



# DevOps: Hidden Risks and How to Achieve Results

---

Industry Insights from DevOps Experts Including Gene Kim,  
Co-Author of *The Phoenix Project*, and The Dynatrace Center  
of Excellence®

# Table of Contents

1

DevOps  
& High  
Performance

Slide 7

2

Hidden Risks

Slide 11

3

"The Three  
Ways"

Slide 15

4

Your Real Life  
in Dev & Ops

Slide 20

5

Start Small  
but Start  
Now!

Slide 23

6

Some  
Homework  
for You

Slide 26

This paper is gleaned from Gene Kim's Keynote Presentation at Dynatrace's Perform User Conference October 2014 and highlighted with insights from Martin Etmajer, Senior Technical Strategist, and Wolfgang Gottesheim, Technical Strategist, Dynatrace Center of Excellence.

DevOps: Hidden Risks and How to Achieve Results

---

# Introduction

# Industry Insights From Guys That Eat, Drink and Breathe DevOps



Gene Kim

*Co-Author of The Phoenix Project*

Gene Kim is a multiple award-winning entrepreneur, the founder and former CTO of Tripwire and a researcher. He has written three books, including co-authoring *The Phoenix Project*.

[Twitter](#): @RealGeneKim

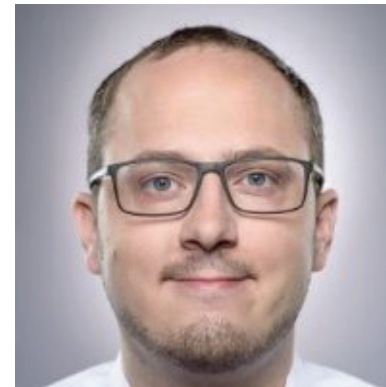


Martin Etmajer

*Senior Technology Strategist,  
Dynatrace Center of Excellence*

Martin has 10+ years of experience as a developer and software architect, as well as maintaining highly available performance cluster environments.

[Twitter](#): @metmajer



Wolfgang Gottesheim

*Senior Technology Strategist,  
Dynatrace Center of Excellence*

Wolfgang has over a decade of experience as a software engineer and research assistant in the Java enterprise space.

[Twitter](#): @gottesheim

# Executive Summary

The top performing organizations know that they must adopt a continuous and collaborative IT delivery model to meet the ever-increasing market demands for more features, faster.

DevOps is proven to produce faster deployments with fewer problems. And the leaders are seeing real business gains: they are 2.5x more likely to exceed profitability, market share and productivity goals and have 50% higher market capital growth over three years.<sup>4</sup>

The market leaders' results show us that DevOps is necessary to be competitive, but there are many challenges in the way. Fortunately, we have some experts to help. Recommendations in this eBook are based upon Gene Kim's keynote presentation at the Dynatrace annual Perform conference and the book he co-authored, *The Phoenix Project* and the expertise of the Dynatrace Center for Excellence.

Leaders are seeing real business gains; they are 2.5x more likely to exceed profitability, market share and productivity goals.

This eBook is for all Development Managers, Operations Managers and DevOps Managers. Whether you are just starting your DevOps journey or still considering how a DevOps transformation can help your organization, this eBook will give you insight into why it is important, what makes the leaders successful and provide guidance on how to accelerate your results.



## Section 1

---

# DevOps & High Performance

# Devops is Not Just Last Year's Buzzword.

High-performing DevOps is the continuous delivery of technology made possible through a collaborative IT culture and shared performance metrics tied to business results, supported by automated tools throughout the development lifecycle.

Google, Amazon, Netflix, Facebook, Etsy and Microsoft and countless tech start-ups are doing it. So are The Gap, Nordstrom, the World Bank, and Intuit. Does it surprise you that even the UK Government and the U.S. Department of Homeland Security are doing DevOps?

All of these organizations know that today's marketplace is driven by ever-increasing demands for features, speed and reliability; they must continuously deliver new value without disruption to their customers' experience. And the way to get there is DevOps.

**A little skeptical? Are you thinking: wasn't DevOps just the buzzword of the year last year? Is there really value there for my organization and me?**



There are many different views on what it means to have a high-performing DevOps organization. From our experience working with IT organizations across many industries, we have found certain qualities to be critical for today's high-performing organization.

The principles that make DevOps work are far from new. Lean manufacturers embraced these ideas in the 1980s to build with unprecedented speed and

quality. Other organizations like big-box retailers and service firms have gained a competitive advantage through the use of cross-functional teams, shared performance metrics and workflow automation for years. And, Agile software development has been around since 2001.

# Today, Your IT Organization is Your Competitive Advantage.



vs.

Yesterday's IT
IT as a cost center
Developers blame Ops—Ops blames developers
Waterfall development
Performance considerations begin at code check-in
Deployments only 1-2x a year
Change approval board required to deploy
Version control for code only or no version control at all
Failures found by NOC or worse, by customers

Today's High Performing DevOps
IT as a competitive advantage
High trust organization with shared goals
Continuous integration and delivery
Performance is a critical requirement and is considered throughout the development lifecycle
Multiple deployments a day
Peer review of code quality
Version control for code, system, application configurations and data
Proactive monitoring to find failures first



# The Result?

Organizations that have truly adopted DevOps principles across the application development lifecycle are deploying faster, having fewer problems and are seeing real business gains.

- > Flickr was one of the first to announce they were following DevOps principles with 10 deployments a day way back in 2009.<sup>1</sup>
- > TurboTax recently made 165 production changes during peak tax season resulting in 50% increase in website conversion rate.<sup>2</sup>
- > Amazon deploys at an amazing pace: every 11.6s with 23,000 deployments a day. They have had 75% fewer outages since 2006, 90% fewer outage minutes, and only 0.001% deployments cause a problem.<sup>3</sup>

Here are some more interesting facts about high-performing DevOps from the two groundbreaking PuppetLabs reports in 2013 and 2014:<sup>4</sup>

## They are Faster, More Reliable and are Winning the Marketplace

- > 30x more frequent deployments
- > 8000x faster lead times than peers
- > 2x the change success rate
- > 12x faster mean time to recover
- > 2.5x more likely to exceed profitability, market share and productivity goals
- > 50% higher market capital growth over 3 years



## Section 2

---

# Hidden Risks

# What Are the Biggest Obstacles You Will Face?

The experiences of market leaders make it clear. DevOps, if it is not now, will be a necessity in the near future just to keep up. But a culture change is never easy. Here are some of the top DevOps challenges that organizational teams face:

## 1

Too many hand-offs between work centers

As the number of hand-offs between teams increase due to lack of communication, alignment and automation, feature lead-time increases.

## 2

Testing is happening too late in the system development lifecycle

If you do not integrate an automated testing discipline throughout the development lifecycle, testing teams will have to re-do manual testing every time a code or configuration change is made, and problems will be found too late to make systemic fixes.



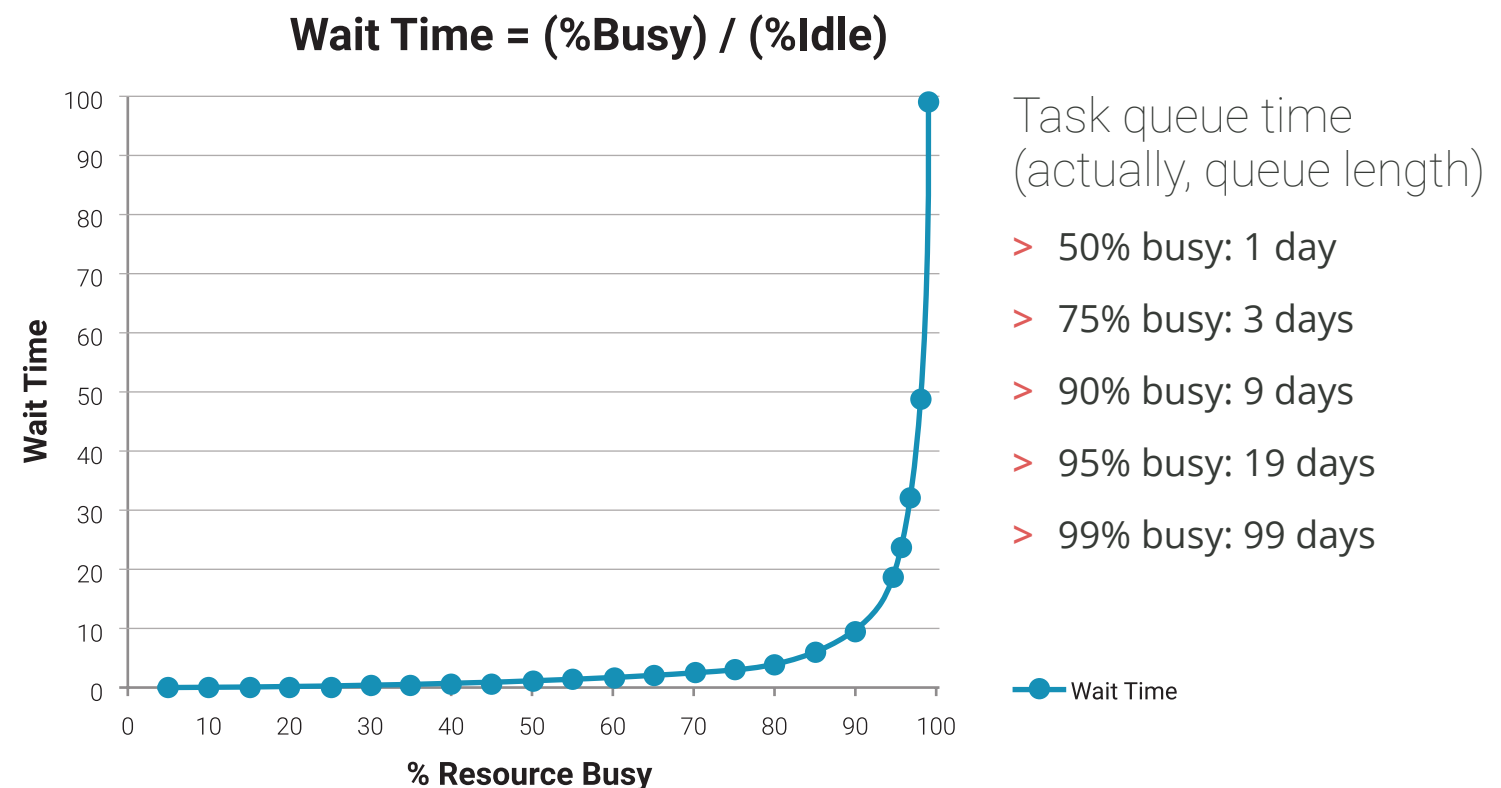
# What Are the Biggest Obstacles You Will Face?

## 3

### Why 15 min tasks take weeks to complete<sup>6</sup>

Work sits in-queue as opposed to being actively in-work

Your work teams are not building quality in at the beginning, e.g. by implementing automated testing and deployment strategies. Work isn't being completed on time and quality suffers. Create smaller work packages and make quality everyone's #1 concern.



# What Are the Biggest Obstacles You Will Face?

**4**

Work is bouncing back and forth between work centers

If your teams are not working towards the same goals and are measured upon them, they will act independently of each other. Expectations and tooling are not aligned and when there is a problem, investigation involves a lot of guesswork and finger pointing.

**5**

Development can't reproduce test or production failures

IT Operations report an ambiguous log message from the production environment. Your Developers have no clue how to fix it because they are missing vital contextual information and there is no unified view of performance data across teams.

**6**

There is high resource utilization in both dev and test

Key resources are being overused and all too often approached directly to do ad-hoc tasks. Shield this unplanned work, increase flow and reduce work in progress.

“Improving daily work is even more important than doing daily work.”

— Gene Kim, Co-author of, *The Phoenix Project: A Novel About IT, DevOps, and Helping Your Business Win*

## Section 3

---

# “The Three Ways”

Principles That Enable High Performance

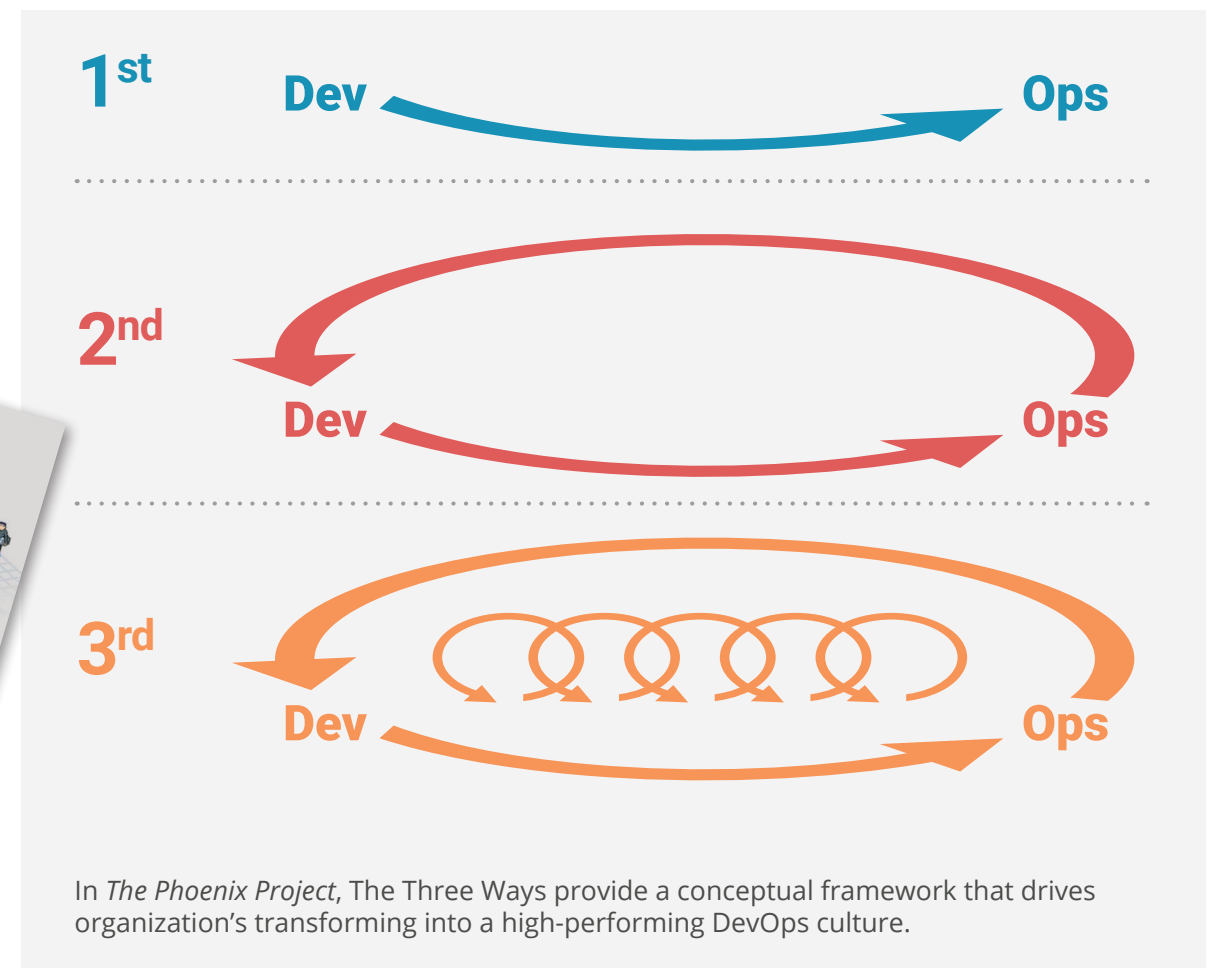
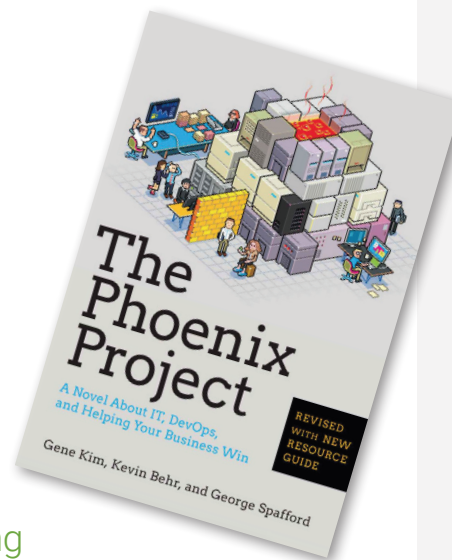
# The Situation Seems Pretty Bleak, Doesn't It?

It doesn't have to be.

Some great input on how to build a true DevOps culture comes from *The Phoenix Project* co-authored by Gene Kim et al. In the book, the IT manager saves the day by using *The Three Ways*, a mysterious yoda-like philosophy. It is a conceptual framework that can help drive any organization's transformation to a high performing DevOps culture.

“A must-read for anyone wanting to transform their IT to enable the business to win.”

— Mike Orzen, co-author of the the Shingo Prize winning book *Lean IT: Enabling and Sustaining Your Lean Transformation*



In *The Phoenix Project*, The Three Ways provide a conceptual framework that drives organization's transforming into a high-performing DevOps culture.

# The First Way: Systems Thinking

This first principle of *The Three Ways* philosophy emphasizes the performance of the entire system and builds quality into the products and the software development lifecycle itself.

Your goal with this principle is to continuously increase the flow from development to operations and minimize any return flow. Operations should be involved early on in the development process so that they can plan for any necessary changes to the production environment (as described in more detail in *Continuous Delivery* by Jez Humble and Dave Farley).

Smaller, more frequent software releases reduce waste and improve your time-to-market. Automated testing should be used to uncover defects before they affect your business.

Facebook released their chat service in less than a year to 98 million users by deploying code directly to production every single day.



## facebook

### Facebook Chat

By Eugene Letuchy on Tuesday, May 13, 2008 at 10:56pm

One of the things I like most about working at Facebook is the ability to launch products that are (almost) immediately used by millions of people. Unlike a three-guys-in-a-garage startup, we don't have the luxury of scaling out infrastructure to keep pace with user growth; when your feature's userbase will go from 0 to 70 million practically overnight, scalability has to be baked in from the start. The project I'm currently working on, Facebook Chat, offered a nice set of software engineering challenges:

#### Real-time presence notification:

The most resource-intensive operation performed in a chat system is not sending messages. It is rather keeping each online user aware of the online-idle-offline states of their friends, so that conversations can begin.

The naive implementation of sending a notification to all friends whenever a user comes online or goes offline has a worst case cost of  $O(\text{average friendlist size} * \text{peak users} * \text{churn rate})$  messages/second, where churn rate is the frequency with which users come online and go offline, in events/second. This is wildly inefficient to the point of being untenable, given that the average number of friends per user is measured in the hundreds, and the number of concurrent users during peak site usage is on the order of several millions.



# The Second Way: Amplify Feedback Loops



The second principle is about establishing a feedback process from operations back to development.

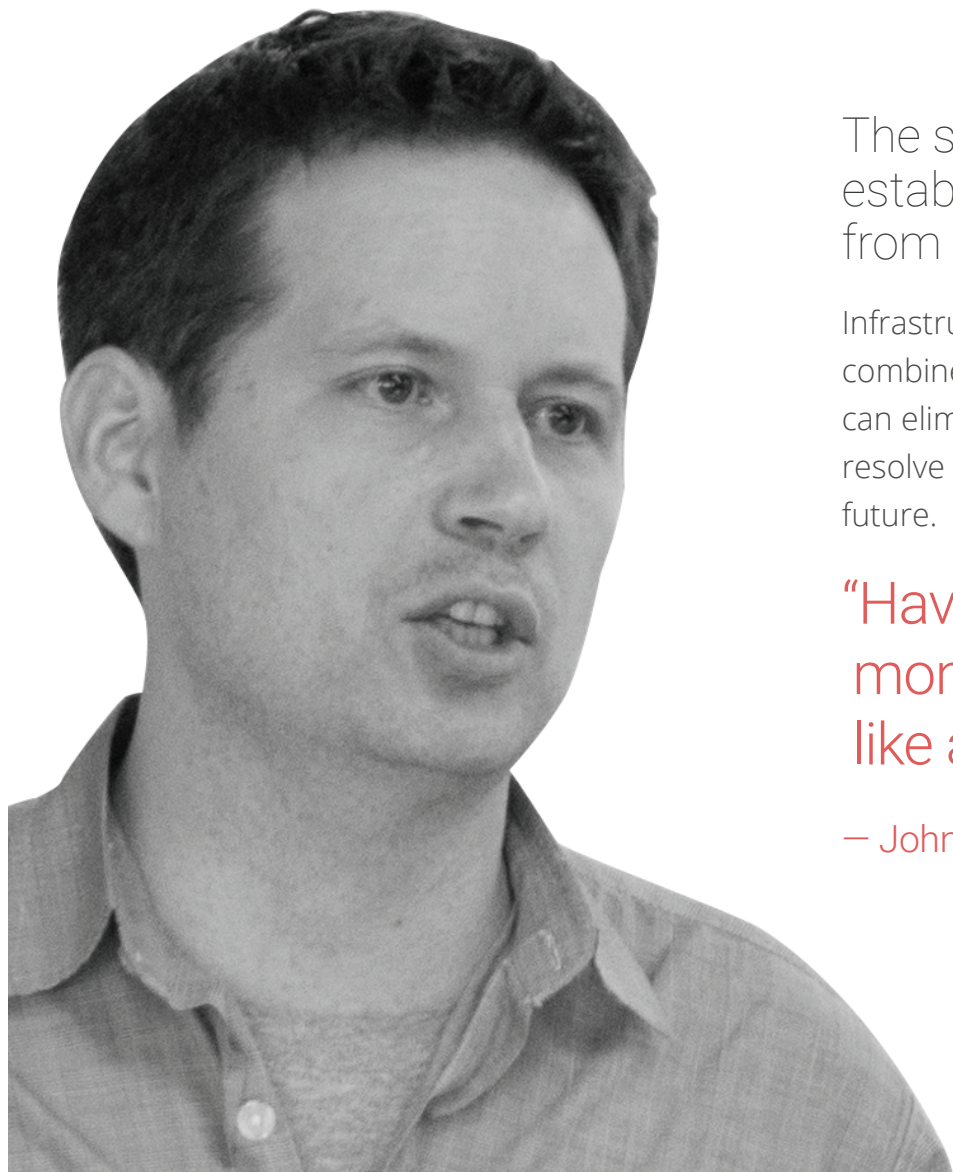
Infrastructure and application performance monitoring, combined with strong communication processes, can eliminate finger-pointing. You can recognize and resolve problems quickly, and then prevent them in the future.

“Having a developer add a monitoring metric shouldn’t feel like a schema change.”

— John Allspaw, SVP Tech Ops, Etsy<sup>1</sup>

An example practice is to let development, testing and operations get together frequently to jointly analyze issues in production. After such a meeting:

- > Development should know how to tackle the issue in the code.
- > Testing should know which kind of tests are required to prevent the same issue from appearing in production again.
- > Operations should put the problem pattern onto their radar to be notified should the failure persist or reoccur.



© John Allspaw — Flickr<sup>7</sup>

# The Third Way: Continual Experimentation and Learning

The third principle is about continuous improvement.

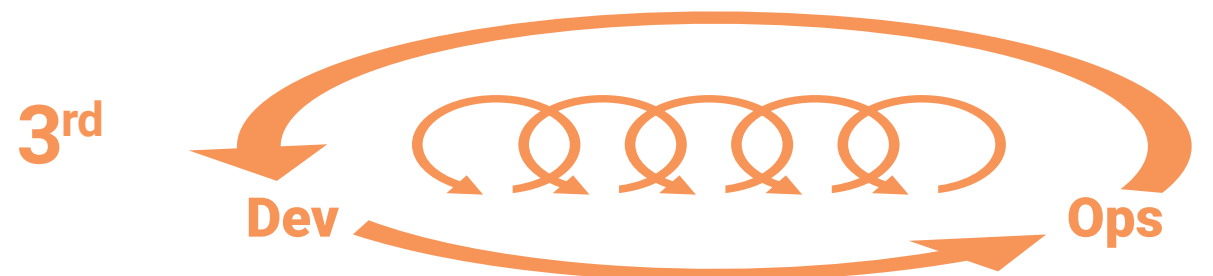
Having mastered the first two ways, you can now afford to experiment and take risks. You can even inject faults into your system, find out where it breaks and then fix the system to increase resilience. It is about getting out of your comfort zone and into the danger zone - and if it hurts, do it more often.

Aside from fault injection testing, you could monitor whether your business hypotheses hold: are your users using particular features and how are they using them? Also, keep working hard on improving your development, test and release processes.

“Our goal is to do painful things more frequently, breaking all of human nature.”

— Adrian Cockcroft,  
Technology Fellow, Battery Ventures;  
formerly Cloud Architect, Netflix<sup>2</sup>

This is at the core of what saved Netflix during the EC2 outage in 2011. The site remained operational because of the Chaos Monkey, a piece of code that kills processes in production randomly requiring new code and environment changes to be strong. The result is Netflix was one of the very few businesses that did not go down during this failure.



## Section 4

---

# Your Real Life in Dev & Ops

# Our Recommendations for Making DevOps Work for YOU!

Is a high-performing DevOps culture really possible for everyone? Yes, yes and yes!

It is a necessity, if the lean manufacturing revolution is to be our model. As W. Edwards Deming, the quality management guru of the '80s once said, "Learning is not compulsory... neither is survival."

Here are a few recommendations from our collective experience that we think will help you along your way:

**1. Make deployments business-as-usual.** By decoupling code deployment from feature releases and decreasing deployment size, you can make deployments day-to-day business instead of that Friday at midnight migraine.

**2. Make everyone a performance engineer.** This means making a production-like environment available on-demand to developers, ensuring that non-functional requirements (i.e. performance, security) are handled at every stage of the process and empowering the cross-functional team to conduct a genuine, collaborative line of inquiry when something goes wrong.

**3. Strengthen the safety net with automated testing.** Integrating these tools throughout development (not just at the end) will catch errors early allowing your team to deal with changing requirements while producing reliable software.



"It is only through automated testing that we can transform fear into boredom... The only way you can get people productive is to show them there is a safety net underneath them that will catch errors long before they get into production."

— Eran Messeri, SCM, Google

# More DevOps Recommendations for You

## 4. Be strict with version control.

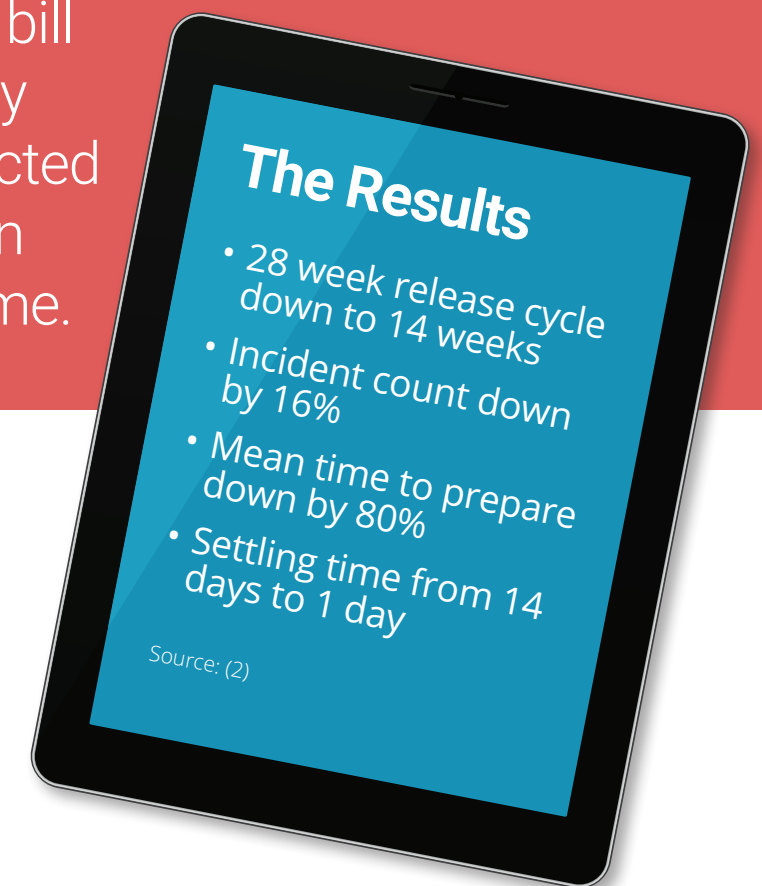
Version control is critical in both development and operations. However, the use of version control (configuration) in operations — on operating systems, databases, storage and networking, etc — has the strongest correlation with high performing DevOps organizations.<sup>4</sup>

**5. Implement end-to-end performance monitoring and metrics.** Providing insight into performance throughout development, testing and production builds better and more robust software with fewer surprises in production. A single tool used across functions can help you to align teams on shared performance goals that support the business objectives.

**6. Peer code review.** Peer code review is strong indicator of high performance, but one of the hardest to implement for large organizations.<sup>4</sup> When command and control increases, performance suffers. If peer review is out of reach for you, putting in a code review process with subject matter experts can be helpful.

**7. Allocate more cycles to the reduction of technical debt.** If unchecked, this debt will be a roadblock to releasing new features and ultimately cause you to lose in the marketplace. You need to set aside time for your development and operations teams to improve—to fix problematic areas of code and implement better testing and deployment processes.

**CSG**, the largest bill printing company in the US, conducted an experiment on COBOL mainframe.



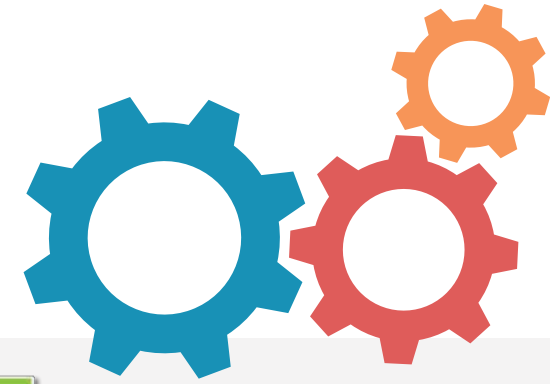
## Section 5

---

# Start Small but Start Now!



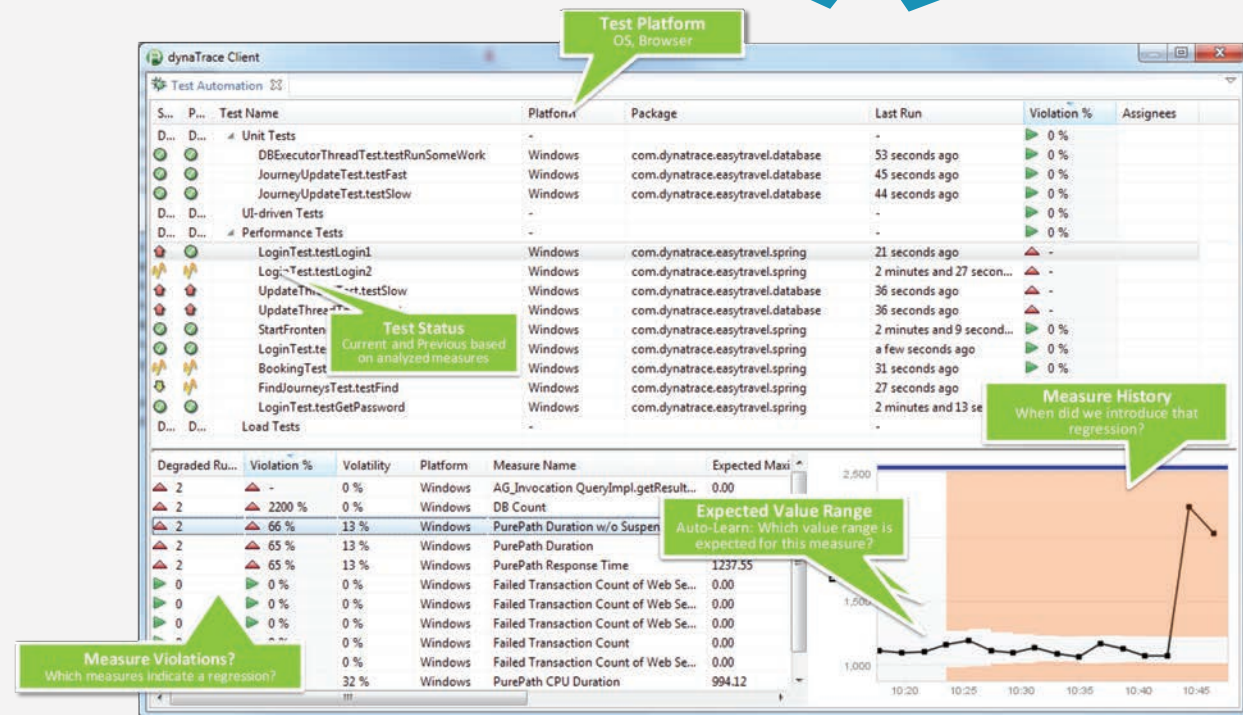
# A Good Place to Start - Application Performance Management



If all the recommendations in Section Four, Your Real Life, seem overwhelming, a great place to start is application performance monitoring.

It is an important capability for DevOps and one that can provide benefits right away. In fact, a [recent survey of 1400 organizations](#) ranked Performance Monitoring and Performance Testing as critical for DevOps.

Application performance management tools can provide insight into performance throughout the development lifecycle. It can also expedite collaboration between development and operations with each considering the same performance metrics from early on in the development lifecycle.



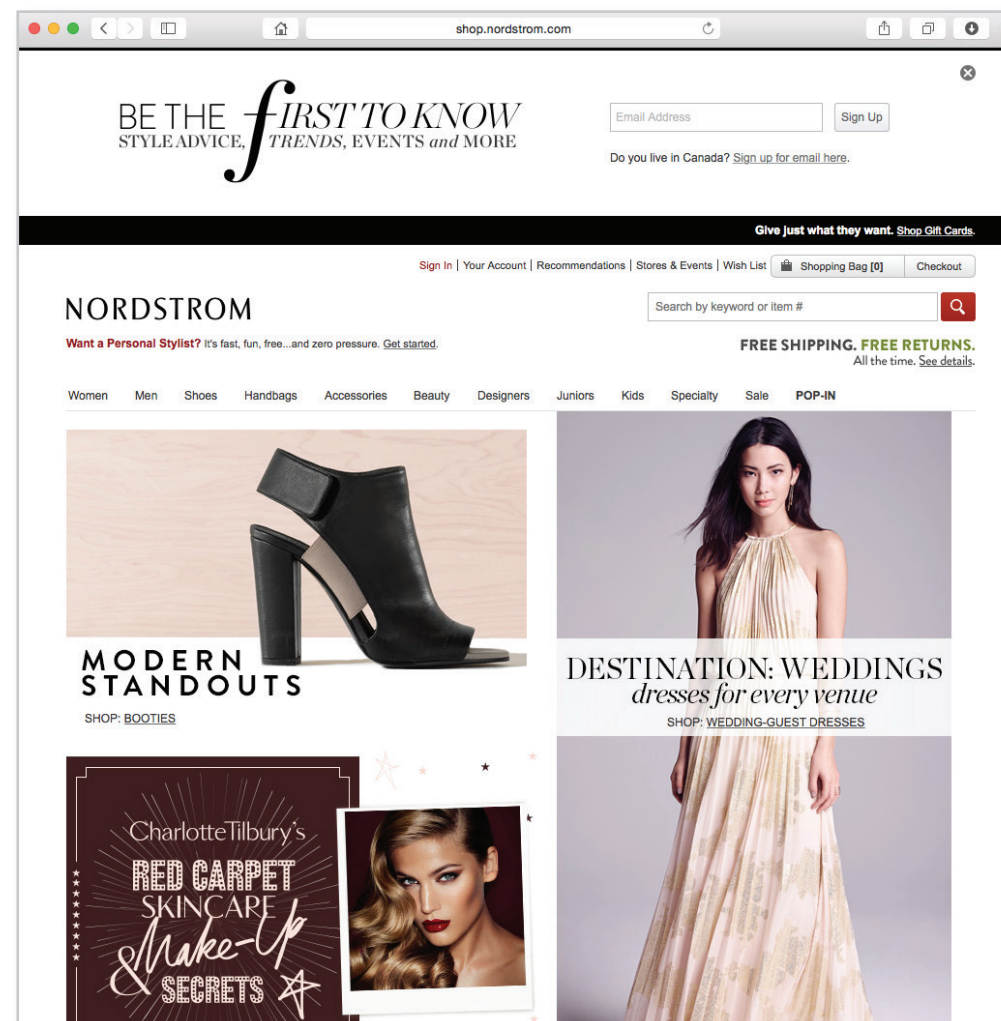
Performance monitoring shouldn't start in production environment. Tracking performance of automated tests, together with analytics to identify outliers and performance problems, helps to keep performance problems from making it into production.

# Nordstrom: An Application Performance Management Case Study

One organization that has integrated performance monitoring into its DevOps approach is Nordstrom. Nordstrom.com has had 30% revenue growth year-over-year and is hosted internally on hundreds of servers.

Nordstrom implemented application performance management tools across the application development lifecycle. These tools helped them design for performance and get features out more quickly.

Before they had the tools, they couldn't do the level of testing necessary to find issues. The issues would end up in production and then it would take time to identify the root cause and ultimately, result in downtime.



“Before [performance management] tools existed it was like beating our heads against the wall. [With the new performance management tools], identifying issues went from days to hours. Our test cycles went from weeks to days.”

— Gupal Brugalette,  
Senior Applied Architect for  
Performance Engineering,  
Nordstrom



## Section 6

---

# Some Homework for You

# Happy Reading!

We know we've given you a lot to think about. A good to great DevOps transformation, like any major culture change, will involve some hard work. We hope that we have given you a few good steps in the right direction.

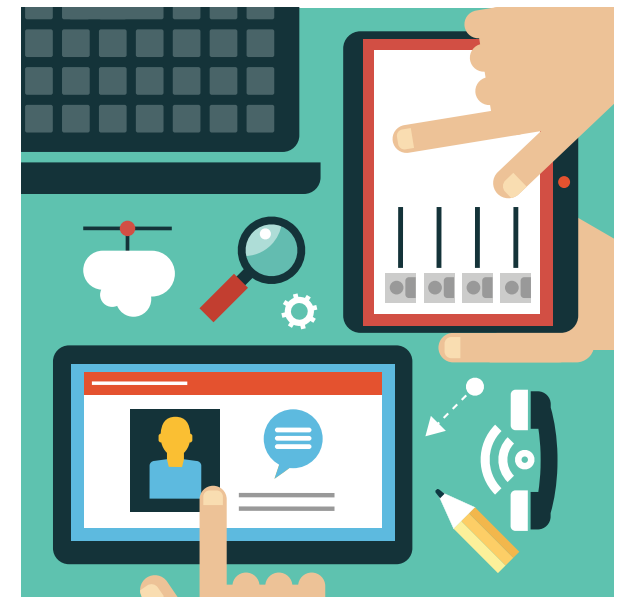
Now, for some homework. Continual learning is a principle of *The Three Ways*, after all! Here we have identified some resources to support you on your journey.

## Recommend Reading

- > *The Phoenix Project* by co-author Gene Kim
- > *The Speed of Trust* by Steven Covey
- > *Release It* by Mike Nygard
- > *Continuous Delivery* by Jez Humble and David Farley
- > *The Other Side of Innovation* by Vijay Govindarajan
- > *The Goal* by Eliyahu Goldratt

## Blogs to Follow / Blog Posts to Read

- > [itrevolution.com/devops-blog/](http://itrevolution.com/devops-blog/)
- > [blog.devopsguys.com/](http://blog.devopsguys.com/)
- > [devopsreactions.tumblr.com/](http://devopsreactions.tumblr.com/) - Enjoy some DevOps humor!
- > [bartongeorge.net/2013/06/27/developers-it-ops-cloud-innovation/](http://bartongeorge.net/2013/06/27/developers-it-ops-cloud-innovation/)
- > [apmblog.dynatrace.com/2014/03/13/software-quality-metrics-for-your-continuous-delivery-pipeline-part-i/](http://apmblog.dynatrace.com/2014/03/13/software-quality-metrics-for-your-continuous-delivery-pipeline-part-i/)
- > [apmblog.dynatrace.com/2014/11/18/continuous-delivery-101-automated-deployments/](http://apmblog.dynatrace.com/2014/11/18/continuous-delivery-101-automated-deployments/)



# DevOps Tools We Love :)

Here is a collection of tools that we enjoy using (at Dynatrace) for our internal processes. They foster collaboration amongst our Product Management, Development, IT Operations and Technical Support teams, allow us to build more quality into our products, and support us in establishing better feedback loops.

Recommended Tools  
From Martin & Wolfgang:

- > Development - GitHub
- > Virtual Machines - Vagrant, Packer and VeeWee
- > Configuration Management
  - Ansible, Chef and Puppet
  - Test Kitchen
  - Elasticsearch and Kibana
- > Fault Injection Testing – Docker

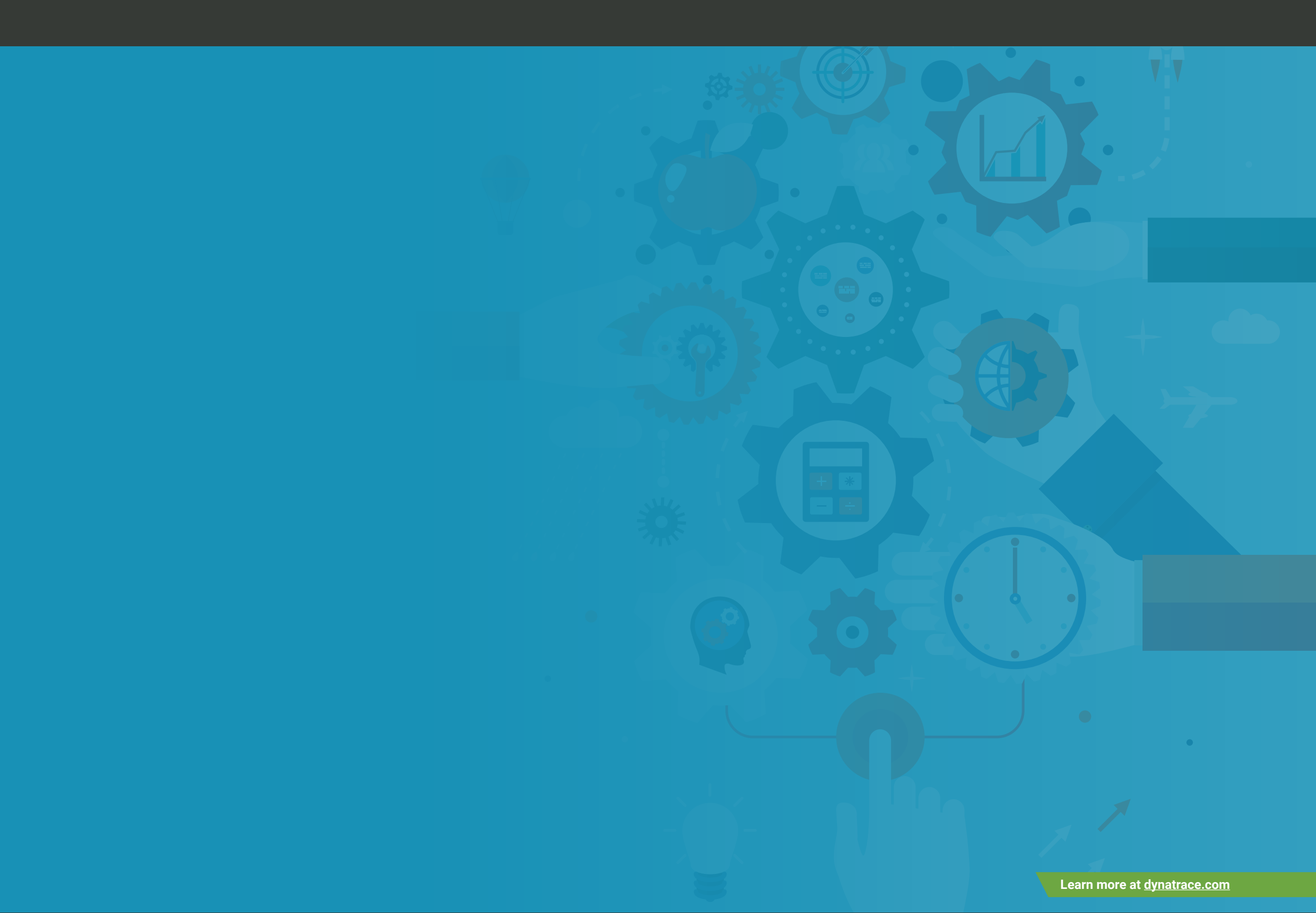


Stop back once in a while, and we'll gradually update this list with more interesting bits.

[apmblog.dynatrace.com/2015/02/05/top-devops-tools-love/](http://apmblog.dynatrace.com/2015/02/05/top-devops-tools-love/)

# Sources

1. Allspaw/Hammond, Flickr, 2009 Velocity Presentation (<http://www.slideshare.net/jallspaw/10-deploys-per-day-dev-and-ops-cooperation-at-flickr>) and Gene Kim, Excerpt Points from Live Presentation, Dynatrace Perform Conference, 2014.
2. Gene Kim, Excerpt Points from Live Keynote, Dynatrace Perform Conference, 2014.
3. Amazon, 2012 Velocity Presentation Presentation.
4. Puppet Labs, IT Revolution Press and Thoughtworks, *2014 State Of DevOps Report* and *2013 State Of DevOps Infographic*.
5. Enterprise Management Associates, *DevOps and Continuous Delivery*, 2014
6. Gene Kim, Excerpt Points from “Mastering Performance and Collaboration Through DevOps” Webinar, Dynatrace, October 2014.
7. This image is licensed under the Creative Commons Attribution-ShareAlike 4.0 International License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-sa/4.0/> or send a letter to Creative Commons, PO Box 1866, Mountain View, CA 94042, USA.
8. Gene Kim, along with Kevin Behr and George Spafford, is author of “The Phoenix Project: A Novel about IT, DevOps, and Helping Your Business Win”



Learn more at [dynatrace.com](https://dynatrace.com)