puppet The shortest path to better software.

The CIO's Guide to DevOps

Contents


Software enables
every business...

...so every company must behave like a software company.

Stop wondering if your business will be disrupted. It's happening. Every aspect of our economy and society is being changed by digital transformation. Countless globally scalable startups are trying out and discarding new business models faster than most companies can launch a new application.

That’s because software drives every business now. It’s creating disjointed — and sometimes incompatible — initiatives inside companies. It’s increasingly common to find the chief digital officer (or as McKinsey says, “transformer in chief”) leading digital integration across the company, forcing the traditional CIO to play catch up — or worse, sit on the sidelines.

Whether you’re selling high-speed commodity trading services or high-thread-count organic sheets, the customer experience is driven by software. The speed with which your team can deploy new software has become inextricably linked to customer acquisition and retention.

If your software gets hung up in the testing stages; if you can't scale effectively from test to full production; if your services keep breaking — you're going to lose customers. And when that happens, a competitor who's already overcome old patterns and learned to operate like a fast-moving software company is going to jump in and give those customers what they want.

These torrents of change are pushing CIOs to embrace digital transformation — to transform IT into an innovative, fast-moving software organization. It's the CIO's responsibility to empower the business with technology that will propel it past competitors and beyond their reach.

As CIO, you have a choice: ignore digital transformation and supporting movements like DevOps, and get disrupted into irrelevance — or clearly understand what's at stake, and lead your organization through its digital transformation.

“Someone is moving faster than you are, and probably much faster. That someone could be a competitor, and if they iterate this much faster than you do, you’re toast.”

Matt Asay of TechRepublic,
“Why the DevOps faithful keep pulling away from their competitors**”**

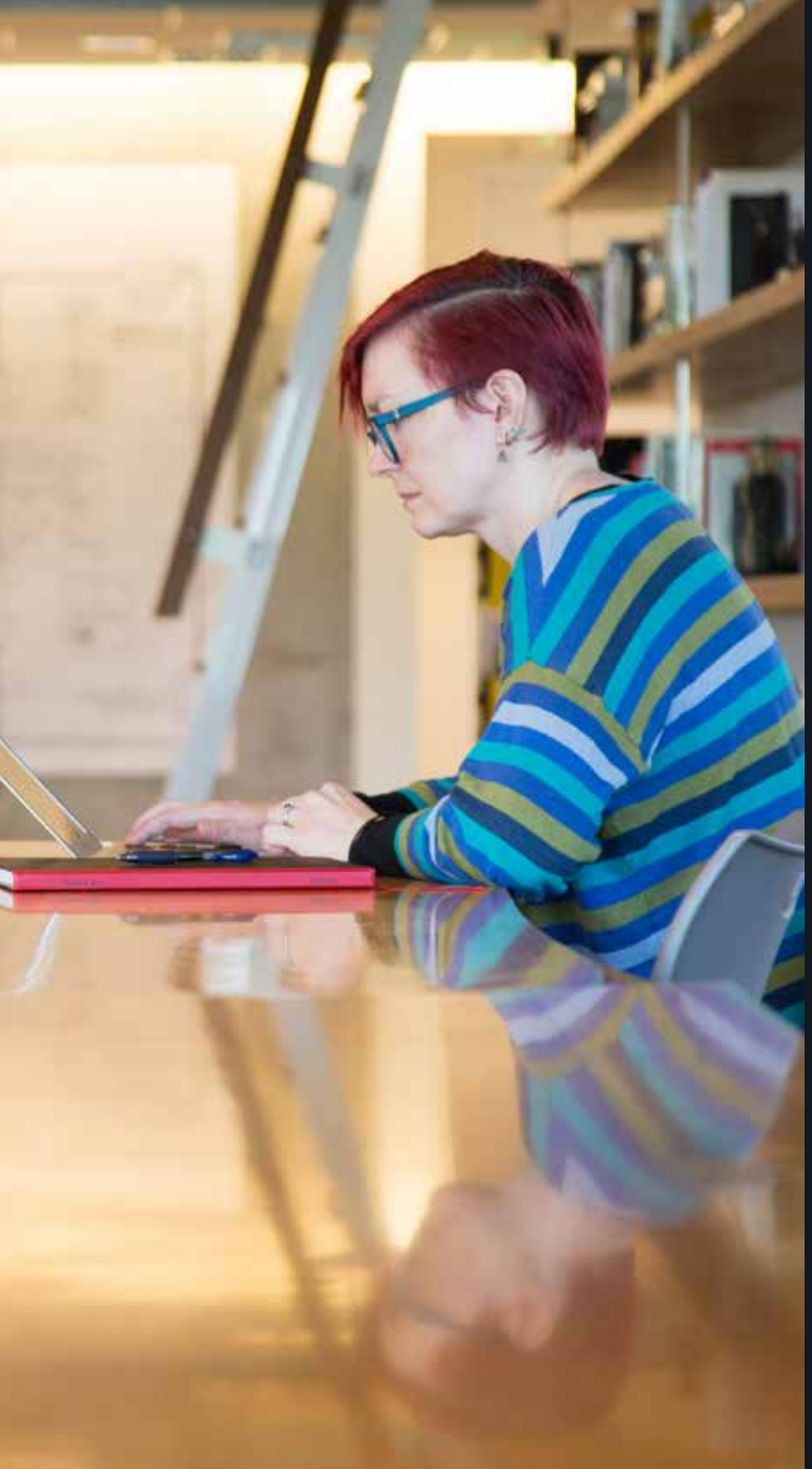
The old ways don’t work anymore

Traditional ways of managing software and the infrastructure that runs it — the endless meetings, layers of approvals, change ticket backlogs and accumulating cruft — won't cut it in a world where companies run in the cloud, quickly adopting new technologies such as Docker containers, or container-cluster managers like Kubernetes and Mesos.

Your IT organization needs to respond quickly to market changes by making more experiments faster, learning from frequent customer feedback, and then using that learning to rapidly deliver new features. To do that, you need to build your change fitness: your organization's ability to transition gracefully from current practices to new practices, leaving behind old tools and processes as you adopt new ones to serve your business better.

Once you boost your change fitness, IT becomes a streamlined, standardized and forward-thinking organization. Instead of being the last one invited to the meeting, you're leading and driving the business. IT gets to influence the customer experience in a more powerful and direct way than any marketing campaign ever could. Ultimately, there won't be a single project at the company that doesn't rely on IT as a strategic enabler or leader. You'll be prepared to meet digital disruption head-on, and perfectly positioned to create new opportunities for your company, faster.

How do you transition from the old bureaucratic ways to the new agility? It's called DevOps.



You want digital transformation? Look to DevOps.

Traditional IT organizations aren't built for this new world. They are organized around functions, not around business value, so they're siloed, with each group using different tools and processes. DevOps isn’t a fixed methodology or process; it’s a community of practice and a set of principles.

DevOps arose originally among practitioners frustrated by the old siloed ways. But plenty of IT managers and C-level executives also recognized early on that DevOps — a natural outgrowth of Agile methodologies and lean principles — could help the business move faster by cutting time-wasting processes and creating shorter feedback loops with the customer.

“Of course, if you’re slow to act, unresponsive and focused on just keeping the lights on, then your organization will find a different kind of CIO.”

Mark Shapland,
CIO.com, March 2015

You'll find quite a few principles and methods from Agile and lean thinking in DevOps — for example, limiting work in progress and working in small batches; pair programming; continuous delivery; including quality measures early in the development cycle; test-driven development; automation of just about anything that can be automated; and monitoring, coupled with making metrics visible to everyone. Continuous learning and improvement are also a core part of DevOps, so you'll see practices like developers and IT operations people attending each other's standups and planning meetings, and regular retrospectives.

Getting the right processes and tools in place vastly improves software quality. But they deliver other important benefits, too. The DevOps toolchain and practices improve how teams work by promoting greater collaboration and faster, more frequent feedback loops. Faster feedback lets teams learn much more quickly from operational results and from customers, so they can build technical and process innovation into the software release cycle — and delight customers, too.

Infrastructure as code: a cornerstone of DevOps

DevOps is about more than simply adopting the right tools and improving processes. It's also about shifting the culture of the organization away from siloed responsibilities to a pattern of ongoing communication, sharing and trust between individuals and teams.

That's why the practice of managing infrastructure as code is so fundamental to DevOps. When you describe all infrastructure configurations and changes in code, you can version that code, review, integrate, test and deploy it with the same tools your organization uses for software. This makes it much easier for your infrastructure and development engineers to collaborate. And IT can shift away from its traditional role of gatekeeper to the new role of enabler, by creating self-service portals for developers and testers.

“Automation is the best documentation. Gone are the complex and out-of-date documents explaining how to create and install systems. They have been replaced by code which is used to automatically and repeatably create our critical IT systems.”

Geoff Clitheroe,
systems development manager for GeoNet,
New Zealand's national geological event alerting service

Infrastructure as code enables all of these practices:

- **Version control and peer code review.** This enables more teammates to review and collaborate on code earlier in the cycle. These practices build trust in the quality of the code, and the process of creating it. Version control assures that code can be rolled back to its last known good version. Commonly used tools include Git, Mercurial, Perforce, Subversion, and Team Foundation Server
- **Configuration management.** Managing configurations across all environments assures that everyone is developing and testing code in environments that match production, building trust in code quality and confidence that deployments will be successful. Configuration management through code also ensures that all infrastructure changes are reviewable, and that you can see the differences caused by changes. Puppet is the most-used tool for managing configurations as code.

- **Continuous integration.** When you develop in small batches and continually integrate these changes, you ensure that every change works within the code base. It's also much easier to roll back to the last known good version when necessary. Like other tools listed above, continuous integration also builds confidence in deployments. Commonly used tools for continuous integration include Jenkins, Bamboo, Hudson, Go, TeamCity, Travis CI, and Visual Studio Team Services.
- **Peer code review**. It's been shown that lengthy and bureaucratic review-by-committee slows down development, and that it doesn't produce better code. Peer code review, on the other hand, *does* result in better code, and it's faster. Common tools include Git, Stash and Gerrit.
- **Monitoring.** Along with displaying the results on dashboards where everyone can see them, monitoring exposes everyone to the results of the work. Making data public and visible allows the team to hold itself accountable, to immediately see the results of experiments, and also allows management to understand what's going on. Monitoring enhances communication and trust. Popular tools include New Relic, Nagios, Splunk, AppDynamics, Loggly, and Elastic.

- **Automated testing and deployment.** For consistent, repeatable code, you need automation. Automated testing makes the process both faster and more reliable; automated deployment is much faster than manual, and less error-prone. Together, automated test and deployment deliver better release velocity, while reducing big fires. That frees your engineers, allowing them to focus on improvement and innovation. The tools for automated testing are legion, and depend on what you're testing and which phase of testing you're talking about. Tools for automated deployment include Capistrano, MCollective (part of Puppet Enterprise), Octopus Deploy, Visual Studio and Nomad.

By now you're probably thinking, "Show me the numbers." If you're going to consider DevOps as the path to digital transformation for your organization, you need to know how likely it is to deliver success.

“If we have the package already developed, what took days before now takes literally minutes. For a new capability — say a new application server container we need to install — what took several weeks is now down to a week.”

Jeff Quaintance,
senior cloud & automation engineer, Staples

The metrics of DevOps: proof that it works

Over the past five years, we've surveyed more than 25,000 technical professionals worldwide to better understand how the technical practices, cultural norms, and lean management practices we associate with DevOps affect IT performance and organizational performance.

We have discovered that DevOps culture and practices can deliver significant improvements in IT processes and business outcomes.

The 2016 DevOps survey analysis, for example, showed that high-performing organizations — which employ DevOps practices and live DevOps culture to a high degree — far outperform others, as measured by throughput and reliability.

- **High performers deploy 200 times more frequently than lower-performing peers.** That gives the DevOps organizations opportunity to experiment more, learn more and get new features out to customers more rapidly. Organizations that develop the muscle and habits of frequent deployment have a distinct competitive edge: In a race to get a new service or capability to market, the team that's good at frequent deployment is likely to be first.

- **Their change failure rate is three times lower, and they recover from failures 24 times faster than lower-performing peers.** High performers reported a mean time to recover of less than one hour, while respondents from other organizations reported it takes as much as a day to recover. Being able to make changes so much more reliably lets high-performing orgs move faster, and with a high degree of confidence — confidence that inspires even greater willingness to perform experiments and learn.
- **They create more shareholder value.** We analyzed 2014 survey results from 1,000-plus respondents who volunteered the names of companies where they worked. We found that **publicly traded companies with high-performing IT teams had 50 percent higher growth in market capitalization over three years** than companies with low-performing IT teams. Respondents from high performers also reported their organizations were nearly twice as likely to exceed their own targets for profitability, market share and productivity.

While these findings are dramatic, they should not surprise anyone who understands that DevOps enables tighter customer feedback loops, enabling a company to learn more quickly what's working for customers and what isn't. DevOps also enables organizations to release higher-quality code, eliminating a lot of pullback and rework, because quality is built in from the beginning. At another level, DevOps makes it much easier for organizations to quickly cycle old technology out and new technology in — a critical competitive capability.

With technology research firms like Gartner and Forrester issuing reports on the advantages of DevOps, it's not surprising that leading companies no longer consider DevOps a fad. Instead, DevOps is increasingly seen as a highly effective and desirable set of skills and behaviors — a must for companies wanting to gain and hold a leading position in their markets and industries.

“We’ve had some great results from our DevOps initiatives so far. We reduced our cost per release on one application by 97 percent … By automating our testing, we’ve reduced multiple man days of effort down to an overnight hands-free process.”

Jonathan Fletcher,
enterprise architect and lead for technology, platform and DevOps at Hiscox

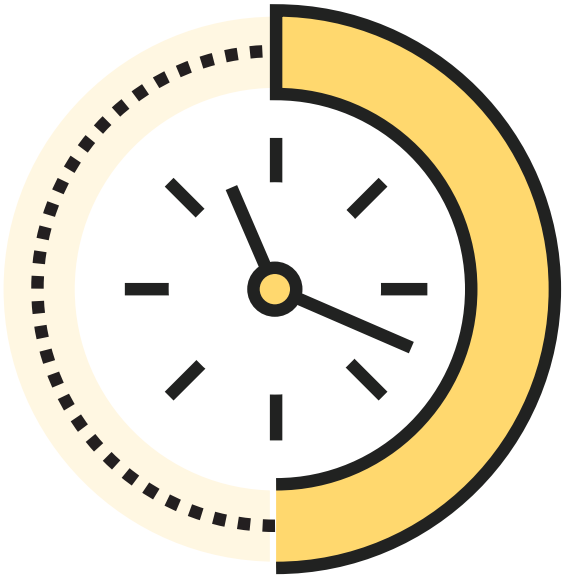
Don’t fear the DevOps: security and compliance

Most of us who have built careers in IT are cautious by nature — even a tad conservative. We know just how badly things can mess up. And it seems like every week there are more headlines to justify our caution. Emails between entertainment executives get hacked. An airline loses millions when flights are cancelled due to the failure of a single router. A system failure causes a huge telecom to send wildly inaccurate bills to customers, triggering a deluge of customer calls, service cancellations and all-too-public complaints.

So it's natural for some to view DevOps with skepticism, wondering if faster code throughput will cause more errors or compromise security.

In fact, the opposite is true. Organizations that adopt DevOps reduce downtime; release cleaner code because they find (and correct) flaws earlier;
and improve both their regulatory compliance and their ability to report to auditors, by improving transparency and trackability.

The 2016 State of DevOps Report showed that high-performing IT teams — which make extensive use of DevOps practices — spend 50 percent less time remediating security issues, compared to lower-performing IT organizations. That's because they build security in from the beginning by including security colleagues in the planning process, and by building security policies and tests into software development and infrastructure architecture.



50%

less time remediating security issues

DevOps is great for talent recruitment and retention

Smart developers, operations people, network and database admins need and want to work with the latest tools, and to work on challenging problems. Younger employees, in particular, expect to see the results of their work quickly, and the tighter customer feedback loops that DevOps enables give them that satisfaction. If employees don't get what they want at your company, there's another one down the road where they will get it — and that's where they'll go.

Most executives understand that highly engaged employees contribute more value. In fact, research has shown that **companies with highly engaged workers grow their revenues 2.5 times as much as companies with low employee engagement.** Publicly traded companies with a high-trust work environment (a characteristic of DevOps organizations) outperformed market indices by a factor of three over a 14-year period.

Today's engineers want and expect DevOps, with its emphasis on close collaboration and continuous learning. DevOps practitioners want to stay up to date, so they question the status quo, look for better ways to do things, experiment with tools, and alert you to when you're at risk of falling behind. These are the kind of engaged employees you need to stay competitive.

The 2016 State of DevOps Report shows that employees in high-performing teams are twice as likely to recommend both their team and their organization to a friend as a great place to work. So DevOps can also help you recruit more smart people to grow your team.

2.2x

Employees in high-performing teams were more likely to recommend their organization as a great place to work.

“It’s great that AON cares about the cultural change of DevOps, and invests in the technology to support that change. We have a mature, modern delivery system that most enterprises don’t have. I can use that for recruitment.”

Glenn Mason,
solution architect at AON

DevOps and your change fitness level

You can never really know what changes will come at you from the market, from your company leadership, or the overall economy. All you really know is that change will come, and you need a tech organization that's ready to meet it.

Companies that adopt DevOps culture, processes and tools get very good at managing change. They become both disciplined and flexible: able to evaluate new conditions, brainstorm ideas, come up with a plan, and execute on that plan. These teams experiment, evaluate results, and move forward based on their learning.

DevOps builds the muscle and flexibility you need to cycle new technology in and old technology out as business needs evolve. That's why DevOps is no longer a cult movement, but a proven way to succeed in fast-moving markets.

“You’re more reluctant to pull something back that took six months to release rather than just a couple of weeks to release, regardless of its actual business value.”

Juan Rivera,
manager of storage and platform engineering at Getty Images

Introducing and enabling DevOps culture: it's a journey

Adopting new tools and processes can look daunting. But as it turns out, the hardest part of digital transformation turns out to be changing people's behavior. Change is scary. No one wants to fail. So adopting a mindset that tolerates failure and encourages experimentation is exactly what you need to transform your company.

That mindset is the culture of DevOps. Beyond introducing automation, version control and other technology changes, it's critical to shift your organization's culture to one of continual experimentation and learning, backed by the accountability that comes from exposing performance metrics to everyone. In DevOps, the practices support the culture, and the culture supports the practices. You can't have one without the other, and you can't transform your company without both.

Companies that successfully implement DevOps don't do it all at once — they embark on a journey of organizational transformation. This journey is, naturally, different for every company, but there are still distinct stages:

1. Discovery
2. Standardization of tools and processes
3. Continuous iteration and improvement
4. Implementing business strategy through technology innovation

1. Discovery

Before you make any changes, you need to discover the state of your technology organizations, including their tools, processes and practices. This is the time to ask penetrating questions.

- How do we release code? Who's involved, and what do they do?
- How long does it take to for developers to get the environments they need?
- Do we use version control? Which system do we use, and what's kept in version control?
- Do we use continuous integration, and to what extent?
- How often do we build?
- How often do we deploy?
- Which elements of our software cycle are automated, and which are manual?
- At what point in the software cycle are our security and compliance policies and processes involved?

The answers to these questions should help you understand what to tackle first. If not, ask your technical managers where they think the greatest obstacles are to faster, easier deployments.

2. Standardization of tools and processes

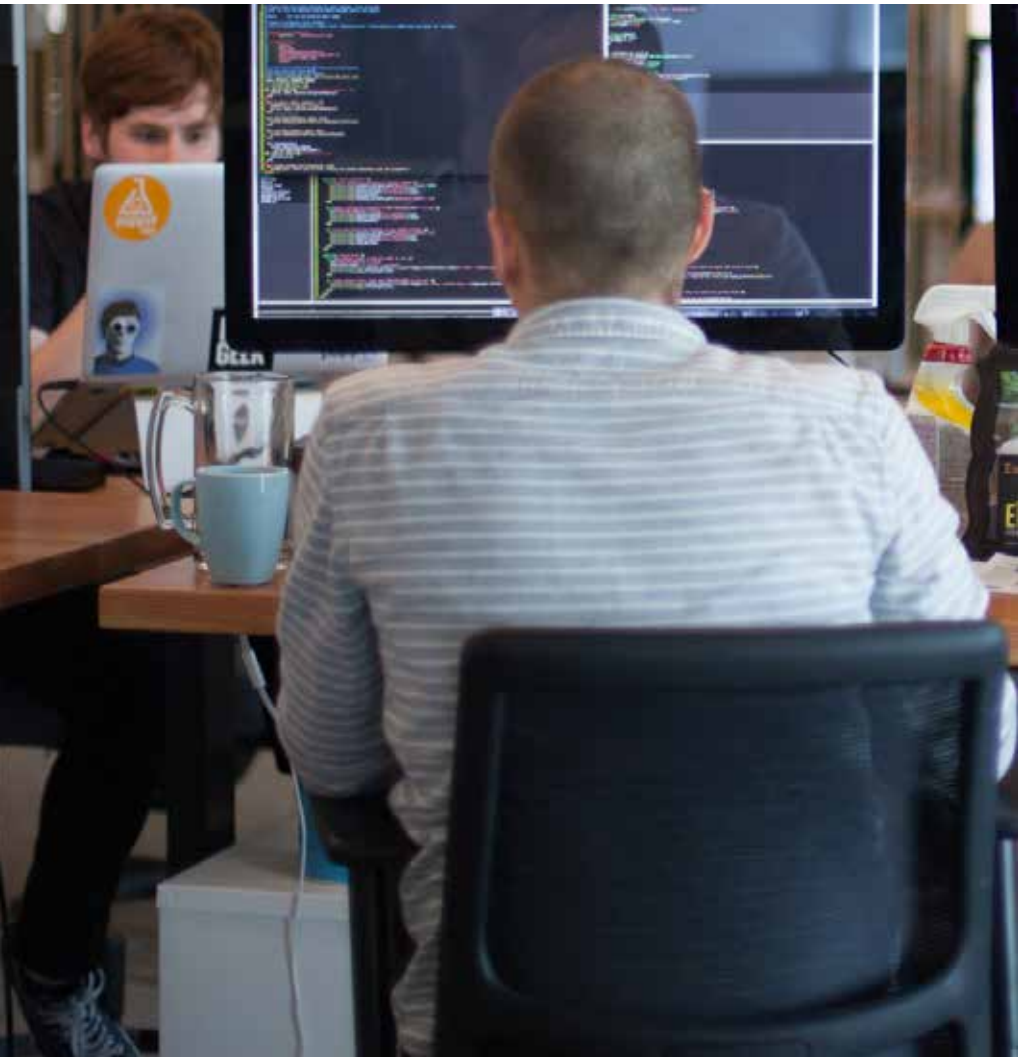
Now it's time to eliminate ad-hoc processes, and standardize the tools and processes used in your technical teams. What to change first really depends on the answers you got to your questions about what your teams are doing and where the greatest blockages exist. Here are the top changes to make at this stage.

- **Automate.** Technical teams are often so buried in manual work, they can't even think about how to improve efficiency. DevOps experts often recommend that you start by automating something small but painful. This can improve morale and free people up to think about what's next. Make sure environment configuration gets automated so that development environments and test environments match production.

- **Get everything that's needed to produce a build under version control.** This includes application code, infrastructure code, deployment scripts, test scripts, etc.
- **Implement continuous integration** if it's not already in use.
- **Make metrics visible.** Insist on displaying in workplace areas your organization's key quality and productivity metrics, plus the state of work in progress.
- **Build self-service platforms.** Developers should be able to provision themselves with correctly configured environments. Same for test and QA. And other departments, such as finance and HR, should also be able to safely provision themselves with tools in a way that's consistent with your organization's security policies. This is a later stage for many organizations, but some are able to begin building these platforms pretty quickly.

Cultural changes must also take place at this stage. Teams that have successfully adopted DevOps recommend a number of practices that help improve understanding and communication between technical teams, and that will help your teams see the how and why of standardizing processes and tools. Encourage or mandate these behaviors:

- Development teams should invite IT operations people to software planning meetings.
- Operations should invite developers to planning meetings.
- Have operations hold blameless postmortems after incidents, and invite developers to these meetings.
- Encourage developers and IT operations people to shadow the other team.
- Institute peer review of changes, rather than having a change approval board review changes.



- Have managers publish progress reports. This will surface any potential blockers early, and also encourage teams to share successes with each other and with the company.
- Bring your security people into software planning meetings and infrastructure planning meetings. The earlier security is involved, the better you can build policy into your software and infrastructure.

You can get more suggestions for implementing and supporting the cultural changes of DevOps in **Get Started with DevOps: A Guide for IT Managers**.

3. Continuous iteration and improvement

Many companies start their DevOps journey with a specific initiative, or by creating a new team to implement and prove a new set of tools and processes. Once that initial DevOps project has shown success, it's time to spread the goodness further: to other projects, to other teams, or both.

How you do that depends on how you started and what your business needs most. Hiscox, a global speciality insurance company based in the United Kingdom, started with a newly created DevOps team and a new application. After this team proved that the process worked — **they reduced the cost of release by 97 percent** — Hiscox has spread its DevOps learnings to its VMware virtualized infrastructure and cloud infrastructure. Eventually, DevOps will just be how Hiscox does everything, including the use of Puppet to deploy every type of software Hiscox runs.

Some companies start with one type of environment and then expand to another. Walmart first deployed Puppet to its in-store servers. After **successfully deploying Puppet to more than 49,000 Linux machines**, the retail giant now in the process of deploying to its 50,000-plus Windows servers.

FINRA's DevOps initiative started with the regulatory agency's desire to move its entire application portfolio from its on-premises data centers to Amazon Web Services. FINRA knew it would need to treat infrastructure as code in order to make that transition and to manage applications successfully in AWS; hence its decision to adopt Puppet. **The agency was able to move two applications to AWS within six months,** and then develop self-service infrastructure provisioning for developers in AWS. Using the same infrastructure code for all of its infrastructure, whether on-premises or in AWS, has allowed FINRA to port the strict security standards it applies in on-prem data centers to its AWS infrastructure, and to take new server tagging methods learned in AWS and apply them to the agency's on-prem servers.

In your own organization, you'll want to ask some questions at this stage:

- Which pieces of deployment and steps in testing are still being done manually?
    - Get these automated, starting with the pieces that cause the most blockage.
- How successful are our deployments, as measured by error rates, service outages, unplanned work and other important quality metrics?
    - Here's where you may discover some gaps in monitoring. Fill them. Add more automation where needed to correct deployment issues.
- Which tools in our toolchain can be integrated?
    - Examples: Integrate deployment and chat tools so the teams involved with deployment can work together more efficiently. And make sure to integrate monitoring and configuration management.

4. Implementing business strategy through technology innovation

Finally, your team gets to work on new, high-value projects! By now, you should have a list of ideas for what to improve — both your own ideas and ideas from your technical teams — tied to your organization's growth strategy. These might include any of the following:

- Migrating workloads to the public cloud.
- Building self-service portals for developers and other teams (such as accounting and HR).
- Expanding automation to legacy environments or other functional areas in the organization.
- Replacing monolithic management platforms with your newly established DevOps toolchain.
- Adopting containers, microservices or other technologies to increase your velocity.

There may already be pockets of DevOps culture and practice in your organization. After all, the movement was originally driven by practitioners facing exponentially increasing workloads and static or shrinking resources. You should seek out those parts of your company where DevOps is happening, learn what people are doing, and socialize these efforts to the rest of your organization. DevOps can grow much more quickly when executives champion the transformation.

“The business value of self-service is to allow development teams to rapidly respond to emerging business needs. Puppet allows us to delegate a lot of these infrastructure tasks — traditional sysadmin tasks — to development teams, so that they can stand up their own servers, configure their own servers, and deploy their application code, all through the magic of Puppet.”

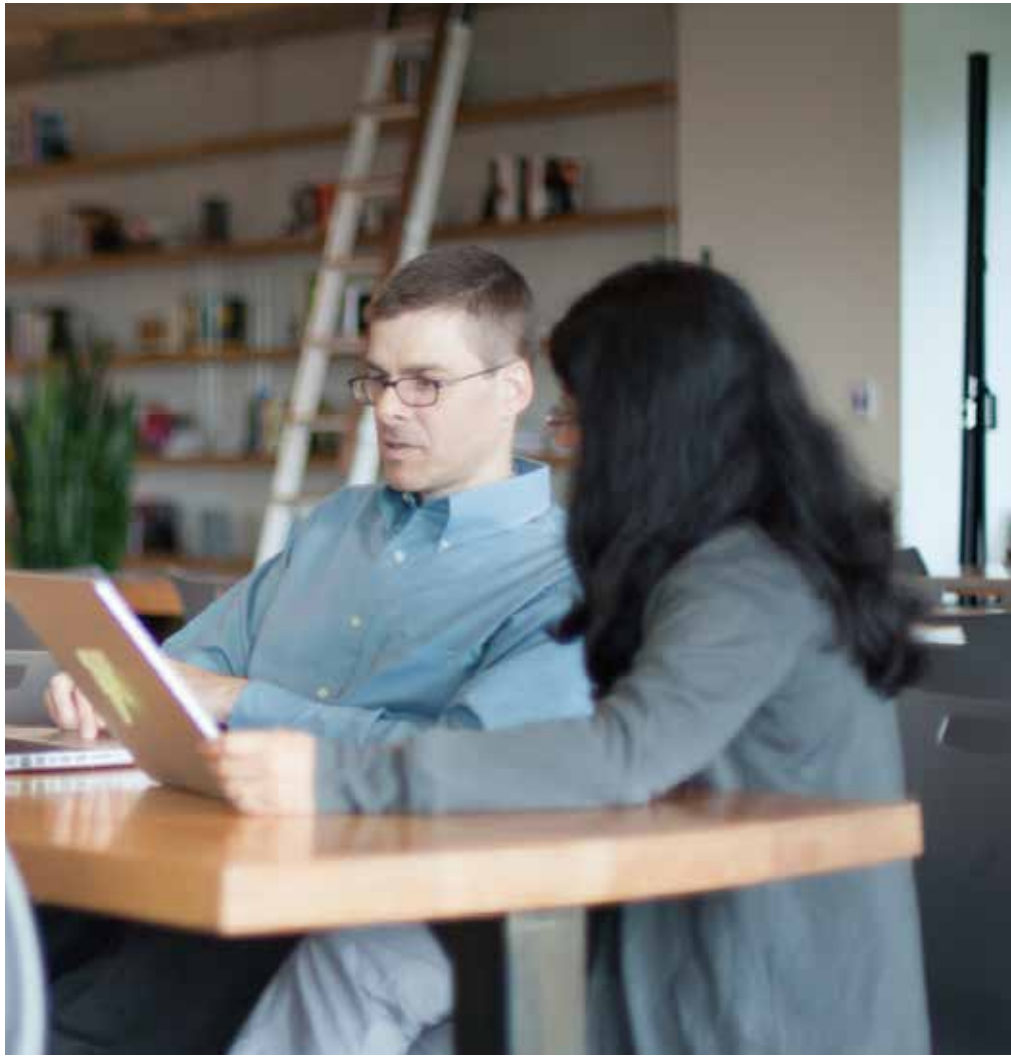
Peter Magnaye,
director of systems engineering at FINRA

Puppet and DevOps.

Puppet isn't the only thing you need to adopt DevOps. It is, however, the cornerstone of any successful DevOps initiative, providing both powerful automation and the ability to define your infrastructure as code. Automation is the foundation of DevOps, and Puppet is the industry standard for automation in **every** part of your technology platform — across operating systems; from laptops and workstations to physical and cloud servers; from switches and routers to load balancers and storage devices; from big on-prem data centers to public cloud services, and private clouds, too.

Today's IT leaders use Puppet to manage and optimize the entire software delivery pipeline — from initial provisioning of dev environments to version control and continuous integration, through every stage of automated testing, to automated deployment. When it's time to scale from hundreds to thousands to tens of thousands of machines, Puppet assures you'll scale consistently, reliably and verifiably.

Unlike other IT automation software, Puppet was built from the ground up to support infrastructure as code: the practice of managing IT infrastructure with the same tools and practices used in software development. With Puppet, you can version-control all aspects of your IT infrastructure, employing the human-readable Puppet language. Puppet is easy to learn, and it works across a huge range of operating systems, major equipment vendors and cloud providers. Puppet lets all your technical staff collaborate on their work, across the data center and throughout the entire software delivery cycle. That collaboration is the heart of DevOps.




Shifting to DevOps can seem daunting — but it doesn’t need to be.

With Puppet as your foundation, your team can start with one initiative, adopt new practices, and expand from there. The same language, methods and even code modules you use at the outset can be applied again and again to bigger projects. Your team will have the power and flexibility to surf the next new technology wave — and the ones that keep rolling in over the years to come.

Resources

2016 State of DevOps Report

The fifth annual industry-leading survey of IT professionals discusses the state of deployment, security, stability and employee loyalty at organizations that have (or have not) adopted DevOps. The report digests findings from the most recent survey of more than 4,600 professionals, plus learnings distilled from more than 25,000 respondents worldwide over the past five years.

Cutting-Edge IT: Moving from Nodes to Applications at Wells Fargo

Read about the DevOps journey that led one of the world's largest banks to shift its IT organization from a server-centric view to an application-centric view. Now Wells Fargo can use code to model its applications and business processes in alignment with business goals.

Hiscox Reduces the Cost of Release by 97% with DevOps and Puppet

A global specialty insurance company details its DevOps journey, including initial steps, how it overcame obstacles, and its plans for spreading DevOps throughout the organization.

Get Started with DevOps: A Guide for IT Managers

Practical advice for managers of IT teams on how to work effectively and empathetically with their own teams, development teams and upper management.

About Puppet, Inc.

Puppet is driving the movement to a world of unconstrained software change. Its revolutionary platform is the industry standard for automating the delivery and operation of the software that powers everything around us. More than 32,000 companies — including more than two thirds of the Fortune 100 — use Puppet's open source and commercial solutions to adopt DevOps practices, achieve situational awareness and drive software change with confidence. Based in Portland, Oregon, Puppet is a privately held company with more than 400 employees around the world.

Learn more at **puppet.com**.