DevOps: Anger Management 101

https://media.giphy.com/media/87b6lrxvf4uJIL1zl3/giphy.gif

So far, we have heard of **Continuous Integration**, **Continuous Delivery**— even **Continuous Deployment.** These processes are steering DevOps forward in a blaze of glory.

However, this industry's future requires solutions that will make the deployment process even more progressive.

Progressive Delivery

For those ‘*outside the loop,*' progressive delivery allows you to ship **feature flags** in an “off” mode. This way, you can gradually release parts of a new software update, having full control of when to stop or continue.

This approach provides an effective and fast way to exert more delicate control over delivery. Also, progressive delivery makes the rollout of new functionality and testing in a production environment attainable **without disrupting the whole system**.

https://media.giphy.com/media/I2Sq5GffrCyUMEXjW/giphy.gif

Progressive Delivery in Development

The progressive delivery exercise can’t be tested on a whim. It comes complete with a set of practices fit for an application development environment. Below are the major pillars of progressive delivery.

Canary in the Coal Mine

You didn’t think we’re talking about caged birds here, did you? This is just an origin phrase for canary testing.

Unlike the gruesome mining tradition, canary testing helps developers **quickly detect and resolve any issue** with new software ***before*** it degrades the user experience.

Once a new release is ready, it gets deployed to one of the *two identical environments*. The main purpose is to route a small percentage of users to this canary release. This way, *only a small number of users get impacted if any bug is rolled out*.

Blue-Green Deployment

In the DevOps universe, **blue-green deployment** is a magic technique that helps release bug-free software while decreasing application downtime and rolling out updates as needed.

It is done by transferring user traffic from an old app version to an almost identical new release in *two production environments*.

A/B Testing

With split testing, we can put ourselves in charge and make decisions based on accurate data about how users actually behave when they interact with your product.

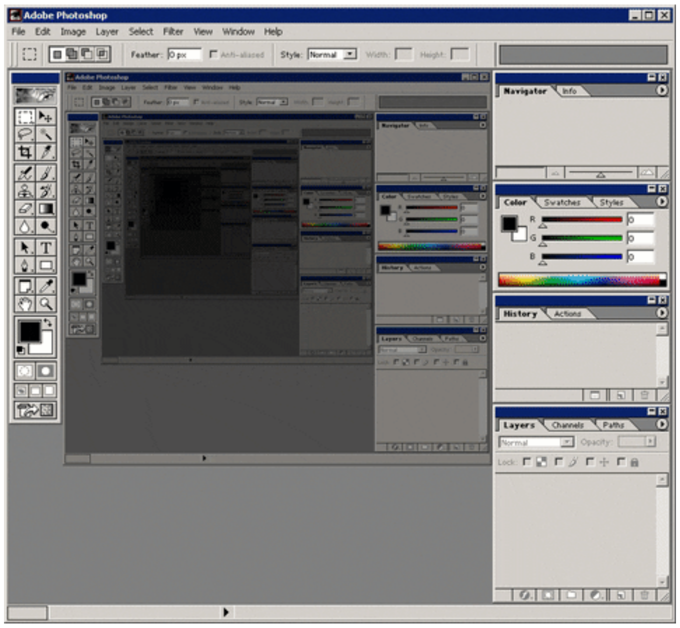
Essentially, it’s about running a simultaneous experiment between two or more variants of a page to see which scores higher.

Other building blocks of progressive delivery include the famous *feature flagging technique, the power of observability, and the art of chaos engineering*. These joined forces pave the way for pushing changes to a product iteratively, introducing progressive and controlled delivery.

The Future of Software Development?

In general, Progressive Delivery can be coined as a practice bridging the gap between user’s and developer’s experience. It comes in as the next step after the CI/CD methodology, as a modified version of continuous delivery.

Feature toggles are already setting the stage for innovative **software development** and delivery practices with supreme support and control. It seems like the bright future of controlled delivery is just around the corner.



Join us today to thank our newsletter sponsor, LaunchDarkly. LaunchDarkly is the #1 platform for managing Feature Flags. Feature Flagging is an industry best practice that allows for the faster and smarter deployment of code. This makes life easier — and less frustrating —for both an application's users and developers. With the ability to strategically control the internal process of releases, your team can be confident with deploying new features.

***

Got a tech story to share with our readers? Everything you've ever wanted to know about how to get published on Hacker Noon - **get it here**