DevOps – The Basics

Contents

This document is derived with permission from multiple sources including the DevOps Institute's DevOps Foundation course and other sources as noted.

What is DevOps and what are its Origins?

Over the past ten years, a number of technological advancements and process improvements have enabled IT organizations to improve their performance and increase their velocity. Often, however, siloed teams, bloated processes and disjointed tools limit improved performance and prevent the business from benefiting.

DevOps is a cultural and professional movement that stresses communication, collaboration and integration between software developers and IT operations professionals. The resulting improved workflow provides businesses the flexibility to change, and change quickly, without sacrificing the quality and reliability of their IT-based business services.

> DevOps provides companies a competitive advantage by delivering better software, faster and by enabling sustained innovation.

Despite the name, DevOps extends beyond software developers and IT operations professionals. Generally speaking, 'Dev' represents all the people involved in developing software products and services (including business representatives and suppliers) and ‘Ops’ includes all the people involved in delivering and managing those products and services (including suppliers).

The term ‘DevOps’ was popularized during a series of DevOps Days starting in 2009 in Belgium. Since then, DevOps-related events worldwide, along with an active online community of practice, have enabled the spread of this experience-based movement. This community of practice is dedicated to studying and sharing practices and technologies that enable the rapid development and deployment of quality software products and services.

DevOps enablers include:

- Agile and lean software development practices
- Agile and lean service management practices
- Virtualized and cloud infrastructure from internal and external providers
- Treating infrastructure as code
- Data center automation and configuration management tools
- Monitoring and self-healing technologies

In recent years, Dev and Ops have each taken steps to use these enablers to improve their performance. Unfortunately, Dev and Ops have not always worked together on these initiatives. As a result, constraints arise and the cadence – or flow – of work is impacted. Conflicting goals further compound the problem. Dev is incented to generate change, rapid change, and more change. Ops is tasked with maintaining stability and often does so by inadvertently slowing down the pace of change. The resulting standoff leads to a dysfunctional culture and missed business goals. DevOps recognizes the need for cultural improvements and shared goals that are aligned with business goals.

DevOps Business Value

Today’s business environment is fast moving, competitive and technology driven. To thrive, enterprises must do more than simply leverage technology; they must use technology to enable constant and strategic innovation.

DevOps recognizes that traditional approaches to software development and IT operations are not meeting this accelerated demand for IT services. The more iterative, incremental and streamlined approaches introduced by agile and lean practices are needed. DevOps also recognizes that advancements such as cloud computing and smart mobile devices have prompted a paradigm shift in terms of how IT services are developed and delivered. In turn, these advancements have shifted business and customer expectations both in terms of how quickly IT services are developed and delivered and in the reliability of those services.

> Companies with high IT performance are twice as likely to exceed their profitability, market share and productivity goals.

DevOps responds to the demands of business and customer stakeholders for increased agility and stability by increasing overall IT performance. According to the 2014 State of DevOps report, companies with high IT performance are twice as likely to exceed their profitability, market share and productivity goals. This same study found that the longer DevOps practices are in place and continuously improved, the better the organization performs. Strong IT performance is a competitive advantage.

DevOps Perspectives and Values

Defining the term DevOps is easier said than done because there are several complimentary perspectives to consider.

- Senior IT management views DevOps as an effort to increase the overall efficiency of the IT department by streamlining how everyone works together for the benefit of the business.
- The business wants the IT department to meet two simultaneous business goals: (1) deliver high-quality software more quickly and (2) provide stable, reliable and secure IT services to customers.
- Developers, particularly those using agile development methodologies, talk about DevOps in terms of a continuous flow of delivery into production, potentially several times a day.
- Operations professionals view DevOps as promoting a more effective relationship with development teams and as an opportunity to increase the use of automation, self-service and proactive monitoring practices.
- Operations, information security and support professionals, such as service desk professionals, see DevOps as an opportunity to be engaged earlier in the development lifecycle to ensure their non-functional requirements are understood and are being met.
- Support professionals, given their proximity to the company’s employees and/or customers, want to ensure that DevOps practices make it easy to pass along customer feedback and to (as needed) escalated incidents and change requests once a solution is released.

Given these varying viewpoints, are there values all DevOps stakeholders can agree upon? There are. After the first US-based DevOps Days in Mountain View California (2010) John Willis and Damon Edwards coined the acronym **CAMS** (Culture, Automation, Measurement and Sharing) in an effort to identify values that are the essence of the DevOps movement. Jez Humble later added an L, standing for Lean, to form CALMS.

CAMS - Culture, Automation, Measurement and Sharing

- **Culture** – Culture relates to the people and process aspects of DevOps. Organizations need to ensure they have 'just enough' process in place to enable people to effectively communicate and collaborate. Without the right culture, automation attempts will be fruitless.

- **Automation** – Technologies such as release management, configuration management, and monitoring and control tools that enhance flow and enable automation are important aspects of DevOps.

- **Measurement** – There's an old adage that 'if you can't measure it, you can't manage it.' It can also be said that 'if you can't measure it, you can't improve it' – A successful DevOps implementation will measure everything – people, process and technology performance.

- **Sharing** – Sharing is the feedback loop in the CAMS cycle. Creating a culture where people share ideas and problems is critical not only because it enables improved communication and collaboration but also because it helps organizations to improve.

DevOps principles and practices bring the CAMS values to life and in doing so:

- Respond to the business' accelerated demand for quality software products and services
- Acknowledge the interdependence of IT functions (*all* IT functions)
- Recognize the need for cultural improvements
- Support and leverage agile, lean and IT service management (ITSM) practices
- Encourage the use of automation
- Require a commitment to continuous improvement through the use of performance metrics and sharing

DevOps Principles and Practices

DevOps is not *just* about culture or *just* about automation. All of the CAMS values enable organizations to bring DevOps principles and practices to life.

DevOps Principles – The Three Ways

The Three Ways are introduced in '*The Phoenix Project: A Novel About IT, DevOps, And Helping Your Business Win*' by Gene Kim, Kevin Behr and George Spafford. The Three Ways are an effective way to frame the processes, procedures and practices of DevOps, as well as the prescriptive steps.

- **The First Way – Flow**
    - Understand and increase the flow of work (left to right)
- **The Second Way – Feedback**
    - Create short feedback loops that enable continuous improvement (right to left)
- **The Third Way – Continuous experimentation and learning**
    - Create a culture that fosters
        - Experimentation, taking risks and learning from failure
        - Understanding that repetition and practice is the prerequisite to mastery

DevOps Practices

DevOps isn't a framework or methodology in and of itself. It doesn't stand alone. DevOps adopts and leverages multiple frameworks and methodologies such as agile, lean and ITSM.

DevOps has benefitted tremendously from the work the Agile community has done, showing how small teams, operating with high-trust, small batch sizes with smaller, more frequent software releases, can dramatically increase the productivity of development organizations. DevOps applies Lean principles such as increasing flow and reducing waste to the IT value stream. DevOps requires agile service management processes to remove bottlenecks and achieve faster lead and cycle times.

By adopting and adapting practices from multiple frameworks we generate more productivity and economic value for the business.

DevOps practices support and enable The Three Ways.

The First Way – Flow

- Practices include (but are not limited to):
    - **Continuous integration** – a development practice that requires developers to integrate code into a shared repository on a daily basis
    - **Continuous delivery** – a methodology that focuses on making sure software is always in a releasable state throughout its lifecycle
    - **Continuous deployment** – a set of practices that enable every change that passes automated tests to be automatically deployed to production

  - **Value stream mapping** – a lean tool that depicts the flow of information, materials and work across functional silos with an emphasis on quantifying waste, including time and quality
  - **Kanban** – a method of work that pulls the flow of work through a process at a manageable pace
  - **Theory of constraints** – a methodology for identifying the most important limiting factor (i.e., constraint) that stands in the way of achieving a goal and then systematically improving that constraint until it is no longer the limiting factor
- **The Second Way – Feedback**
  - Practices include (but are not limited to):
    - Automated testing
    - Peer review of production changes
    - Monitoring/Event Management data
    - Dashboards
    - Production logs
    - Process measurements
    - Post-mortems
    - Shared on-call rotation
    - Change, Incident, Problem and Knowledge Management data
- **The Third Way – Continuous experimentation and learning**
  - Practices include (but are not limited to):
    - Experimentation and learning
    - The Deming Cycle
    - The Improvement Kata
    - Using failure to improve resiliency (e.g., the 'Simian Army' concept first adopted by Netflix)
    - ITSM improvement practices

DevOps Automation Practices

As stated previously, DevOps is not *just* about automation, but there are common enabling practices.

- Treating infrastructure as code
- Repeatable and reliable deployment processes
- Continuous integration, continuous delivery and continuous deployment

- Development and testing (preferably automated testing) performed against production-like systems
- On-demand creation of development, test, staging and production environments
- Proactive monitoring of infrastructure components, environments, systems and services

Automation supports:

- Faster lead times
- More frequent releases
- Less turbulent releases
- Fewer errors
- Higher quality
- Faster recovery
- Business and customer satisfaction

Automation gives rote tasks to computers and allows people to:

- Weigh evidence
- Solve problems
- Make decisions based on feedback
- Use their skills, experience and judgment

Practices that enable the effective use of automation include:

- A tool chain philosophy (vs. a single-vendor solution)
- Shared tools
- Self-service
- Architecting software in a way that enables
  - Test automation
  - Monitoring
- Experimentation

Avoid tools that enforce silos!

Where to begin? Begin by making it easy for people to do the right thing.

- Simplify first – don't automate bad processes
- Automate high value and repetitive tasks
- Automate error-prone work
- Automate to optimize workflow bottlenecks and communication flows
- Improve automated monitoring and notification practices

Adopting a DevOps Culture

It's impossible to talk about any type of major culture change and not bring John P. Kotter into the conversation. In his seminal paper *Leading Change: Why Transformation Efforts Fail*, Kotter laid out steps for adopting any kind of major culture change.

The first of Kotter's eight steps is to 'establish a sense of urgency.' In the context of DevOps, that means getting clear on the business opportunity – the 'Why?' – for your organization in terms of adopting a DevOps culture.

> ***""DevOps is not your why, not your co-workers' why, certainly not your business' why."***
>
> **Damon Edwards**

From there, key steps include:

- **Get the right people together** – ensure core stakeholders are engaged; particularly early adopters who are committed to experimentation and learning.
- **Get everyone on the same page** – seek to understand each other's perspectives and concerns, determine what outcomes you want to achieve and set measurable goals – be realistic!
- **Build capabilities that lead to lasting change** – use education to introduce a common vocabulary, provide ongoing, just in time training, leverage early adopters and informal networks of peer motivators, build trust through transparency, and generate and celebrate short-term wins.
- **Focus on critical behaviors** - Every culture has behaviors that help enable change and others that hinder it. Find ways to nurture the enabling behaviors that matter most.
- **Experiment and learn** – prioritize improvement opportunities, take a holistic approach (i.e., address people, process and technology-related improvements), select and run pilots, capture lessons learned and share, rather than enforce, improved practices.

> ***"Your tools alone will not make you successful."***
>
> **Patrick Debois**

- **Consolidate gains and produce more change** – in the spirit of transparency, communicate successes, failures and lessons learned. Document and make available reusable artifacts and measurements. Continuously invest in needed education, training and technologies, and expand your cycles of improvement.

- **Avoid inertia** – use metrics to prove that the new way of doing things is better. Reinforce new behaviors with incentives and rewards.

> ***"Change sticks when it becomes the way we do things around here."***
> **John. P. Kotter**

DevOps Critical Success Factors

Any type of culture change involves a number of critical success factors. In the context of DevOps these include:

- Management commitment to culture change
- Creation of a collaborative, learning culture
- Common values and vocabulary
- Systems engineering that spans Dev and Ops
- Meaningful metrics
- A balance between automation and human interaction
- Application of agile, lean and agile service management methods
- Open and frequent communication

Summary

DevOps benefits the business by improving communication, collaboration and the integration of people, processes and technologies across the IT value stream.

Ultimately, DevOps enables companies to deliver better software, faster by…

- Improving flow
- Shortening and amplifying feedback loops
- Fostering a culture of continuous experimentation and learning

Get Involved!

DevOps practices will continue to evolve through communities of practice. Seek out opportunities to collaborate with others and to share what you've learned.

Change related to DevOps initiatives will affect organizational culture. Effective communication plans, training, and clear policies and procedures are all needed to achieve the desired performance outcomes and enable collaboration between the many stakeholders involved in DevOps. Culture change and progress cannot happen without the support of people like you.

Take action!

Contribute to your organization's DevOps effort by expanding your knowledge of DevOps principles and practices and by using what you learn to lead improvement activities. Be a change champion!

Bibliography/Additional Sources of Information

- www.devops.com
- www.devopsinstitute.com
- *2014 State of DevOps Report.* Puppet Labs, IT Revolution Press, ThoughtWorks
- *Lean Enterprise: Adopting Continuous Delivery, DevOps, and Lean Startup at Scale.* J. Humble, et al, O'Reilly Media, 2014
- *The Phoenix Project*. G. Kim, et al, IT Revolution Press, 2013
- *Continuous Delivery*. J. Humble, et al. Addison-Wesley Professional, 2010
- *Lean IT: Enabling and Sustaining Your Lean Transformation.* S. Bell and M Orzen. Productivity Press, 2010