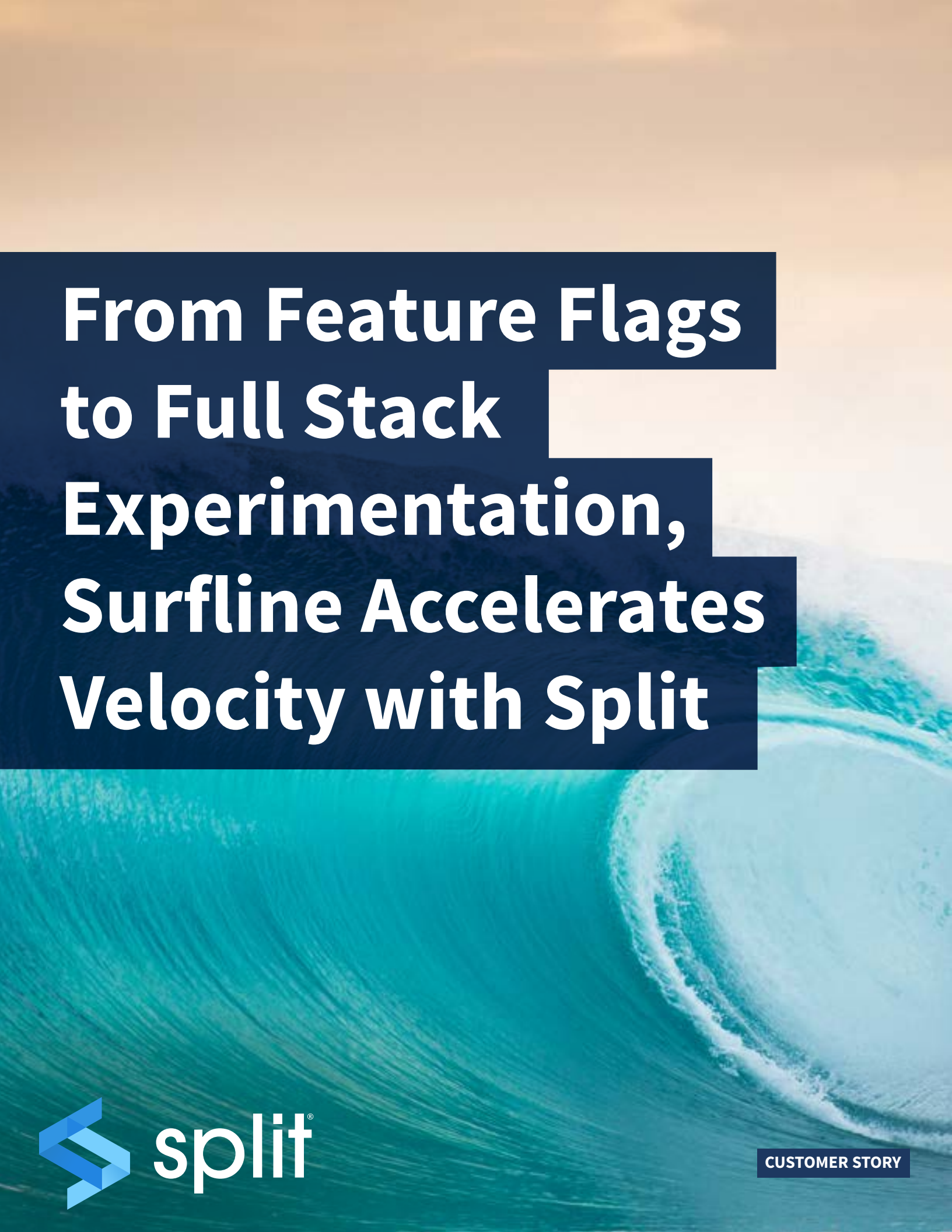


From Feature Flags to Full Stack Experimentation, Surfline Accelerates Velocity with Split

split®

CUSTOMER STORY

Business Background

The world's largest surf forecasting site looks to Split to help the engineering team scale. Split separates feature release from code deployment, enabling phased rollouts for major feature updates. And, Split enables the analytics team to experiment on feature changes anywhere in the codebase.

"Without Split, rollbacks could take a couple hours to a day. And after that, there might still be a lot of code cleanup. Split helped us increase our engineering velocity and code stability."

— Drew Newberry, Senior Software Engineering Manager

Data Points

- One half of a full-time software engineer's time regained from not needing to manage rollbacks and code cleanup.
- Minimal engineering time to configure feature flags.
- Split integration to Segment makes it easy to integrate Split impression data with other platforms such as Amazon RedShift.

Situation

As Surfline grew to become the world's largest surf forecasting site, with the largest marine camera network in the world, the product and engineering team building and maintaining the site grew as well. The team at Surfline starting using various approaches to handle the increasing scale of the team and the enhancements and new products Surfline was rolling out.

The team had been inching closer and closer to a continuous delivery model. Development was done on a master (i.e. trunk). Roughly twice a week release branches were cut off the master and put into staging for quality assurance (QA). If the release candidate passed all tests, it would then be pushed into production. A standard process that many modern engineering teams follow.

The issue was that these deployments were all or nothing. If a release candidate had some important fixes or new features that the team was under pressure to go live with, however it also introduced some new bugs, there was no way to choose only the desired code updates to be deployed in production.

The need to physically move bits around between staging and production introduced significant inefficiency. As Senior Software Engineering Manager Drew Newberry explains, "When landing really large code change sets, rollbacks could take a couple hours to a day. And after that, there might still be a lot of code cleanup." With deployments happening twice a week, this inefficiency could add up to the equivalent of half the time of a full time engineer. Valuable time that Surfline would much prefer to be using toward improving the site for their growing customer base.

In addition, these hurdles served as a constant brake on development velocity. Surfline has been replatforming from a monolith to microservices. In that process they have had very long feature branches that stayed open for a long time. They could not get early versions of them deployed in the production environment for feedback with real customer data. The right data to test the new services was not available in staging or testing environments.

Finally, the Surfline analytics team has advanced capabilities to do the right data analysis, however it was challenging to scale the statistical analysis to all feature releases. The team leverages data from a variety of sources and passes that data to a RedShift data warehouse via Segment. They use both off the shelf tools such as Google Analytics to conduct analysis as well as custom code written in-house by the analytics team. Yet with all these tools, it was a challenge to just have baseline measurement against KPIs at scale for every feature release.

Choosing Split

The team was familiar with feature flagging, and considered building their own in-house solution. One option was to build something basic that managed feature state through config values. As they were evaluating Split, they saw that not only could it handle their basic needs to separate feature release from code deployment, Split also brought additional functionality. The Split management console made it accessible for product managers to easily monitor and change the state of a feature rollout.

The granular targeting capability in Split made it possible to create different cohorts of users in a wide variety of ways. Without Split, this would have to be done programmatically on the back end.

For the analytics team, Split had a built-in integration to Segment, making it easy to push Split impression event data (i.e. the event record of what treatment of a feature a user experienced) to any analytics tool. And, Split has been designed for product experiments from the ground-up, working with the Surfline tech stack, unlike marketing A/B testing products.

“Split provided the platform and integrations we needed to run full stack experiments for key technical and customer experience questions,” - Julian Ganguli, Principal Data Analyst, Surfline

Result

Split was easily rolled out within days, and immediately provided a way to slowly release features and check how they are doing before exposing them to all users. Any larger feature initiative is put behind a feature flag. Generally if a code change is broad, and spans a number of stories, they put it behind a Split.

Today, Surfline is using Split to manage a migration to a new platform that is currently in beta. Once they are 100% on the new platform, they anticipate putting every code change behind Split before exposing it to all users.

All of the tech leads and most software engineers use Split. It is very simple to set up a feature flag (called a “Split”) in the system. That makes the team thrilled, since it provides critical functionality without taking engineers away from coding to manage the feature flagging system.

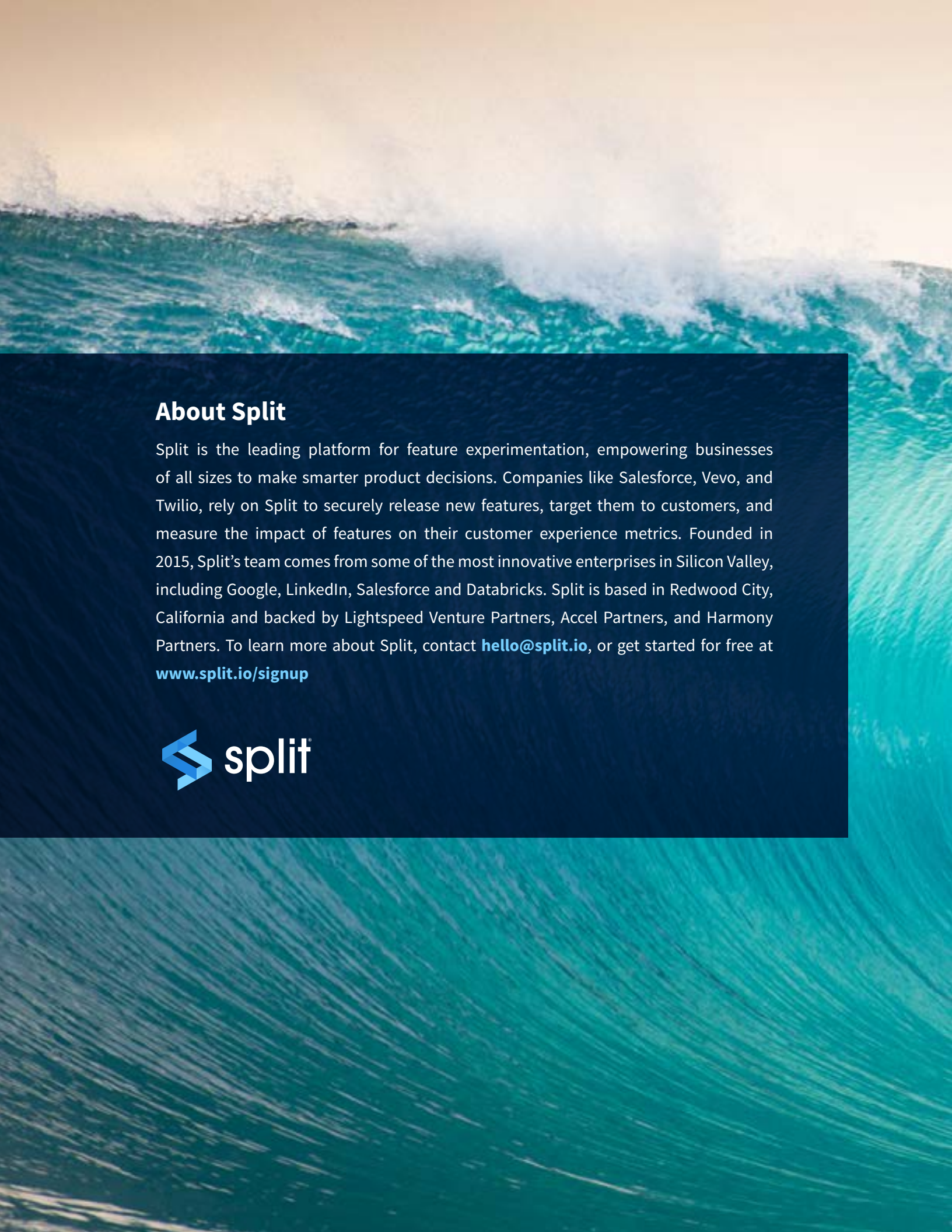
The analytics team, with the Segment integration, is able to analyze the result of a feature rollout or an experiment across any of their metrics. Split provides them with flexibility to run experiments on any feature release, as well as set up additional experiments.

To date, the team has been able to test parameters such as, if forced user registration impacted engagement across the site, or be able to test different options for how to merchandise, or present, products on the site. Additionally, they were able to test a new search results UX, and how it impacted click-through rates and engagement.

Future Plans

As Surfline looks to the future, the team would like to be able to experiment much more broadly. Today, the concepts of a feature release managed by engineering and an experiment managed by the analytics team are somewhat separate. However, the team sees this converging over time, as every feature release becomes an experiment. As Julian Ganguli, Principal Data Analyst states, “For everything we roll out, we want to have concrete data on how the change impacted our users and our business.”

With Split’s increased investment in feature experimentation and the Split Intelligent Results Engine, the team anticipates continuing to grow in how they leverage all that Split has to offer. Future possibilities include increasing usage of the platform to more incremental code changes behind Splits, and using the latest analytics functionality to measure the impact of feature releases against customer experience metrics.



About Split

Split is the leading platform for feature experimentation, empowering businesses of all sizes to make smarter product decisions. Companies like Salesforce, Vevo, and Twilio, rely on Split to securely release new features, target them to customers, and measure the impact of features on their customer experience metrics. Founded in 2015, Split's team comes from some of the most innovative enterprises in Silicon Valley, including Google, LinkedIn, Salesforce and Databricks. Split is based in Redwood City, California and backed by Lightspeed Venture Partners, Accel Partners, and Harmony Partners. To learn more about Split, contact **hello@split.io**, or get started for free at **www.split.io/signup**

split®