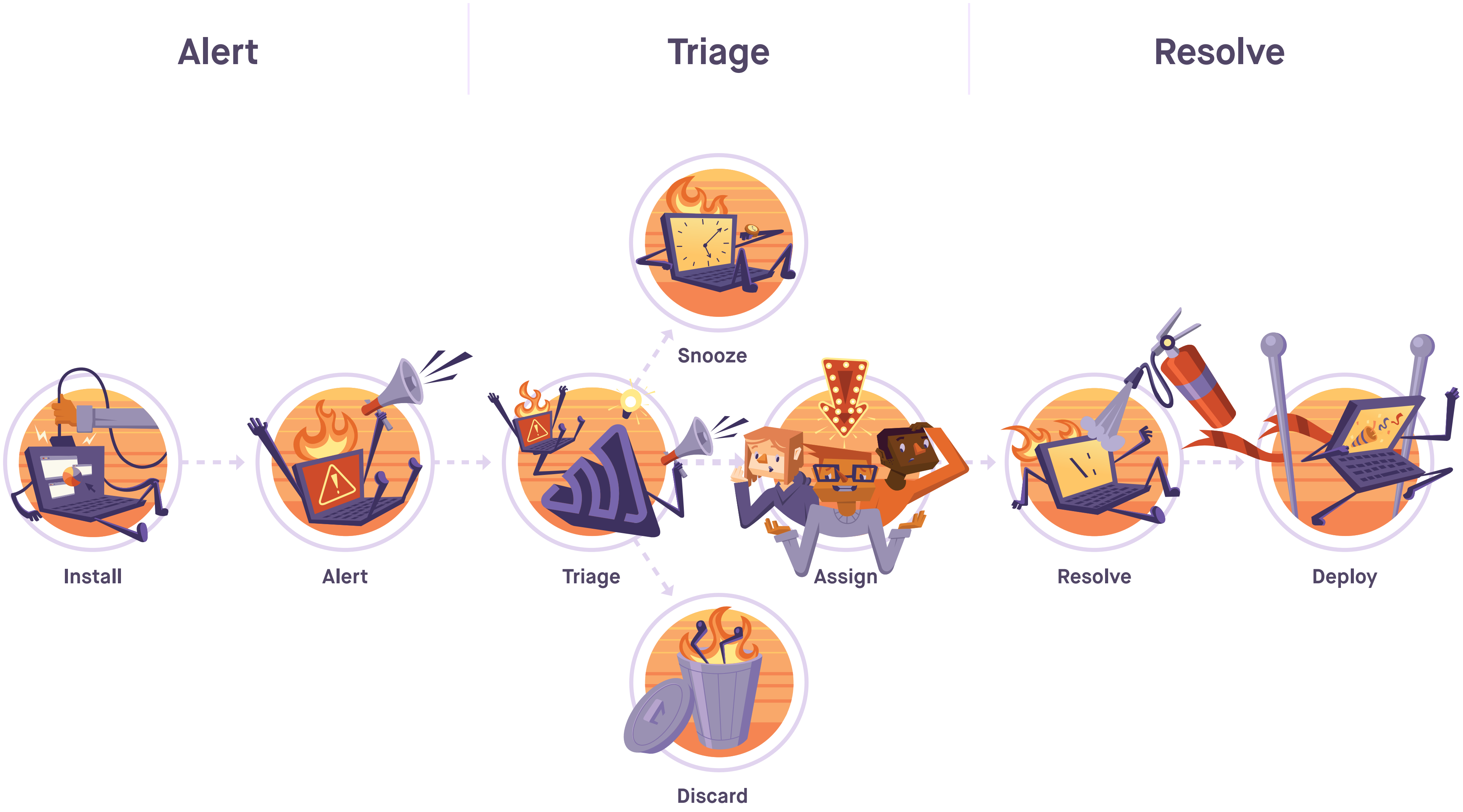
Workflows are predictable

Fortunately, the predictability of development workflows provides space for this mindful infrastructure and culture. With every development workflow, an application is designed and built. Code is reviewed, tested, and deployed. Issues are discovered.

Without an error-monitoring tool like Sentry, your customer support team will probably first hear about an issue through an email complaint, a note to the customer support team, or perhaps whining on Twitter. From there, a tech lead, product manager, or engineering manager will look through logs and changes to find the error.

They might also try to identify which engineer made the commit, what files and functions have changed, and contextual evidence such as the browser and operating system. Of course, all of this is done while looking for key indicators of whether it was a reproducible front-end error or an issue on the server side.

Eventually, a developer will continue triaging, fix the issue, and deploy. The entire process is long and can be frustrating for everyone involved, especially the developer, who's almost always the last one to know there's a problem but the only one who can fix it. When there's a problem, developers want to know about it early, solve it, and move on. That's where Sentry can help you develop a more efficient strategy for handling errors.



Alert

Triage

Resolve

Snooze

Install

Alert

Triage

Assign

Resolve

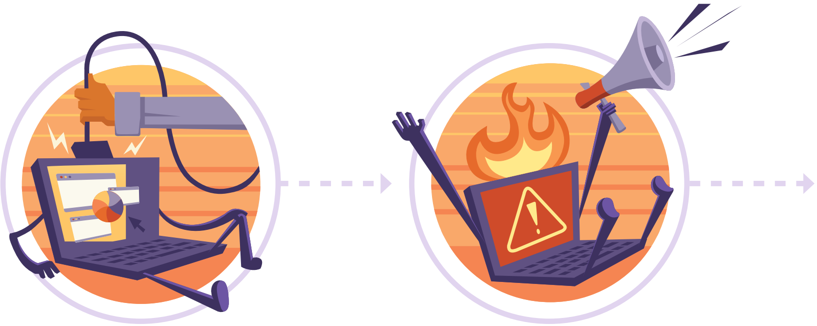
Deploy

Discard

Sentry unlocks workflow efficiency

Sentry is more than just error tracking. It’s how you maintain the quality of your code and work more efficiently with your team to improve the user’s experience. Merging Sentry into your existing workflow — specifically the alert, triage, and resolution phases — provides the insight to work at your highest efficiency.

> “Sentry is how we continuously iterate, support, and build, to make our product better and better.”
>
> — Lucas Willett, Engineering Manager of Observability
> Braintree (a PayPal service).



Alert

Sentry's notifications give immediate visibility into the impact of errors on your users and why you are being alerted. Instead of reacting to a series of problems run rampant, each developer on your team can proactively decide the appropriate next step, fitting iteration into the existing development workflow.

Sentry has SDKs that support most languages — virtually all of them, to be precise — and you'll want to set your project up with one of them first. After the SDK is installed and configured, you can:

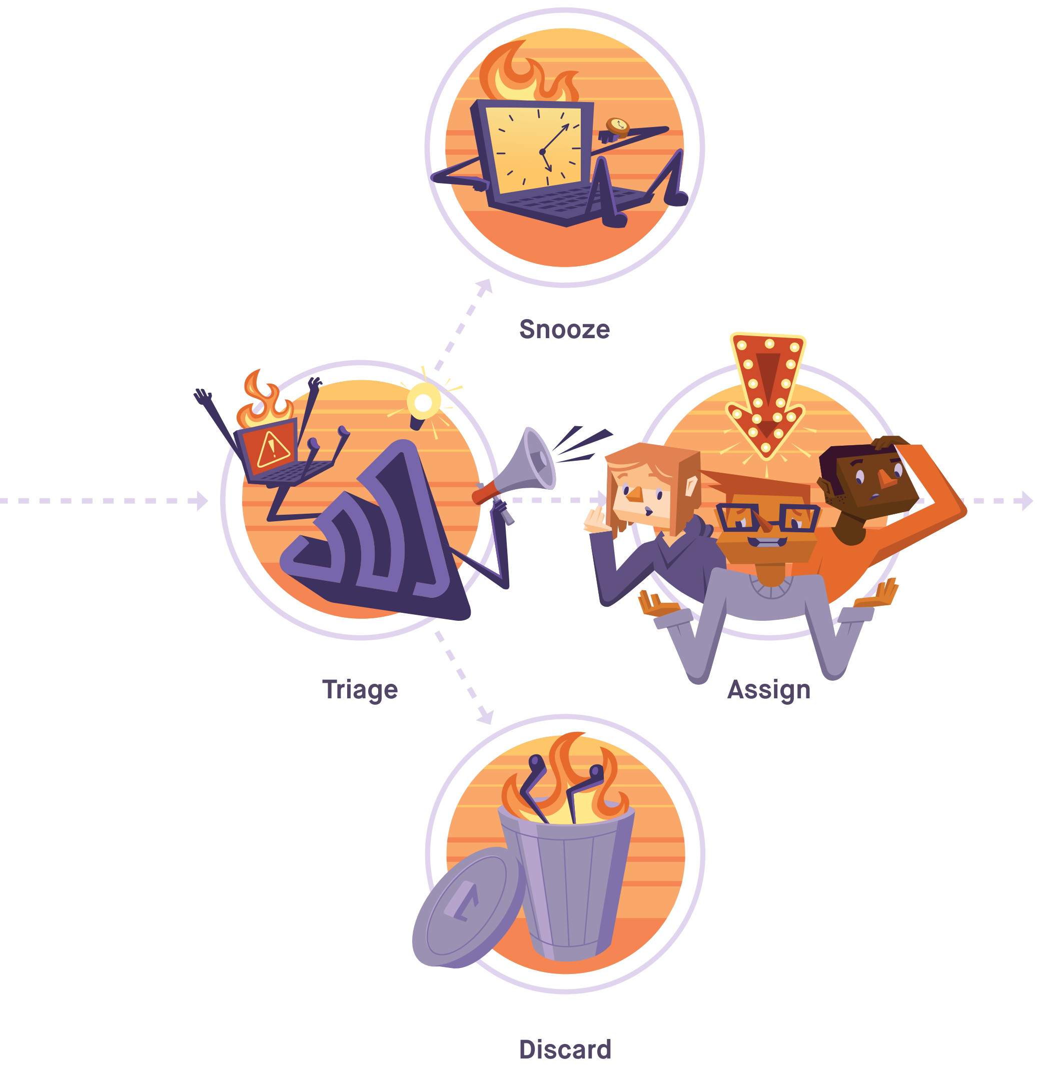
- See how many times each error occurred and how many users it affected with aggregated errors.
- Fix multiple instances at once and streamline the process of receiving notifications by grouping related errors by stack trace.

By customizing alert rules and integrating into the tools you already use, you can receive alerts when (and if) you want them, without disruption. Configure alerts according to your organization's workflow, stack, communication style, and objectives. Keep yourself organized by integrating with **Slack** and **Jira**.

Contextual alerts can lead to increased accountability for the customer's experience. Visibility into previously hidden issues means a more timely remediation process in your existing workflow *and* a much higher signal-to-noise ratio (plus happier developers).

> My favorite part of Sentry is collecting related errors into a single item. Seeing only my 30 issues out of thousands makes resolution easy. With Sentry, I can fix a million problems all at once.

— Andy Tuba, Sr. Software Engineer, Reddit



Triage

Alerts carry the information that transitions you into the triaging phase by including a link to an error's context.

Define the **issue state** upfront and expedite the triaging process by marking the error as resolved, unresolved, or muted. If there is an unactionable issue that continues to occur, you can **remove it** from the issue details page entirely.

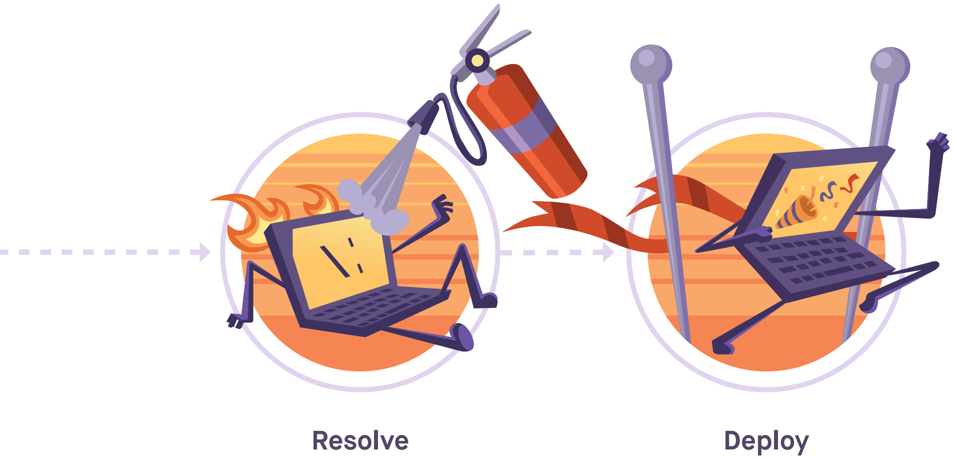
If the issue is actionable, triaging is easy as a few simple steps:

- Check out the **context** and stack trace to gather useful insights.
- Leverage tags to introduce a new level of issue impact specificity, including browsers, devices, **environments**, or **releases.** Create custom tags to fit your specific organization's needs, like custom type.
- Add **Breadcrumbs** for a better understanding of the trail of events that happened in the code before the issue.
- Filter to see if it's a **client-side** error or reproducible **server-side** error.

The key to an efficient workflow is knowing the *who and when*: the owner of an issue and where it occurred. Sentry's deep integrations with source code management platforms like GitHub, Azure DevOps, Bitbucket, and GitLab, expose the commits that likely introduced the error and **suggest a specific owner** who can most readily fix the problem.

Add another layer of relevance by setting rules in Sentry's UI based on URL or codepath to notify only the person or team with explicit knowledge of the affected code. Based on the rules that you define, you'll see an implicitly suggested assignee who matches those rules. Via the information in the suspect commit, you have more visibility around who should be responsible for the error and can assign it to them as an **issue owner**.

> There's no retry with transactions. Sentry's new Teams workflow is a key to our accountability-based dev culture.
>
> — Lucas Willett, Engineering Manager of Observability, Braintree



Resolve

The newly assigned issue owner now has the context to iterate on the code and commit the fix. **Resolve via commit** is a useful way for that engineer to stay organized and productive. Referencing the issue in the commit (e.g., Fixes ISSUE-ID-123) will mark it to resolve in that specific release and mute notifications for that issue before the fix deploys.

The deployment pipeline then picks up the change and deploys it. You can also configure **deploy emails** to send details on how many files have changed and what commits are being released once the code gets deployed.

Congratulations. You've just made iteration a seamless part of your existing development workflow. When done right, fixing bugs becomes a fluid part of the development process, just like building.

> "Sentry has changed our life. With the error context that Sentry captures, we're able to be alerted to and root out client-side errors immediately. Because of Sentry, we've been able to fix just about every broken deploy within minutes."

—Matin Movassate, CEO, Heap

> “Sentry is a game-changer. We use it not only to alert us of errors in our production app, but also issues in our command line tools and builds. Logs are for auditing. Sentry is for real-time automation. Be kind to yourself and use Sentry!”

— Dan LaMotte, Sr. Principal Software Engineer, Symantec

For more information, contact **sentry.io/contact/enterprise**