sonatype

Resources > Blog > The Three Ways of DevOps: A principle-based DevOps framework

The Three Ways of DevOps: A principled framework

September 16, 2019 By Ember DeBoer

5 minute read time

In our first two articles in the DevOps track of Educational Foundations, we:

- Dispelled some DevOps myths and presented our own principle-based definition of DevOps
- Presented the business case for DevOps principles, including the problems movements like Lean and Agile attempt to solve, as well as metrics from some of the latest research on high-performing DevOps organizations

In this article series, we'll discuss the following three principle-based DevOps frameworks, below, as well as the common themes we can take from them to apply to your own DevOps transformation:

- The Three Ways of DevOps as described in *The Phoenix Project* and *The DevOps Handbook*
- Mature capabilities in technical and management practices found in high-performing DevOps teams, based on the research presented in *Accelerate*
- The CALMS (Culture, Automation, Lean, Measurement, Sharing) framework for assessing DevOps

Framework #1: *The Phoenix Project/ DevOps Handbook's* Three Ways

If you've read either *The Phoenix Project* or *The DevOps Handbook*, you've been introduced to The Three Ways of DevOps framework:

- The First Way: Principles of Flow
- The Second Way: Principles of Feedback
- The Third Way: Principles of Continuous Learning

The First Way: Principles of Flow

The First Way of the DevOps Three Ways is mostly concerned with accelerating the "flow" of work throughout a process. Gene Kim also refers to the First Way as Systems Thinking in his article The Three Ways: Principles Underpinning DevOps. Whether you're calling it Flow or Systems Thinking, the DevOps principles underpinning the First Way are working toward the same end: Viewing the flow of work as one continuous system (unsiloed) that can be refined and optimized in line with a continuous improvement culture.

Some of the key DevOps best practices of the First Way are:

- Making work "visible." Unlike manufacturing processes, which are easily observable on a plant floor, the flow of software through its development lifecycle is not easily seen. Using methods such as Kanban boards can surface the activities going on behind the scenes, by showing the left-to-right movement of a user story through the development phases.
- Limiting work-in-progress (WIP). Keeping work-in-progress to a minimum has also been shown to accelerate work flow, because it minimizes multi-tasking and context-switching.
- Reducing batch sizes. "Chunking" work into smaller pieces like a two-week sprint can also help deliver features (albeit smaller ones) and bug fixes to the customer faster. Issues are often caught earlier when those updates and additions are released sooner.
- Reducing hand-offs between teams. The risk of "dropping the baton" increases as the hand-offs do. Although hand-offs can't be completely minimized, the key is to keep the teams in tight communication with one another so that the hand-off itself is almost a non-event rather than a large ordeal with the potential for communication missteps along the way.
- Identifying and removing constraints and waste. Constraints might be bottlenecks in the process, such as environments, test setup, and overly tight architecture, while waste includes things like manual work, heroics, and context-switching.

The Second Way: Principles of Feedback

The Second Way works to enable a DevOps transformation through fast and constant feedback cycles at all stages of a development cycle.

Some of the key DevOps principles of the Second Way are:

- Swarming and solving problems to build new knowledge. This principle fits into the "fail fast" mentality, so that teams can find issues with an implementation as soon as

teams can find issues with an implementation as soon as possible and address them early and often as iterations continue.
- Pushing quality closer to source. This principle is at the core of the DevSecOps movement, which is concerned with addressing security concerns during the development cycle, instead of at the end, when rework to remediate is more difficult and costly.
- Optimizing for downstream work centers. This principle works against the "throw it over the wall" mentality, by underscoring that development should be just as invested in their application being deployable, working with operations to bridge that gap (and vice versa).

The Third Way: Principles of Continuous Learning

The Third Way of DevOps seeks to create a culture of continual learning and experimentation within the development organization.

Some of the key DevOps principles of the Third Way are:

- Enabling organizational learning and a safety culture. Leaders must help "set the tone" for the organization, making it okay to learn, make mistakes, and try again.
- Institutionalizing a continuous improvement culture for daily work. Improving what you do and how you accomplish it should be part of everyone's daily thinking and call to action.
- Transforming local discoveries to global improvements. Surfacing and sharing improvements at all levels will help enable a "bubble up" culture of continuous improvement.
- Injecting resilience patterns into daily work. Some examples might include rehearsing failures, and working toward improving key metrics for deployment.
- Leaders enforcing a learning culture. Organization-wide learning is unlikely to take hold and become pervasive unless it is sanctioned and exemplified by its leaders. So being intentional about communicating the value of learning and problem-solving is crucial to building that culture.

Next we'll look at

- Mature capabilities in technical and management practices found in high-performing DevOps teams, based on the DevOps best practices research presented in *Accelerate*
- The CALMS (Culture, Automation, Lean, Measurement, Sharing) framework

This post was originally published as a Sonatype Guide, a part of our Sonatype Community. The Sonatype Community is a place where you can ask questions to other Sonatype Nexus Repository users and the Sonatype team. It also provides content and learning paths from a team of experts that help make using Sonatype Nexus Repository even easier. If you haven't spent time there, I definitely recommend it.

Sources:

- The Phoenix Project: A Novel About IT, DevOps, and Helping Your Business Win by Gene Kim, Kevin Behr, & George Spafford
- The Three Ways: The Principles Underpinning DevOps by Gene Kim

Written by **Ember DeBoer**

Ember is Senior Tech Content Developer at Sonatype.

TAGS

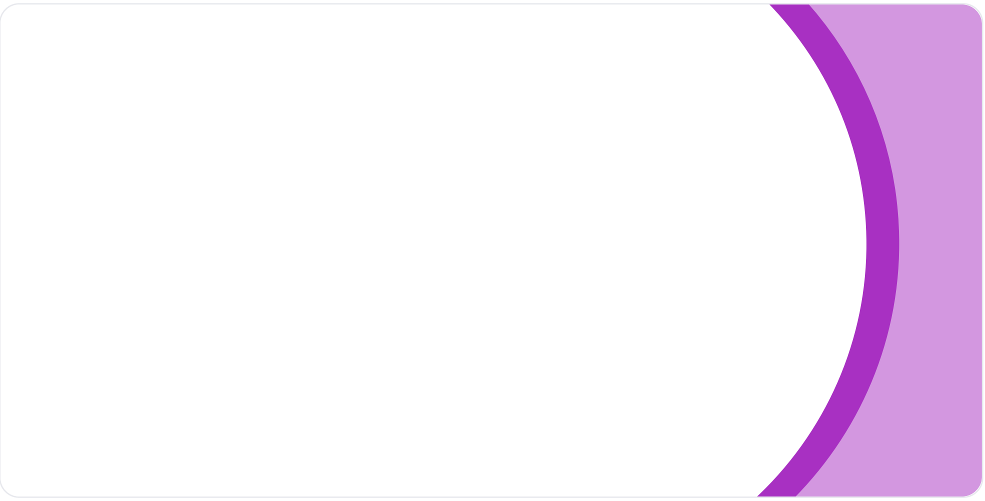
DEVOPS BEST PRACTICES DEVOPS DEVSECOPS

DEVOPS ADOPTION DEVOPS MATURITY DEVOPS FRAMEWORKS

SONATYPE NEXUS REPOSITORY

SHARE

Related Resources

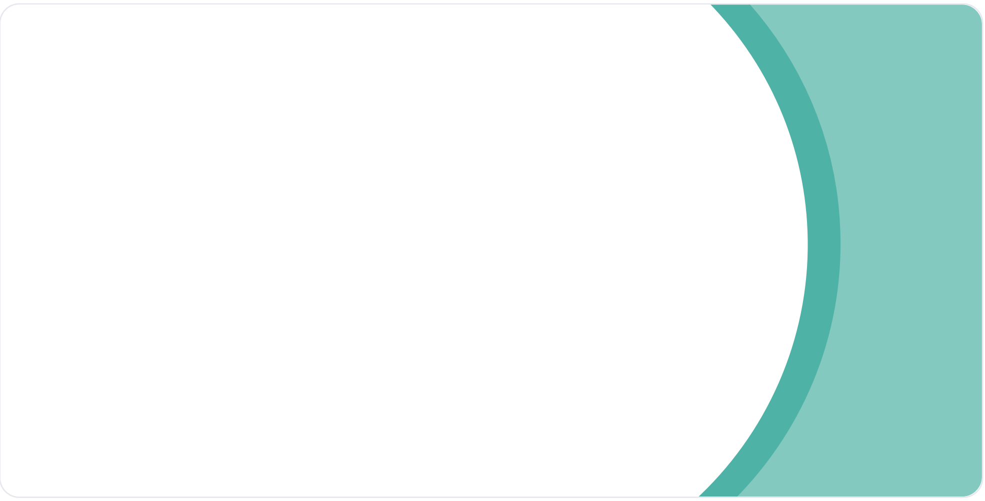


GUIDE

Comprehensive Comparison Guide: Sonatype vs. JFrog

BLOG POST

Sonatype welcomes Antoine Harden as Regional Vice President of Federal



BLOG POST

Elevate your organization's success: Submissions now open for the 2025 Sonatype Elevate Awards

sonatype

PLATFORM

Overview

Repository

Pricing

Firewall

SBOM Manager

Lifecycle

Integrations

AI/ML

SOLUTIONS

Application Security

Developers

Integrated Innovation

Legal & Compliance

COMMUNITY

All Day DevOps (ADDO)

Nexus Repository Community Edition

Sonatype OSS Index

RESOURCE CENTER

Blog

Whitepapers & eBooks

Log4j Updates

Customer Stories

Articles

Webinars

PARTNERS

Amazon Web Services (AWS)

Partner Program

Become a Partner

Find a Partner

Red Hat OpenShift

CUSTOMER PORTAL

Documentation

My Sonatype

Customer Support

Training & Workshops

COMPANY

Careers

Contact

About

Newsroom

Trust Center

Investors

Press Kit

SUBSCRIBE FOR ALL THE LATEST SOFTWARE SECURITY NEWS AND EVENTS

Subscribe

Terms of Service

Privacy Policy

Modern Slavery Statement

Event Terms and Conditions

Do Not Sell My Personal Information

Cookie Preferences