

**SECOND
EDITION**

Bonus Case Studies

The

DevOps Handbook

**HOW TO CREATE WORLD-CLASS
AGILITY, RELIABILITY, & SECURITY
IN TECHNOLOGY ORGANIZATIONS**

**GENE KIM,
JEZ HUMBLE,
PATRICK DEBOIS,
& JOHN WILLIS**

Featuring new foreword
and updated material by
NICOLE FORSGREN, PhD

Original foreword by
JOHN ALLSPAW



25 NW 23rd Pl
Suite 6314
Portland, OR 97210

The DevOps Handbook © 2016 by Gene Kim, Jez Humble, Patrick Debois, and John Willis.
The DevOps Handbook, Second Edition © 2020 by Gene Kim, Jez Humble, Patrick Debois,
John Willis, and Nicole Forsgren, PhD.

All rights reserved. For information about permission to reproduce selections from this guide, write to:
Permissions, IT Revolution Press, LLC, 25 NW 23rd Pl,
Suite 6314, Portland, OR 97210.

Cover and design by D. Smith Creative

ness into the stores that ordinarily occurred with every seasonal change—decisions that were made in these daily action meetings and fueled a broader enterprise mindset.

More recently, Target has rolled out a refreshed culture framework, which reminds the entire organization of what it takes to reinvest in the business and each other as they “care, grow, and win together.” This framework promotes connection, inclusivity, and drive, and encourages team members to “choose progress over perfection.”

Moving forward, Target sees room to grow in maturing product thinking, investing in technology talent to help drive development, and continuing to emphasize the principles of DevOps in the organization.

Iterative Enterprise SRE Transformation at Vanguard (2016–2021)

Vanguard, a global asset management company, began to adopt SRE best practices to make their DevOps teams more effective around 2016, and shared their experiences at the 2021 DevOps Enterprise Summits.

At the time Vanguard began its transformation, they had not yet begun public cloud migration. All monolithic applications were hosted in a privately owned data center. All deployments were released on a quarterly schedule by deployment and operations teams—not development teams. These deployments were monitored via an “alert-only visibility” policy, under which it was assumed that if there were no alerts, the application was up and running. Ownership of the alerts was centralized. In order to get an alert configured, the application team had to submit a request to a central team and wait for them to have a spare cycle to set up an alert.



Private Data Center



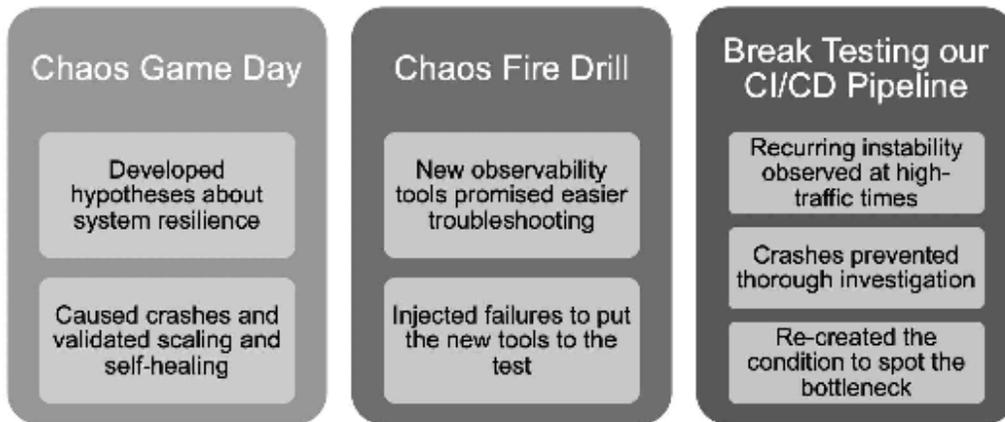
Controlled deployments



Alert-only visibility



Development and Operations



In a unique process that blurred the line between chaos experimentation and performance testing, Vanguard also performed brake testing on the CI/CD pipeline. As a large IT organization, they ran into growing pains as they were onboarding microservices into the CI/CD pipeline, observing recurring instability during high-traffic times, in which crashes were preventing thorough investigations because they were wiping out the critical logs before they could be offloaded to log-aggregation tools.

To troubleshoot this issue, they performance-tested the pipeline over a weekend by creating builds and deployments that recreated specific resource-intensive conditions, recreating crashes while someone watched the log file be created and then captured the relevant log files and thread dumps, allowing them to identify and address the bottleneck. By the following Monday, they saw immediate improvements to the pipeline.

Vanguard also had to find the right tools for their observability journey. “Alert only” visibility led to legacy alert consoles that were still almost exclusively used by operations teams. As microservices were carved out, the key benefit of the PaaS was that all the applications were operating in the same containerized environment. Their logs were filtering in the same ways and into the same places, so they were able to create standard microservice platform dashboards, originally intended for use by the platform owners.

