By extending the use of LaunchDarkly feature flags across teams, iPipeline recognized significant DevOps benefits as well as unforeseen operational and business improvements.

LaunchDarkly Feature Management Enables iPipeline to Drive Quality and Speed with a Safety Net

December 2021

Written by: Jim Mercer, Research Director, DevOps and DevSecOps, and Katie Norton, Senior Research Analyst, DevOps

Introduction

iPipeline is a leading provider of cloud-based software solutions for the life insurance and financial services industries. The company provides a SaaS-delivered solution that helps end customers accelerate and simplify insurance sales, compliance, operations, and support. iPipeline is headquartered in the United States, and of its 841 worldwide employees, nearly 70% are focused on application development or IT operations management. The iPipeline DevOps teams, which are geographically dispersed in the United States and the United Kingdom, support customer-facing applications with over 600,000 registered users globally.

While iPipeline was using DevOps CI/CD automation processes, the company still released software in the evening using a waterfall approach, lacking the confidence to roll out new application features incrementally during the day. New feature releases came out monthly and were deployed to a pre-production environment where iPipeline integrated its new releases with other development partners' software updates. This pre-production environment enabled testing of the new release before rolling out the software into the actual production environment.

iPipeline developed software on long-lived source code feature branches descended from the master branch. The updated feature branch was not merged back into the master codeline until the work was completed. Since the master codeline is a single shared codeline, it receives ongoing updates from other feature merges, so the underlying code is often different from when the original feature branch was created. As a result, application developers had to manually alter the master codeline to address source code merge conflicts. This process not only was time consuming but also introduced the risk of undetected merge-related software bugs late in the application development process.

SOLUTION SNAPSHOT

ORGANIZATION:
iPipeline is a provider of software solutions for the life insurance and financial services industries.

ORGANIZATIONAL CHALLENGES:
» Improve application software quality
» Move away from "big bang" releases
» Accelerate problem isolation and resolution
» Adopt progressive software delivery

SOLUTION:
LaunchDarkly feature flags

BUSINESS BENEFITS:
» Customer satisfaction and renewals
» Faster time to market
» Operational cost improvements
» Margin and profitability increases
» Improved employee morale and retention

The iPipeline DevOps teams wanted to be able to release more frequently using a blue-green progressive delivery technique. This style of application release allows DevOps teams to gradually transfer user traffic from a previous version of the application to a new release while both versions are concurrently running in production. The older version of the application is considered the blue environment, and the latest version is known as the green environment. However, there was not only a lack of adequate tooling but also a fear that a problematic update could disrupt the production environment without the ability to roll back the change swiftly.

Implementation

The iPipeline team had already been using DevOps for .NET application delivery and needed to incorporate LaunchDarkly feature flags into its hybrid stack of DevOps tools, which included the following: Jira for agile requirements and planning; GitHub and Microsoft Team Foundation Server (TFS) for source code repositories and versioning; Microsoft Teams to enable collaboration across the teams; Azure DevOps for CI/CD build, testing, and delivery; Checkmarx CxSAST to identify susceptible code and security vulnerabilities; Selenium and Tricentis SpecFlow for QA test automation and behavior-driven development (BDD); ActiveXperts and PRTG for production monitoring and observability; and Octopus Deploy for automated deployment and release management.

The implementation of LaunchDarkly feature flags was straightforward and did not require engagement with LaunchDarkly professional services or any other service providers to roll out the solution. iPipeline's engagement with LaunchDarkly was limited to a few support calls that were general knowledge related as iPipeline acclimated to feature flags and incorporated them into DevOps processes. The adoption of LaunchDarkly feature flags occurred across the iPipeline teams in a grassroots manner. The initial feature flag adoption started with some of the United States-based DevOps teams and then spread to the teams in the United Kingdom.

Some team members were apprehensive at first about making runtime changes, fearing they might break the production environment. To get more comfortable with the technology, the iPipeline DevOps teams put simple code changes behind LaunchDarkly feature flags as part of a hackathon. The hackathon events helped engender substantive production usage across teams and geographies.

Benefits

As anticipated, the LaunchDarkly feature flags considerably impacted how the iPipeline application developers built new application features. Since iPipeline could now release features incrementally using a blue-green style methodology, the company improved the frequency of releases from monthly to multiple times per week. The increased release frequency enabled faster customer feedback, which allowed DevOps teams to be more responsive to evolving customer needs. In addition, since all their updates were behind feature flags, they could deploy code confidently because they could recover from errant changes simply by turning off the associated feature flag. Further, features can be rolled out in a targeted manner such that the changes will be exposed to certain user groups and validated before being available across the entire application landscape. Consequently, the previous need to maintain long-lived feature code branches is eliminated as new code can be quickly integrated into the main codeline. This mitigates the need for merging dated code in the main codeline, avoiding the inherent risks and saving application developers precious time.

"LaunchDarkly feature flags enabled us to eliminate 'big bang' software releases."

The iPipeline team stressed that quality was a big driver for adopting LaunchDarkly feature flags; however, the team discovered that the flags also helped with QA test automation. The DevOps teams had already been aggressively improving their software test capabilities, with some applications reaching automation levels of 80%. Incorporating feature flags improved their test automation capabilities and added what they referred to as "smart automation." Automated tests now turn feature flags off and on using the LaunchDarkly APIs, enabling immediate problem isolation when a QA test fails.

While application development was the driving force behind deploying feature flags, extending usage of feature flags across QA and IT operations improved transparency and collaboration. For example, now QA testers understand where the code lives and have a line of sight into the GitHub pull requests, making them better equipped to collaborate with the application developers on building stronger test plans.

Business Benefits

The iPipeline team indicated that LaunchDarkly feature flags also helped increase the overall efficiency of the business. Although iPipeline customers are not aware they are using feature flags, the improved quality and agility feature flags provided helped improve overall customer satisfaction. In addition, the reduced lead time for software updates improved iPipeline customers' overall time to market. By reducing the toil of code merges, bug fixing, and off-hours releases, while enhancing collaboration among teams, iPipeline reduced the operational cost of delivering software. This also had the knock-on effect of creating a more pleasant work environment, improving employee morale and retention.

Future Plans

The iPipeline team plans to continue expanding its usage of the LaunchDarkly platform. Presently, it is using facets such as feature workflows, flagging rules, access controls, the .NET SDK, APIs, and the Microsoft Teams integration. The iPipeline team is interested in using platform capabilities such as experiments, the client-side SDK, and integrations with VS-Code and Jira. Further, the team is planning to expand the usage of feature flags within the company to include product marketing teams.

Methodology

LaunchDarkly identified iPipeline as a suitable candidate for an IDC Customer Case Study. IDC obtained the company and background information in this document through an extensive interview with the iPipeline team based in the United Kingdom. LaunchDarkly was not involved in the interview process. All the questions were posed directly to the iPipeline team by the IDC analysts.

About the Analysts

Jim Mercer, *Research Director, DevOps and DevSecOps*

Jim Mercer is a Research Director within IDC's DevOps Solutions research practice. In this role, he is responsible for researching, writing, and advising clients on the fast-evolving DevOps market. Mr. Mercer's core research includes topics such as rapid enterprise application development, modern microservice-based packaging, application security, and automated deployment and life-cycle/management strategies as applied to a DevOps practice. In addition, he examines how the move to DevOps methodologies impacts enterprise use of open source and preferences for using on-premises computing and development platforms versus public cloud services. He looks at how organizations are prioritizing DevSecOps and using automation to insert security assessments at the beginning of the DevOps delivery pipeline (i.e., shift left). Mr. Mercer advises senior IT, business, and investment executives globally in the creation of strategy and operational tactics that drive the execution of digital transformation and business optimization.

Katie Norton, *Senior Research Analyst, DevOps*

Katie Norton is a Senior Research Analyst within IDC's DevOps Solutions research practice. In this role, she is responsible for researching, writing, and advising clients on the fast-evolving DevOps market. Katie's core research includes topics such as rapid enterprise application development, modern microservice-based packaging, application security, and automated deployment and life-cycle/management strategies as applied to a DevOps practice. In addition, she examines how the move to DevOps methodologies impacts enterprise use of open source and preferences for using on-premises computing and development platforms versus public cloud services.

MESSAGE FROM THE SPONSOR

Fundamentally change how you deliver software. Software powers the world. But even in the Cloud and DevOps era, the risk, complexity and fear of launching new features bottlenecks innovation and grinds engineering efforts to a halt. LaunchDarkly fuels developer productivity and creativity by transforming how teams deliver software to customers. With the ability to progressively release features to any segment of users on any platform, organizations can standardize safe releases at scale, accelerate their journey to the cloud, and foster better collaboration between development and business teams. LaunchDarkly serves trillions of flags every day and provides critical development infrastructure for some of the world's most recognized brands. Learn why over 20% of the Fortune 100 choose LaunchDarkly for feature management.

IDC Custom Solutions