DevOps distilled, Part 1: The three underlying principles

Gene Kim

January 22, 2013

In this series of articles, learn about DevOps and how it can: create a collaborative relationship between development and IT operations; enable high deployment rates; and increase the reliability, resilience, and security of your production environment. In this first article, get a history of DevOps, explore the three underpinning principles of the DevOps movement, and learn how DevOps complements existing frameworks such as Agile and ITIL.

View more content in this series

Introduction

I’ve been studying high-performing IT organizations since 1999. After benchmarking over 1500 high-performing IT organizations, I found that they were 5 to 7 times more productive than their non-high-performing peers. This journey has taken me into the heart of the DevOps movement, which is undoubtedly one of the most exciting developments in the past 30 years in how IT organizations do their work.

DevOps typically refers to the emerging professional movement that advocates a collaborative working relationship between development and IT operations. DevOps results in the fast flow of planned work (for example, high deploy rates) while simultaneously increasing the reliability, stability, resilience, and security of the production environment. Development and IT operations typically represent the value stream between the business (where requirements are defined) and the customer (where value is delivered).

The origins of the DevOps movement took place around 2009 during the convergence of numerous adjacent and mutually reinforcing movements:

DevOps for Dummies

Today’s fast-moving world makes DevOps essential for any business aspiring to be agile and lean to respond rapidly to changing customer and marketplace demands. This book helps you understand DevOps and explains how your organization can gain real business benefits from it. You’ll also discover how a holistic view of DevOps that encompasses the entire software delivery life cycle—from ideation and the conception of new business capabilities
• The Velocity Conference movement, especially the seminal 10+ Deploys Per Day: Dev and Ops Cooperation at Flickr presentation (see Resources).
• The "infrastructure as code" movement (Mark Burgess and Luke Kanies), the "Agile infrastructure" movement (Andrew Shafer), and the Agile system administration movement (Patrick DeBois).
• The Lean Startup movement (Eric Ries).
• The continuous integration and release movement (Jez Humble).
• The widespread availability of cloud and platform as a service (PaaS) technologies (for example, Amazon Web Services).

DevOps and Agile

Many people wonder how DevOps differs from Agile development. One tenet of the Agile development process is to deliver working software in smaller and more frequent increments as opposed to the "big bang" approach of the waterfall method. This is most evident in the Agile goal of providing potentially shippable features at the end of each sprint—typically every two weeks.

High deployment rates will often result in work to be deployed piling up for IT operations. Clyde Logue, founder of StreamStep, is attributed with saying:

"Agile was instrumental in development regaining the trust in the business, but it unintentionally left IT operations behind. DevOps is a way for the business to regain trust in the entire IT organization as a whole."

DevOps is especially complementary to the Agile software development process. DevOps extends and completes the continuous integration and release process by ensuring that code is production ready and will provide value to the customer.

DevOps enables a far more continuous flow of work into IT Operations. When code is not promoted into production as it is developed, deployments will accumulate for IT operations. If development delivers code every two weeks but it's deployed only every two months, customers don't get value and the deployments often result in chaos and disruption.

There is an inherent cultural change component of DevOps; it modifies the flow of work and local measurements of development and IT operations.

DevOps and ITIL/ITSM

Although many people view DevOps as backlash to IT Infrastructure Library (ITIL) or IT Service Management (ITSM), I take a different view. ITIL and ITSM still are best codifications of the business processes that underpin IT operations. They actually describe many of the capabilities needed for IT operations to support a DevOps-style work stream.

Agile, and continuous integration and release, are the outputs of development, which are the inputs into IT operations. To accommodate the faster release cadence associated with
DevOps, many areas of the ITIL processes require automation, specifically around the change, configuration, and release processes.

The goal of DevOps is not just to increase the rate of change. It can also successfully deploy features into production without causing chaos and disruption of other services, while quickly detecting and correcting incidents when they occur. This brings in the ITIL disciplines of service design and incident and problem management.

**Underlying principles of DevOps**

*The Phoenix Project: A Novel About IT, DevOps, and Helping Your Business Win* (see Resources) describes the underpinning principles of DevOps, from which all the DevOps patterns can be derived, as "The Three Ways." They describe the values and philosophies that frame the processes, procedures, practices, and the prescriptive steps. Figure 1 shows The First Way, which is systems thinking.

**Figure 1. The First Way**

The First Way emphasizes the performance of the entire system, as opposed to the performance of a specific silo of work or department. The work can be as large as a division (development or IT operations) or as small as an individual contributor (a developer or system administrator).

The focus is on all business value streams that are enabled by IT. It begins when requirements are identified by the business or IT, are built in development, and then transitioned into IT operations where the value is then delivered to the customer in the form of a service.

The outcomes of The First Way include:

- Never passing a known defect to downstream work centers
- Never allowing local optimization to create global degradation
- Always seeking to increase flow
- Always seeking to achieve profound understanding of the system (per Deming)

The Second Way, as shown in Figure 2, is about creating the right to left feedback loops. With almost any process improvement initiative, the goal is to shorten and amplify feedback loops so necessary corrections can be continually made.
Figure 2. The Second Way

The outcomes of The Second Way include:

• Understanding and responding to all customers, internal and external
• Shortening and amplifying all feedback loops
• Embedding knowledge where you need it

The Third Way, shown in Figure 3, involves creating a culture that fosters two things:

• Continual experimentation, which requires taking risks and learning from success and failure
• Understanding that repetition and practice are the prerequisites to mastery

The two items above are equally important and necessary. Experimentation and risk-taking are what ensure that you keep pushing to improve, even if it means going deeper into the danger zone than you've ever gone. And, you need mastery of the skills that can help you retreat out of the danger zone when you've gone too far.

Figure 3. The Third Way

The outcomes of The Third Way include:

• Allocating time for the improvement of daily work
• Creating rituals that reward the team for taking risks
• Introducing faults into the system to increase resilience

Conclusion

In this first part of the series, you learned a bit about the history of DevOps, the three underlying principles of the DevOps movement, and how DevOps complements existing frameworks.

Next, continue to Part 2, which discusses a specific DevOps pattern that ensures that code and environments work together in Sprint zero and lead to predictable and secure releases.
Read more on the history of DevOps, as told by two people instrumental in catalyzing the movement: John Willis and Damon Edwards. You can also read more in *The Phoenix Project: A Novel About IT, DevOps, and Helping Your Business Win*.