

# Introduction to DevOps

May 2018

# Agenda

- What is DevOps
- The History of DevOps
- DevOps Enablers
- DevOps Business Value
- DevOps Perspectives
- DevOps Values - CALMS
- DevOps Principles and Practices
- DevOps Automation Practices Adopting a DevOps Culture
- DevOps Critical Success Factors

# What is DevOps

**DevOps** is a cultural and professional movement that stresses communication, collaboration and integration between software developers and IT operations professionals.

# The History of DevOps

- The term 'DevOps' was popularized during a series of DevOps Days starting in 2009 in Belgium by *Patrick Debois*.
- 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

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

# DevOps Business Value

- DevOps provides companies a competitive advantage by delivering better software, faster and by enabling sustained innovation.
- 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.
- DevOps responds to the demands of business and customer stakeholders for increased agility and stability by increasing overall IT performance.

# DevOps Perspectives

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.

# DevOps Values - CALMS

There are **values** all DevOps stakeholders can agree upon:

- **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.
- **Lean** – We’re seeing the same Lean practices that were applied to manufacturing in the 1980’s being applied to IT now. Do we understand the end-to-end process we use to deliver value (in this case, with software) to our customers? Do we know where the inefficiencies and waste in that process are? Do we have a plan for reducing that waste? The primary Lean tool in our toolkit is value stream mapping.
- **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.

## CALMS



# 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

- **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 I

## 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

# DevOps Practices II

## The First Way – Flow (cont.)

- **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

# DevOps Practices III

## **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**

# DevOps Practices IV

## **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**
- **ITSM improvement practices**

# DevOps Automation Practices

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

# DevOps Automation Practices - Benefits

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

# DevOps Automation Practices – Best Practices

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



# DevOps Automation Practices – Best Practices II

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

1. **Getting clear on the business opportunity** – the ‘Why?’
2. **Get the right people together** – ensure core stakeholders are engaged; particularly early adopters who are committed to experimentation and learning.
3. **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!
4. **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.
5. **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.
6. **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.
7. **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.
8. **Avoid inertia** – use metrics to prove that the new way of doing things is better. Reinforce new behaviors with incentives and rewards.

# 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

“Your tools alone will not  
make you successful.”

Patrick Debois