

For: Infrastructure  
& Operations  
Professionals

# The Seven Habits Of Highly Effective DevOps

by Amy DeMartine and Kurt Bittner, October 2, 2014

## KEY TAKEAWAYS

### **You Can Reverse The Negative Behaviors**

The plague hampering DevOps progress in most organizations is rooted in some awful stereotypes. Learn how DevOps behaviors can fuel remarkable progress when you destroy the stereotypes and embark on a genuine organizational fitness program that spans the service life cycle.

### **You Can Learn From Real-Life Success Stories**

High-profile web pioneers are revolutionizing their industries with technology. The scale and velocity of this technology would be impossible without stellar DevOps practices in place. You may not be able to match their capabilities (yet), but you can emulate their behaviors and improve dramatically.

### **Automation Is Essential To DevOps, But There Is Much More**

Open communication, strong systems engineering methods, and different processes must augment the automation software at the center of the DevOps hype. If you merely automate poor practices, you do bad things faster. Invert this equation and do good things faster. That is the secret to real DevOps glory.

# The Seven Habits Of Highly Effective DevOps

by [Amy DeMartine](#) and [Kurt Bittner](#)

with [Glenn O'Donnell](#), [Jean-Pierre Garbani](#), [Eveline Oehrlich](#), Michelle Mai

## WHY READ THIS REPORT

Organizations have for years sought a means of more quickly delivering business value in the form of working software. Inspired by innovations in Agile software development and the application of the Lean principle of continuous improvement, development and operations have been chipping away at the obstacles that have prevented faster delivery. Joining together, the movement, called DevOps by some, has been gaining momentum and achieving impressive results. The walls between development, operations, and the business are coming down, ushering in a new age of collaboration that, in leading organizations, is already proving itself in delivering real business value faster and more frequently. This report is a comprehensive update to the original published report on September 3, 2013, and includes the current DevOps landscape and our most recent Business Technographics® survey data. This report presents the seven main principles of DevOps that have been proven by some of the most dominant technology innovators in the world.

## Table Of Contents

- 2 **DevOps Means True Collaboration Of Dev And Ops**
- 7 **The Seven Habits To Develop For DevOps Success**
- 13 **DevOps Is Not Only Possible But Necessary In Both IT And BT**
- WHAT IT MEANS
- 14 **Instant-On Built-To-Order Services Are The Future**
- 14 **Supplemental Material**

## Notes & Resources

Forrester incorporated feedback from numerous end user and vendor interactions, as well as various other client inquiry and consulting engagements.

## Related Research Documents

[Elevate The ALM Discussion To The Business Plane](#)

February 7, 2013

[Improving The Ops In DevOps](#)

July 21, 2011

[Augment DevOps With NoOps](#)

April 26, 2011



## **DEVOPS MEANS TRUE COLLABORATION OF DEV AND OPS**

The velocity of change in business requirements is undeniably increasing at a frightening rate for those organizations unable to keep pace. The technology management services necessary to fulfill those requirements are therefore accelerating accordingly. Application development professionals (AKA Dev) are responding to this challenge with approaches like Agile development, scrum, and automated testing to yield faster code development and release. Infrastructure and operations professionals (AKA Ops) are likewise responding with virtualization, continuous design and delivery, service management, cloud, and more. The bottom line here is the need to accelerate change, which often takes days or weeks. The time frame has shrunk to hours or minutes, and some web pioneers are measuring change intervals in seconds!

DevOps is a movement within technology management that is driven by a need to improve the collaboration between development and operations. DevOps concisely captures an organizational philosophy already employed by the most effective people and organizations in the world. DevOps is not so much a destination as a journey, a kind of fitness program that can help you transform your business into a world-class competitor. Like any fitness program, it will be difficult and, at times, painful. The end result, however, is a lean, nimble, and muscular technology management organization that will allow your business to vanquish its foes.

## **Negative Stereotypes Cripple Your Business**

Dev and Ops often sit on opposing sides of a great divide, separated by a gulf of misunderstanding. There are many reasons for these misunderstandings, but collectively, the detritus of accumulated and reinforced stereotypes is destructive to organizational harmony (see Figure 1).

Without good interpersonal harmony, your business cannot adapt to the changes in the competitive landscape. As long as these stereotypes continue to exist, they will remain a barrier that will prevent you from achieving greatness.

Obviously, these stereotypes are gross caricatures, but in every caricature there is a kernel of the truth. Regardless of the underlying partial truths, these stereotypes, left unchecked, are preventing you from delivering the value that you need to achieve superior results.

The reality is that neither Dev nor Ops really understands what the other does, and what they don't understand, they can't respect. Therefore, your first step as a leader is to broker some "relationship counseling." Just getting Dev and Ops talking can be a good start, but it's not going to be pleasant, at least at first.

Misunderstandings are reinforced by the myopic isolation of technology silos. For DevOps, the silos are exacerbated by additional organizational isolation. Acknowledging the stereotypes with a sense of humor can defuse the situation. After that, agree on a common goal and benchmarks; the success of the business in delivering value rapidly, reliably, and repeatedly provides a foundation on which to build a new shared culture of performance.

The DevOps divide is amplified by additional factors:

- **Service complexity is inexorably accelerating.** This complexity means it is now virtually impossible for any one person — or any one group — to possess sufficient visibility and knowledge of the service provided and/or consumed by the business. Even the developers — who once knew everything about their applications — are left with a visibility deficit. Cooperation, aided by technology, is increasingly necessary to manage this complexity.
- **Dev is more directly connected to business processes and business customers.** This reinforces the introspection among Ops teams that they are the “blue collar laborers who do the real work.” Such a perspective magnifies the divide because the Ops people unwittingly suppress themselves, and others feed into that cycle, relegating Ops to that position even more. This is a very dangerous cycle that you must interrupt and reverse.
- **Many Dev groups reside in the business units, not anywhere in technology management.** This adds even more organizational separation between Dev and Ops. Reaching across this chasm can prove more difficult, but certainly not impossible.

You must address all of these factors. It requires commitment across all parties to confront and destroy the obstructions and to therefore foster a more congenial atmosphere.

**Figure 1** Stereotypes Destroy Trust And Thwart Opportunities For Collaboration

Dev's stereotypes of Ops	Ops' stereotypes of Dev
Ponderously slow to act, whereas Dev feels agile and able to move quickly	Careless and lacks discipline; Ops needs to perform heroics to keep the mess running
Wants to maintain the same old obsolete systems that cannot adapt to Dev demands	Likes to play with the latest “toys”
The department of “no” — there’s always a reason something can’t be done	Thinks anything is possible and there are no costs or other implications to its demands
Runs unreliable infrastructure that negatively affects Dev’s credibility	Delivers lousy code that negatively affects Ops’ credibility
Not creative enough to understand the complex artistry of apps	Doesn’t understand the real world, and creates unrealistic test environments
Just a bunch of process zealots and bureaucrats	Prima donnas who practice “art for art’s sake”
Cobbles together infrastructure — it’s not engineering	A bunch of hackers, not real professionals
Dev can do much better by replacing Ops with the cloud and supporting itself.	Ops can do much better by replacing Dev with packaged software and SaaS.

## Measurable Speed, Productivity, And Quality Are Your Goals

The ultimate mission of both Dev and Ops is to create competitive supremacy for your business (see Figure 2).

To achieve this supremacy, you must dramatically improve three organizational performance measures:

- **Speed of execution.** To gain competitive advantage, you need to execute faster to develop new business capabilities and put these capabilities into the hands of those who wield them. The Dev team is often perceived to provide Agile development, but it is an empty promise without Ops. The typical metric cited for this is “cycle time,” the time from when work on a new feature or fix starts and when it can be deployed in production.
- **Staff productivity.** Every organization seeks to do more with less; the recent global economic crisis has amplified the imperative. In the DevOps context, organizations need to be able to support more frequent releases to improve results without adding staff or complex and costly manual processes. This means that DevOps rests on a foundation of *highly automated processes*. To achieve superior results, you must also streamline and standardize manual processes. One of the barriers to effective DevOps is past reliance on personal heroics, a cultural value that many organizations unwittingly reward and so cement in place behaviors that stand in the way of systematic improvement.
- **Service quality.** Business leaders are understandably frustrated: They have repeatedly invested heavily in the systems and people who provide technology-based services, yet the payoff continues to elude them. Similar investments in other business functions provide tangible and worthwhile returns, yet returns on technology management investments remain an enigma. It is no surprise that business leaders are seeking alternatives more openly and aggressively than in the past. Cloud-based service providers now offer increasingly viable options. Failure of technology management to increase value delivery and service quality at competitive cost will lead to an increase in outsourcing entire technology management functions and the jobs along with them.

**Figure 2** The Top Business Concerns Are Driving IT Staff To Improve Business Competitiveness



Source: Forrester's Business Technographics® Global Priorities And Journey Survey, 2014

93781

Source: Forrester Research, Inc. Unauthorized reproduction or distribution prohibited.

### There Is No ‘We And They’ – Only ‘Us’

In every problem there is an opportunity struggling to emerge. Rather than taking a defensive posture, with Dev and Ops playing off one another to avoid the blame for the frustrations of the business, Dev and Ops can join together to optimize the stream of business value delivery in ways that outsourcing service providers cannot approach.

To achieve this, both Dev and Ops professionals face considerable change in the way they work. Adhering to the same old ways and working in a vacuum will prove fatal to both. Only by working together can they adapt collectively to keep creating the future. At some point in that future, the independent functions of Dev and Ops will cease to exist and instead be merged into one — and this merged function will be nothing like the Dev or Ops of today. Create the future; don't be a victim of it.

### DevOps Is Science Fact, Not Science Fiction

The principles of DevOps are being practiced by several organizations with names that many will recognize: salesforce.com, Etsy, Facebook, and, perhaps surprisingly, IBM, to name just a few. The following are some feats that might appear extreme to many of you:

- **Salesforce.com has up to 9,000 code changes and performs 1.1 million tests per hour.** With a goal of delivering the highest quality product possible, salesforce.com combined several principles from common Agile methodologies that allow it to react, adapt, and collaborate in over 600,000 files. As a result, salesforce.com feels it is able to leverage the innovative capabilities of its engineering teams, release meaningful product enhancements throughout the year, and deliver on engineering excellence.
- **Facebook showed results of some early work with Opscode's Chef automation software.** In an April 26, 2013, keynote at ChefConf, Facebook Production Engineer Phil Dibowitz and his team tested their Chef automation on a large collection of servers. They were easily able to automate full OS and application deployments to 17,000 heterogeneous servers and 17,000 homogeneous servers, all with a single Chef server. In both cases, the Chef server suffered very little performance degradation, especially for homogeneous server automation. Facebook uses significantly more than 17,000 servers, but Dibowitz feels confident that these results will allow his team to support it all with only a few Chef servers and only four people to maintain the entire automation system. These people are certainly not your typical systems administrators. They have evolved to a much higher state. They are creating the future of Facebook and demonstrating some methods we can all borrow to “do more (much more) with less,” as Dibowitz said, so we can all create our own futures.<sup>1</sup>
- **IBM accelerated capabilities delivery in its IBM APM SaaS portfolio by 400% in eight months.** Committing to the software-as-a-service (SaaS) market means that IBM is getting aggressive in how it approaches the software market. Starting with its Application Performance Management (APM) offering, IBM's Cloud and Smarter Infrastructure division shifted to new automated delivery processes and a new organization structure that optimizes both development and operations. Director of Product Management and Strategy Pratik Gupta says that a key driver for the transformation was to become nimble enough to adapt to changes in market conditions.
- **Etsy improved a typical web deployment.** Etsy, an online marketplace for independent artisans, went from using three developers and one operations engineer, with everyone else on standby, which took over an hour (when things went smoothly), down to one person in under 2 minutes.

“Once deploying is no big deal, a lot of things can change. Features can go out a piece at a time instead of one all-or-nothing push. Your app configuration options can be in code — and changed quickly. Your hair can grow back.” (Erik Kastner, lead engineer at Etsy)<sup>2</sup>

These accomplishments are not from some distant galaxy far, far away. They are real. Admittedly, there are some important differences between their environments and those of most enterprises: most notably, a clear recognition that technology is not *aligned* with their business — technology *is* their business. They use their incredible technology capabilities to not only enable their business but to dominate their fields. They have also embraced seven habits that are essential to great DevOps results.

## THE SEVEN HABITS TO DEVELOP FOR DEVOPS SUCCESS

Achieving DevOps glory requires you to engage in seven essential habits (see Figure 3). These habits are valuable regardless of where you are in the journey, and you may already be doing some of them, although perhaps not consistently and across your organization. When you pursue them all and exercise that pursuit across organizational boundaries, remarkable improvements are not just possible, but probable.

Some of these will be met with resistance in your organization, and some may spark rebellion. Nevertheless, these behaviors are essential to achieving superior results. Change is hard, but, as the saying goes, doing the same thing and expecting different results is insanity.

**Figure 3** Engage In Seven Habits To Achieve Highly Effective DevOps

- 1 Establish trust and transparency between Dev and Ops.
- 2 See everything through the eyes of the customer.
- 3 Streamline your application delivery pipeline.
- 4 Adopt a loosely coupled service-oriented architecture.
- 5 Reward solution simplicity and reliability.
- 6 Adapt and improve how you use customer experience data.
- 7 Dev and Ops need to walk in the other's shoes.

93781

Source: Forrester Research, Inc. Unauthorized reproduction or distribution prohibited.

### Establish Trust And Transparency Between Dev And Ops

The stereotypes that stand in the way of progress are deeply rooted in the cultures of both Dev and Ops. Largely born of ignorance and isolation, they melt away when Dev and Ops actually work together. Stereotypes dissolve when confronted with the specifics. Just getting Dev and Ops to talk to one another, to work together on a personal level is an important start.

It sounds superficial, but when people talk openly and frequently, they begin to understand what life is like in the other person's world. The worlds of others become less mysterious when we see the similarities and linkages to our own world. People then gain empathy for their colleagues, and empathy is a foundation of trust and (dare we say it?) friendship.



## See Everything Through The Eyes Of The Customer

The best place to start is with a shared mission. Today, Dev is rewarded for delivering code into testing environments, while Ops is rewarded for keeping apps running with acceptable performance and security. The conflict created by these differences in motivation has carved a deep chasm that we must now bridge, if not fill. Rather than rewarding them for different and conflicting actions, align the rewards on customer satisfaction. The business is rewarded that way — why should Dev or Ops be different?

View everything from the customer's perspective. Technology management's traditional view is inside-out; it looks at technology first and customers second (at best). An outside-in viewpoint places customers at the center of attention and delivers what they need, which is not always what they initially desire. They often don't quite know what they need, but open dialogue between partners enables both sides to co-create solutions.

Instilling customer thinking requires leadership to reshape the focus of the organization. Consistent reinforcement of values and changing measures of success coupled with consistent messaging and reinforcement moves the needle, nothing less. As challenging as the change may seem, the benefits are substantial. Today's organizations continually work at cross-purposes, with small internal forces constantly pulling the organization off track. Consistent customer focus at all levels aligns the effort vectors in your organization. The results are transformative. Great leaders inspire their people with the right incentives, and the best always lead by example.

## Streamline Your Application Delivery Pipeline

Manual builds, manual testing, and manual release management are slow, costly, and error-prone. Much of the high-stress fear of releases is related to the inherent unpredictability of manual processes. Hero-worship culture can mask this, but for every heroic event, there is a manual mistake of heroic proportions lurking behind the scenes. Automation can eliminate these errors and make releases a mundane event, and that is a good thing.

Embark on your automation in a pragmatic, gradual manner instead of a “big bang” attempt:

- **Start by automating the build process.** Start here because you will be doing it a lot, and reducing cost and increasing the reliability of something you do frequently will pay off almost immediately. If your product is too complex to automate the build, you need to simplify the configuration and packaging of its components.
- **Steadily increase the amount of API-driven test automation.** Once your builds are automated, gradually increase the amount of test automation, starting with unit tests and gradually building up to integration and system testing. Test automation should be built around component and service application programming interfaces (APIs), not around automated user interface (UI) testing. Increasing your test automation skills is likely to be the biggest barrier, but steady, focused investment will ensure progress.

- **Automate deployment processes in parallel with test automation.** With test automation and frequent builds providing an increasingly real-time picture of application health, work on increasing the automation of your deployment process. Standardize the configuration of environments. Automate the provisioning of environments, and work up to building environments on demand. Once you do this, automate the deployment of applications into these environments.

None of these steps is particularly hard if you were starting from scratch; it's the mess made by your existing inconsistent manual processes that will be hard to untangle. Consistent focus on automation will get you there, and at each point along the way, your processes will become faster, more adaptable, more reliable, and lower cost.<sup>3</sup>

### Adopt A Loosely Coupled Service-Oriented Architecture

Variety is the source of complexity, and complexity leads to fragility, chaos, and unpredictability. Variety can also lead to innovation through experimentation. The key to success is to know when variety is useful and when it is damaging. Variety is useful when trying out new ideas, new alternative solutions. It is useful in research and development but harmful in production.

Companies like Amazon, Google, and Microsoft's Azure business attain dramatic and impressive results partly because they standardize *everything* in their production environments. Their giant server farms, for example, are built on standard configurations. Simplicity and standardization make managing far easier, the services offered far more resilient, and costs far lower. Economies of scale in production, like in manufacturing, rest on a foundation of standardization.

Even when you cannot achieve this level of standardization, reducing variation will simplify problems and reduce costs. Past practices that tacitly fostered a culture of artistic freedom are partly to blame for the complexities of today. The result is a labyrinth of technology that requires constant heroics to keep it functioning. The future of your business cannot tolerate this chaos sustained by personal heroics. To avoid losing your way in this complexity, you must reverse the trend. Standardizing services is the starting point. The evolution will take time; the labyrinth took years to build, and tearing it down is a long-term task. Application modernization and simplification, if deliberate and sustained, will eventually lead you to a simpler and more easily managed future.

Standard components allow you to more quickly and easily develop new production services, which themselves adhere to a finite set of standards. Technology management customizing everything causes additional — and unnecessary — confusion and complexity.

The collection of services is embodied in a service app store or service catalog that is developed and maintained by a good DevOps collaboration.<sup>4</sup> The new breed of Ops professionals will usually take the lead in managing this catalog, acting as the liaison to customers. They will work with Dev on the possible options and collectively engage the customers, determining costs and performance expectations.<sup>5</sup>

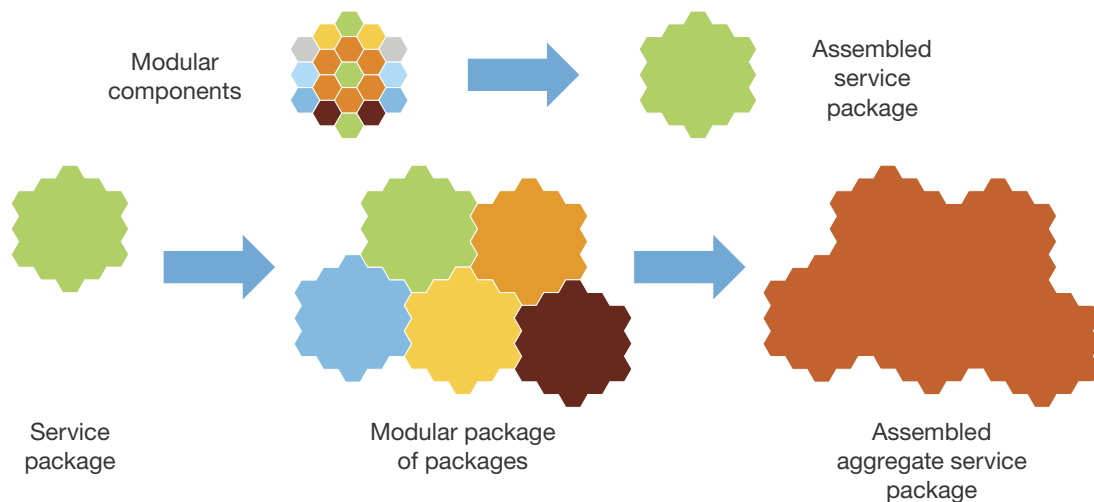
### Reward Solution Simplicity And Reliability

Modular design is the key to good systems engineering (see Figure 4). When you use standard components, assembling them into a new service becomes much easier. You can then package them into a new standard service, which itself can become a component of yet another service defined at a higher level of abstraction.

The beauty of such modular engineering is that it *hides* the complexity within the packages. All products that are produced with sound systems engineering follow the same practice. The complexity within a microprocessor is mind-boggling, yet we need not concern ourselves with it. Technology providers insulate us from this complexity. From the chip level to the motherboard to the computer to the operating systems and all the way up to the applications, each level offers us another layer of protection from the complexity within. A simple email involves an astounding variety of actions to which we are all oblivious, and that's a great thing.

The time to change is now. Applications, services, and the business processes built atop them are getting more complex at an exponential rate. If you are struggling now, the future will be impossible. The only way out is to develop, deploy, and run your apps as if your organization's life depended on it, which, in fact, it does. Implement feedback and feed-forward in the service life cycle. Reward both Dev and Ops for the simplicity and reliability of both the application and the environment.

**Figure 4** Modular Design And Packaging Hide Complexity



## Adapt And Improve Using Customer Experience Data

Just as good systems engineering is crucial to the stability of any complex system, feedback is equally important. Every complex system — whether manmade or natural — has its behavior governed by feedback mechanisms of various sorts. The squeal of a public-address system is an example of the wrong kind of feedback, for example, so circuitry is added and tuned to invert that feedback to squelch the squeal.

Technology and business services are complex systems that need feedback. This is inherent in the design, much of which is hidden from even the developers and designers. The main aspect of feedback that you must address is in the life cycle. In short, it is lacking, woefully in most cases.

Life-cycle feedback comes in multiple forms. The two most notable to DevOps (and the entire service, for that matter) are:

- **Involve Ops in the earliest stages of service conception.** Ops professionals offer intelligence about production environments that may be overlooked by Dev professionals and others involved in the early stages. Also, when they are involved early in the planning process, Ops professionals can be better prepared for the demands they will soon need to fulfill. This latter point means there will be fewer surprises and better service quality.
- **Involve test engineers early, too.** The benefits are similar, as they have plenty of intelligence to add to the early stages of development, and they can also prepare for the coming testing demands.

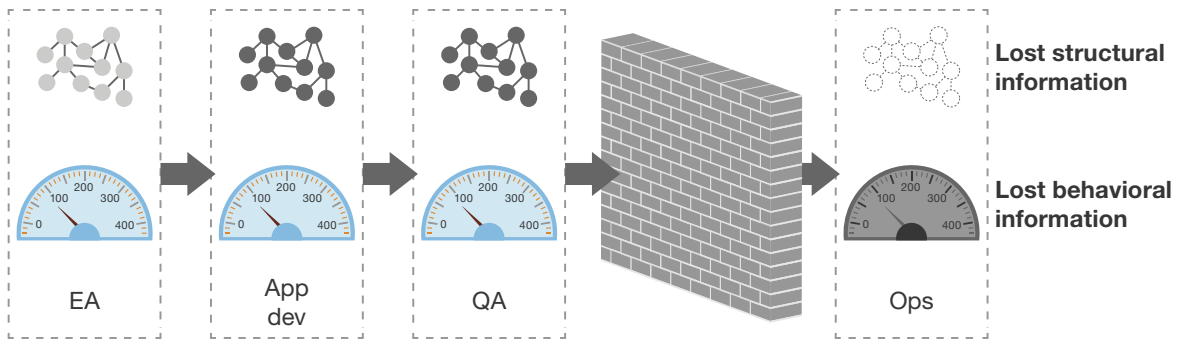
Feed-forward is a similar notion, but one that goes in the opposite direction. The key concept here is one of information preservation through the transition stages in the life cycle. Technology management often sanitizes the information that gets passed along to the next stage, if there is even any information at all. The result is that people in the next stage struggle from a lack of information. This is most unfortunate because the information need exists elsewhere. It just hasn't been passed forward.

A good example of feed-forward is the preservation of application dependency maps (ADMs). Such models map the relevant application components, the relationships between them, and the relationships to certain infrastructure elements, such as servers. Without preservation of this information, Ops teams are forced to use ADM discovery tools to reverse-engineer the applications and obtain this information. ADM tools are wonderful, but their necessity represents a failure of technology management's life-cycle processes.

The life cycle followed in most enterprises is, in fact, not much of a life cycle at all. It is mostly a one-way flow with little or no feedback and limits to the feed-forward (see Figure 5). This flow is riddled with problems. Most notable is the wall prior to Ops, where information is lost, hampering the ability of Ops to manage the services.

A genuine life cycle is rich with feedback and feed-forward mechanisms (see Figure 6). Feed-forward still follows a mostly sequential flow, from one stage to the next. If you skip a stage with feed-forward, the process can get sloppy, negating the value of the other positive aspects of the DevOps flow.

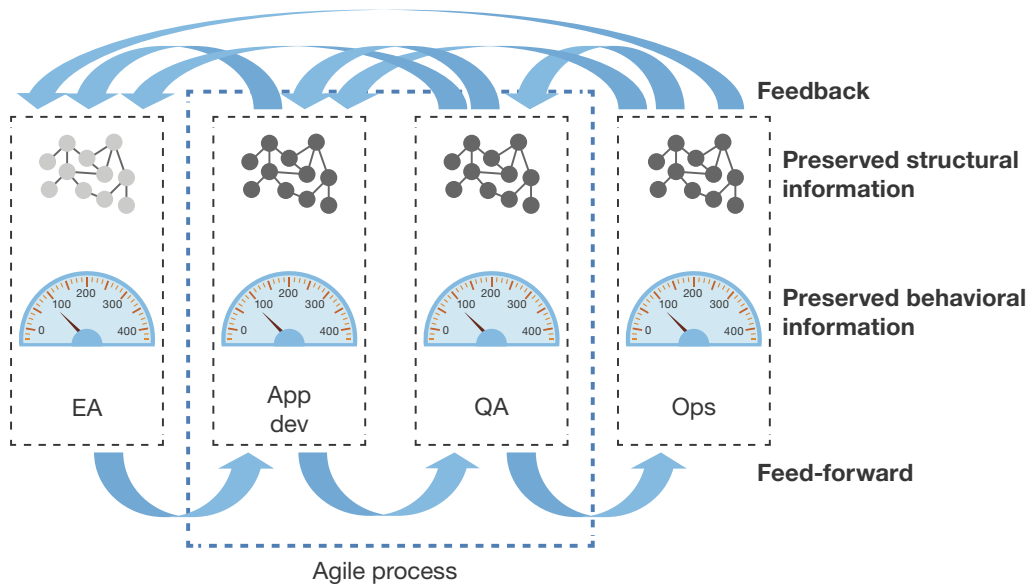
**Figure 5** Applications And Services Too Often Suffer From A One-Way Life Cycle



93781

Source: Forrester Research, Inc. Unauthorized reproduction or distribution prohibited.

**Figure 6** Feedback And Feed-Forward Enable A True Life Cycle



93781

Source: Forrester Research, Inc. Unauthorized reproduction or distribution prohibited.

### Dev And Ops Need To Walk In The Other's Shoes

Increase not only trust and transparency but also an appreciation of what the other group is doing by having Dev and Ops participate in each other's daily activities. For example, put developers on the frontline of support. Application support is ill defined in many organizations. Even where explicitly defined, it is rare to have the actual application developers involved directly in the support of their creations. Developers shy away from support because it distracts them from the creative aspects of their work. You have a great opportunity to change this, although it may require some adept negotiations with these developers and adjustments of their goals.

When developers have to be actively involved in the support of their applications, their work is naturally going to be of higher quality. There are two major reasons for this quality boost:

- **Developers can eliminate problems before they happen.** They will strive harder to produce good quality applications so they can minimize the need to do the dreaded support work.
- **The person who knows the application best can fix it.** When a fix is needed, it's better to go to the person who knows that code best — its creator. Fixes happen more quickly and they are more reliable.

Another example is to put operations personnel in various developer activities. Not only will this foster the feedback and feed-forward mechanism but it will show Ops the mixture of discipline and creativity required by Dev. First, operations must participate in planning meetings to not only get information about how the environments should be designed but also to understand the pressure that developers are under to create meaningful code changes for consumers with limitations set by scope, quality, resources, and time. Second, operations should rotate representatives to attend development daily standups to understand the cadence of the release and the issues that developers face to get their job done. After seeing firsthand any development issues that can be solved through technology, operations can get creative and increase developer productivity.

### DEVOPS IS NOT ONLY POSSIBLE BUT NECESSARY IN BOTH IT AND BT

Solid DevOps practices are not science fiction, nor are they limited to the elite names in web-based business. Your organization can start down the right path now — and must. This is a fitness program designed to transform a lethargic business into a lean and nimble one. Like any fitness program, DevOps will take time, will occasionally be painful, and will never truly end. Also, like any fitness program, the results will be wonderful and evident even in the early stages of your journey.

If you have cloud ambitions of any sort (public or private), these DevOps principles are a necessity. The flexibility and velocity needed for real cloud services are otherwise impossible at enterprise scale. Indeed, the BT vision is also impossible without these principles, whether you call it DevOps or not.<sup>6</sup>

---

WHAT IT MEANS

**INSTANT-ON BUILT-TO-ORDER SERVICES ARE THE FUTURE**

As services get more complex and the methods to produce these services evolve, DevOps principles will become commonplace. Any other approach will appear foolish. In the leading DevOps innovators, this is already the case:

- At Facebook, people wouldn't even consider deploying a production environment the old-fashioned way, with system admins performing the tasks manually via the command line. At its scale and velocity, such an old-school approach is considered an act of stupidity.
- At Perforce, the move to automatically build, test, and deliver a "snapshot" release after each code check-in created a lot of concern around product quality. Now, new development teams start with snapshots from inception.

DevOps is a primary movement in the growing trend to industrialize the old I&O and AD&D role into new modern application delivery and modern service delivery.<sup>7</sup> This industrial revolution of IT is based on standardized systems, with a high degree of automation involved. Whereas prior industrialized movements (e.g., manufacturing, agriculture) mandated extreme standardization, IT industrialization can be much more flexible. The end product of IT must still fall into a well-defined and finite set of possible outcomes. We are on the verge of gaining much more flexibility, though. Some innovative companies (e.g., Etsy) are already proving the value and relative ease of this concept.

When software drives the production of software rather than just the production of matter, you have many more possibilities. In fact, new manufacturing technologies are now making manufacturing more flexible, too.<sup>8</sup> Customers can soon request a wide set of variations on standardized services.

DevOps principles will be central to this world of standardized services. You will never be able to offer the speed, customization, and quality to your customers unless you adopt these principles. In fact, we all look forward to the day when DevOps is not just another buzzword. When people reach this future vision of flexible service design and delivery, it won't be called DevOps. It will simply be considered normal practice. At that point, the DevOps term will graduate to its rightful place as a historical footnote that spawned a turning point in technology.

---

**SUPPLEMENTAL MATERIAL**

Forrester's Business Technographics Global Priorities And Journey Survey, 2014, was fielded to 13,822 business and technology decision-makers located in Australia, Brazil, Canada, China, France, Germany, India, New Zealand, the UK, and the US from SMB and enterprise companies with two or more employees. This survey is part of Forrester's Business Technographics and was fielded

from January 2014 to March 2014. Research Now fielded this survey on behalf of Forrester. Survey respondent incentives include points redeemable for gift certificates. We have provided exact sample sizes in this report on a question-by-question basis.

Forrester's Business Technographics® provides demand-side insight into the priorities, investments, and customer journeys of business and technology decision-makers and the workforce across the globe. Forrester collects data insights from qualified respondents in 10 countries spanning the Americas, Europe, and Asia. Business Technographics uses only superior data sources and advanced data-cleaning techniques to ensure the highest data quality.

We have illustrated only a portion of survey results in this document. To inquire about receiving full data results for an additional fee, please contact [data@forrester.com](mailto:data@forrester.com) or your Forrester account manager.

### Companies Interviewed For This Report

BMC Software	Opscode
CA Technologies	Perforce
Etsy	Puppet Labs
Facebook	salesforce.com
HP	Serena Software
IBM	Unnamed global insurance company
Nordstrom	

### ENDNOTES

- <sup>1</sup> Source: Opscode, "Scaling systems configuration at Facebook - Phil Dibowitz," YouTube, April 26, 2013 ([www.youtube.com/watch?v=SYZ2GzYAw\\_Q](http://www.youtube.com/watch?v=SYZ2GzYAw_Q)).
- <sup>2</sup> Source: Eric Kastner, "Quantum of Deployment," Code as Craft, May 20, 2010 (<http://codeascraft.com/2010/05/20/quantum-of-deployment/>).
- <sup>3</sup> The steps outlined here are just an overview of the journey. For a more complete discussion, see the following book. Source: Jez Humble and David Farley, *Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation*, Addison-Wesley Professional, 2010.
- <sup>4</sup> The following report describes the drivers and uses for app stores. See the July 24, 2014, "[Service App Stores Will Reshape Corporate Tech Management](#)" report.
- <sup>5</sup> The following report addresses many of the principles, processes, and technologies helpful to the pursuit of a service catalog as the linchpin of your service management culture. See the June 12, 2013, "[Master The Service Catalog Solution Landscape In 2013](#)" report.



The following report covers additional details about applying supply-chain management concepts to your service portfolio, an important dimension to a good service catalog. See the February 6, 2012, “[Become Customer-Centric, Service-Focused, And Automated](#)” report.

And on the application development side, see the June 23, 2014, “[Brief: Software Innovation Requires A Loosely-Coupled Application Architecture](#)” report.

- <sup>6</sup> The following report describes the attributes of an organization that fits the highly evolved model of empowered business technology. See the April 1, 2014, “[The Age Of The Customer Redefines Technology Management Archetypes](#)” report.
- <sup>7</sup> Modern application delivery is described in more detail in the Forrester playbook by the same name. See the March 4, 2014, “[Modern Application Delivery Drives Digital Business Success](#)” report and see the September 24, 2014, “[Transform Tech Operations To Better Serve Customers](#)” report.
- <sup>8</sup> Source: Chris Anderson, *Makers: The New Industrial Revolution*, Crown Business, 2012.

## About Forrester

A global research and advisory firm, Forrester inspires leaders, informs better decisions, and helps the world's top companies turn the complexity of change into business advantage. Our research-based insight and objective advice enable IT professionals to lead more successfully within IT and extend their impact beyond the traditional IT organization. Tailored to your individual role, our resources allow you to focus on important business issues — margin, speed, growth — first, technology second.

### FOR MORE INFORMATION

To find out how Forrester Research can help you be successful every day, please contact the office nearest you, or visit us at [www.forrester.com](http://www.forrester.com). For a complete list of worldwide locations, visit [www.forrester.com/about](http://www.forrester.com/about).

### CLIENT SUPPORT

For information on hard-copy or electronic reprints, please contact Client Support at +1 866.367.7378, +1 617.613.5730, or [clientsupport@forrester.com](mailto:clientsupport@forrester.com). We offer quantity discounts and special pricing for academic and nonprofit institutions.

---

## Forrester Focuses On Infrastructure & Operations Professionals

You are responsible for identifying — and justifying — which technologies and process changes will help you transform and industrialize your company's infrastructure and create a more productive, resilient, and effective IT organization. Forrester's subject-matter expertise and deep understanding of your role will help you create forward-thinking strategies; weigh opportunity against risk; justify decisions; and optimize your individual, team, and corporate performance.

« IAN OLIVER, client persona representing Infrastructure & Operations Professionals

