2014
STATE OF
DEVOPS
REPORT

2014 State of DevOps Report

Table of Contents

EXECUTIVE
SUMMARY

The 2014 State of DevOps Report by Puppet Labs, IT Revolution Press and ThoughtWorks is an analysis of more than 9,200 survey responses from technical professionals around the world, making this the largest and most comprehensive DevOps study to date.

Last year, we found that DevOps adoption was accelerating, and that high-performing IT organizations were more agile and reliable, deploying code 30 times more frequently with 50 percent fewer failures. This year, we introduced more questions so we could gain a better understanding of the impact IT performance and DevOps practices have on overall organizational performance, using metrics that matter to the business: profitability, market share and productivity.

Ours is the first scientific study of the relationship between organizational performance, IT performance and DevOps practices. While we are used to hearing stories of DevOps success, the results surprised us. Our analysis of the survey data shows:

- **Strong IT performance is a competitive advantage.** Firms with high-performing IT organizations were twice as likely to exceed their profitability, market share and productivity goals.

- **DevOps practices improve IT performance.** IT performance strongly correlates with well-known DevOps practices such as use of version control and continuous delivery. The longer an organization has implemented — and continues to improve upon — DevOps practices, the better it performs. And better IT performance correlates to higher performance for the entire organization.

- **Organizational culture matters.** Organizational culture is one of the strongest predictors of both IT performance and overall performance of the organization. High-trust organizations encourage good information flow, cross-functional collaboration, shared responsibilities, learning from failures and new ideas; they are also the most likely to perform at a high level. These cultural practices and norms found in high-trust organizations are also at the heart of DevOps, which helps explain why DevOps practices correlate so strongly with high organizational performance.

- **Job satisfaction is the No. 1 predictor of organizational performance.** We all know how job satisfaction feels: It's about doing work that's challenging and meaningful, and being empowered to exercise our skills and judgment. We also know that where there's job satisfaction, employees bring the best of themselves to work: their engagement, their creativity and their strongest thinking. That makes for more innovation in any area of the business, including IT.

Executive Summary

While most organizations realize that improving IT performance is critical to long-term success, until now, it has been unclear which investments truly move the needle. Our findings suggest that high IT performance provides a real competitive advantage, with hints that it plays a role in boosting the value of publicly traded companies, as measured by market capitalization.[1]

Our goal is to provide a picture of how DevOps works today, based on real data from real people. We hope this picture will help IT managers and practitioners understand how to build greater value in their teams — and help their companies win in the marketplace.
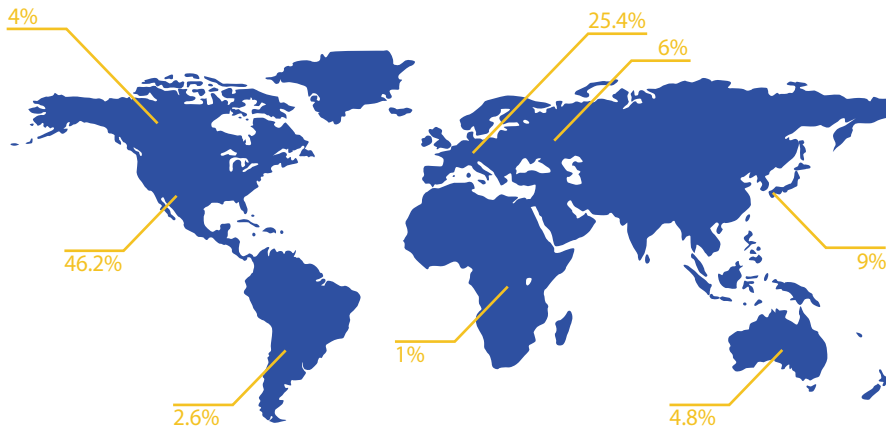
[1] We performed additional analysis on responses from just over 1,000 people who volunteered the names of the companies where they worked, and whose companies are publicly traded. We found that these 355 companies all outperformed the S&P 500 over a three-year period. The publicly traded companies that had high-performing IT teams had 50 percent higher market capitalization growth over three years than those with low-performing IT organizations.

While these results are preliminary, and may not represent a larger general trend, they do suggest that companies paying attention to DevOps practices see organizational performance benefits, and that firms actually investing in DevOps see significant gains compared to their peers. We are continuing to collect data for publicly traded companies, which will allow us to make much stronger predictive analyses and bolder claims, so stay tuned! (And if you took the original survey and haven't shared your company's stock ticker and exchange yet, please do!)

WHO
TOOK THE
SURVEY

More than 9,200 people from 110 countries (more than twice as many people as our previous survey) responded to our December 2013 survey, making this the largest DevOps survey to date.

Overview

4%

25.4%

6%

46.2%

9%

1%

2.6%

4.8%

Industry

Technology	22.7%
Web Software	10.9%
Education	7.5%
Finance/Banking	7.4%
ENTMT/Media	6.8%
Consulting	5.9%
Telecomunications	5.7%
Government	4.5%
Retail	3.7%
Healthcare	3.0%
All Others	21.9%

While the majority of survey participants work in the technology and web software industries (23 percent and 11 percent, respectively), there were strong showings from other industries as well. Education (8 percent), banking and finance (7 percent) and entertainment and media (7 percent) are just a few of the industries represented in our survey.

Organizational Characteristics

Respondents came from organizations of all sizes: from tiny startups to 10,000-employee companies, from shops with fewer than 100 servers to large enterprises with more than 10,000 servers under management.

In terms of organization size, the plurality of responses — 27 percent — came from companies with 500 to 9,999 employees. With respect to the number of servers managed, the majority of respondents (51 percent) said their infrastructure included fewer than 500 servers. Just 13 percent said their organizations had more than 5,000 servers.

Demographics

In the 2012 survey, 57 percent of respondents said they worked in IT operations, and more than 33 percent were in development/ engineering. Most remaining respondents fell into the "other" category. In the 2013 survey, we grouped the most common "other" responses to better understand the job roles of our respondents, and how IT and development functions are structured in organizations.

Company Size by # of Employees:

1-4	5.8%
5-9	3.6%
10-19	5.8%
20-99	17.1%
100-499	21.8%
500-9,999	26.8%
10,000+	15.8%
I don't know	2.1%
Not applicable	1.3%

Size of IT infrastructure by # of servers

< 100	28.3%
100-499	23%
500-1,999	16.9%
2,000-4,999	8.4%
5,000-9,999	4.9%
10,000 >	8.5%
I don't know	8%
NA	2%

Almost a third of 2013 respondents (30 percent) worked in IT operations departments. A similar number (29 percent) were in development/ engineering departments. Out of all respondents, 83 percent identified as practitioners: admins, engineers, architects, developers, etc. Another 14 percent were managers, directors or executives. C-level executives represented 3 percent of respondents.

A Note About DevOps Teams

We were surprised that 16 percent of our survey respondents (1,485 people) were part of a DevOps department, because named DevOps departments have only come into existence in the past five years. Of these respondents, 55 percent identified as DevOps engineers or systems engineers.

"DevOps engineer" and "systems engineer" titles also showed up in IT operations, development and engineering departments.

Incidence of DevOps & System Engineers by Department

	IT Operations Department	Development or Engineering Department
DevOps Engineer	129 / 4.6%	144 / 5.4%
Systems Engineer	503 / 17.8%	240 / 9.0%

We wanted to see whether there are significant differences between people who work in named DevOps departments and those who don't. Though the two groups are quite similar in many respects, we found a few interesting differences. For example, the majority of respondents in DevOps departments work in companies with 20 to 499 employees. Respondents who work in DevOps departments were more likely to work in the entertainment, technology and web software industries than respondents who worked in other departments. They are also slightly more prevalent in California and Texas than in the rest of the United States, though we found no difference in distribution across global regions.

Departments

IT Ops	30.4%
Dev/Eng	28.8%
DevOps	16%
Consultant	5.6%
C-level Executive	2.3%
Network Operations	1.9%
Information Security	1.4%
Quality Assurance	1.3%
Release Engineering	1.2%
All Others	11.1%

DevOps Roles

Architect	10.3%
Automation or Tooling Engineer	10.5%
DevOps Engineer	31.3%
Release or Build Engineer	5.2%
Systems Engineer	23.4%
Manager or Senior Manager	8.3%
Director	4.8%
VP	0.9%
Other	5.2%

The most striking difference, not surprisingly, is the use of DevOps practices. Ninety-two percent of respondents working in DevOps departments said they were currently implementing DevOps practices, or already had. This is sharply higher than the overall group, of which 57 percent say they are implementing, or have already implemented, DevOps practices. What we *don't* know is whether DevOps departments are doing anything that's noticeably different from IT departments that employ DevOps practices.

There's plenty of lively discussion about the pros and cons of creating a DevOps department. Our analysis makes it clear that DevOps teams are a growing trend. We also have evidence that organizations with a DevOps department are successful: More than 90 percent of those working in DevOps departments are in companies with high to medium IT performance. The DevOps-department cohort is 50 percent more likely to be in a company with high IT performance.

Despite a growing trend of DevOps departments, we think a dedicated team can miss the point.

Jez Humble's take:

"The DevOps movement addresses the dysfunction that results from organizations composed of functional silos. Thus, creating another functional silo that sits between dev and ops is clearly a poor (and ironic) way to try and solve these problems. Devops proposes instead strategies to create better collaboration between functional silos, or doing away with the functional silos altogether and creating cross-functional teams (or some combination of these approaches)."

Read the rest of his blog post at:

http://continuousdelivery. com/2012/10/theres-no-such- thing-as-a-devops-team/

This Year's Results Support and Deepen Last Year's Findings

Our goal was to reach a sample population that faithfully represents today's technical professional landscape. While this isn't a truly random sampling, we are confident that we achieved our goal, because our sampling was both so large, and so diverse across a number of demographic measures, including: departments, job titles, organization size, infrastructure size and geographic regions.

Why is diversity of respondents so important? It allows us to be confident that our analysis applies to the population as a whole, and that our conclusions aren't limited to certain types of organizations or narrow bands of job roles. The trends we have identified affect *everyone* — not just big WebOps shops, not just startups in cutting-edge tech regions, not just organizations with less than 500 servers, not just organizations with or without a separate DevOps team.

We've confirmed last year's performance findings: high-performing organizations are still **deploying code 30 times more frequently**, with **50 percent fewer failures** than their lower-performing counterparts.

This year's findings, based on more detailed questions, allow us to determine with greater specificity what high-performing organizations look like, and what practitioners and managers can do to achieve high IT performance in their own organizations. **We can now assert with confidence that high IT performance correlates with strong business performance, helping to boost productivity, profitability and market share.**

ORGANIZATIONAL
PERFORMANCE

IT is typically considered a cost center, and until now, there has been little evidence that investing in IT provides significant returns. We wanted to test the hypothesis that IT performance actually does make a difference to organizational performance. We found that companies with high IT performance are twice as likely to exceed their profitability, market share and productivity goals, giving them a strong competitive edge.

Many factors drive organizational performance, including market conditions, executive leadership and operational effectiveness. There is no simple formula or single factor that guarantees success. However, we now have quantitative evidence that IT performance and DevOps practices contribute to organizational performance.

We found that companies with high IT performance are twice as likely to exceed their profitability, market share and productivity goals. Investing in IT initiatives can deliver real returns, and give businesses a competitive edge.

Three major factors that contribute to organizational performance:

▴ **IT performance and DevOps Practices.** Our data shows that IT performance and well-know DevOps practices, such such as those that enable continuous delivery, are predictive of organizational performance. As IT performance increases, profitability, market share and productivity also increase. IT is a competitive advantage, not just a utility, and it's more critical now than ever to invest in IT performance. In the following sections, we'll discuss which practices deliver business value and drive overall organizational performance.

▴ **Organizational culture and climate for learning.** DevOps has always been about culture, not just about tools and processes. We found that the cultural practices and norms that characterize high-trust organizations — good information flow, cross-functional collaboration, shared responsibilities, learning from failures and encouragement of new ideas — are the same as those at the heart of DevOps. That helps explain why DevOps practices correlate so strongly with high organizational performance.

▴ **Job satisfaction.** A breakthrough finding was that job satisfaction is the No. 1 predictor of organizational performance. Job satisfaction also highly correlates with DevOps practices and culture. Just as some suggest that happy cows make better cheese, DevOps practices increase employee satisfaction, which leads to better business outcomes

In the following sections, we'll dive more deeply into these three factors.

How We Measured Organizational Performance

To measure organizational performance, survey respondents were asked to rate their organization's relative performance across several dimensions, a scale validated multiple times in prior research. For more information on this measure, read S.K. Widener's paper, "An empirical analysis of the levers of control framework."

IT
PERFORMANCE

Last year, we were delighted to discover that we could actually quantitatively define IT performance. We discovered that high-performing IT organizations are more agile and reliable: They deploy code 30 times more frequently than their lower-performing peers, with 50 percent fewer failures. This year, we've learned that IT performance has real impact on the business: Companies with high IT performance are twice as likely to exceed their profitability, market share and productivity goals.

Our analysis showed that the longer an IT organization continues to practice DevOps, the more its IT performance improves. Because business outcomes are linked to IT practices, greater DevOps maturity gives companies a clear lead, as their business outcomes continue to improve over time along with IT performance.

IT performance is measured in terms of throughput and stability, two attributes that seem to be opposed, yet are both essential to achieving IT that's a real strategic asset. The individual measures that make up IT performance are **deployment frequency**, **lead time for changes**, and **mean time to recover from failure**. Throughput is measured by deployment frequency and lead time for changes, while stability is measured by mean time to recover. To increase IT performance, you need to invest in practices that increase these throughput and stability measures.

Top Practices Correlated with Deployment Frequency

Continuous delivery
Continuous delivery ensures that your software is always in a releasable state, turning deployment into a non-event that can be performed on demand.

Use of version control for *all* production artifacts
When it's easy to recreate environments for testing and troubleshooting, throughput goes up.

Top Practices Correlated with Lead Time for Changes

Use of version control for *all* production artifacts
The ability to get changes into production repeatedly in a reliable, low-risk way depends on the comprehensive use of version control.

Automated testing
With a reliable and comprehensive set of automated tests, you can quickly gain confidence that your code is releasable without lengthy integration and manual regression testing cycles.

Top Practices Correlated with Mean Time to Recover (MTTR)

Use of version control for *all* production artifacts
When an error is identified in production, you can quickly either redeploy the last good state or fix the problem and roll forward, reducing the time to recover.

Monitoring system and application health
Logging and monitoring systems make it easy to detect failures and identify the events that contributed to them. Proactive monitoring of system health based on threshold and rate-of-change warnings enables us to preemptively detect and mitigate problems.

How We Measured IT Performance

Coming up with a quantitative definition of IT performance wasn't easy. After all, how do you measure concepts that can't be measured directly, such as happiness or job satisfaction? In statistics, you do this with what's known as a latent construct.

To get a highly reliable and valid latent construct for IT performance, we started with a set of related independent variables: deployment frequency, lead time for changes, mean time to recover and change fail rate.

After a lot of refining and statistical testing, we found that change fail rate was not significantly correlated with the other variables leading us to our current definition of IT performance: **deployment frequency**, **lead time for changes** and **mean time to recover.**

After several additional statistical tests to verify, we can now confidently say that we have a useful and quantifiable definition of IT performance in the context of DevOps.

IT departments also look at change fail rate as a measure of their performance. We did not find that change fail rate correlated with any specific practices. However, we did see significant differences between groups with high, medium and low change fail rates. High performing IT organizations have 50% lower change fail rates than medium and low performing IT organizations.

Practices Correlated with IT Performance Metrics

Throughput Metrics	Stability Metrics

Deployment Frequency
◢ Continuous Delivery
◢ Version Control

Lead Time for Changes
◢ Version Control
◢ Automated Testing

Mean Time to Recover
◢ Version Control
◢ Monitoring System
  and Application Health

Change Fail Rate

Not strongly correlated with specific practices.

Deploy Frequency & DevOps Maturity

DevOps maturity was highly correlated with deployment frequency. Deployment is often the biggest pain point leading to the implementation of DevOps practices. The longer dev and ops teams practice DevOps, the better they get, leading to higher deployment frequency.

In this section, we'll explore the practices that correlate most strongly to high IT performance.

Continuous Integration and Continuous Delivery

Continuous delivery is a methodology that focuses on making sure your software is always in a releasable state throughout its lifecycle. Continuous delivery changes the economics of the software delivery process, making it cheap and low-risk to release new versions of your software to the people who use it.

Implementing continuous delivery means creating multiple feedback loops to ensure that high-quality software gets delivered to users more quickly. Continuous delivery requires that developers, testers, and UX, product and operations people collaborate effectively throughout the delivery process.

Continuous integration is a development practice whereby developers routinely merge their code into trunk (also known as master) in a version control system — ideally, multiple times per day. Each change triggers a set of quick tests to discover serious regressions, which developers must fix immediately. This process is actually the first step towards achieving continuous delivery — quite literally, because the CI process creates canonical builds and packages that are ultimately deployed and released.

Our analysis of the survey results showed that developers breaking large features into small incremental changes and merging their code daily into trunk are strongly correlated with both IT and organizational performance.

These practices, which are a critical part of continuous integration and continuous delivery, also require automated testing and version control — two other practices correlated with IT performance. Automated and manual validations such as performance and usability testing give the team the chance to detect problems introduced by changes as soon as possible, and to fix them immediately.

The goal of continuous integration — and of continuous delivery, in fact — is to make the process of releasing changes to software users a technically straightforward, even boring, process. When that's the case, the IT team can spend more of its time on the kind of proactive, strategic planning that can contribute even more value to the enterprise.
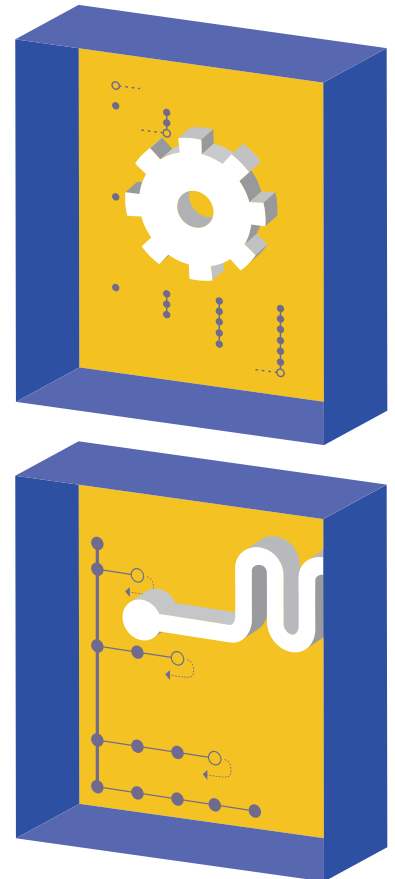
Automated Acceptance Testing

As part of continuous delivery, automated acceptance tests are written alongside new code to ensure that new features meet business requirements, and existing high-value features are protected against regressions. New versions of the system must pass numerous automated tests before they can undergo exploratory testing and be deployed to production, substantially reducing the reliance on time-consuming and error-prone manual regression testing, and improving overall quality and stability of the software.

The key automated testing practices that have an impact on IT performance are:

◢ **Developers get feedback from acceptance and performance tests every day.** Rapid feedback enables developers to quickly fix bugs discovered through acceptance test failures. Because the feedback cycle encourages learning, development teams continue to get better and faster over time.

◢ **It's easy for developers to fix acceptance test failures.**
To troubleshoot and fix acceptance test failures, developers must be able to easily reproduce and debug failures on their development machines. This requires automated provisioning, configuration and management of development environments, as well as software that is architected with test automation in mind. For example, a common approach is to stub out integration points, so expensive integrated environments aren't required to gain confidence that the software is releasable.

Automated Tests and Organizational Performance

In a continuous delivery scenario, every new feature is tested against business requirements, supporting our findings that automated testing is highly correlated with organizational performance.
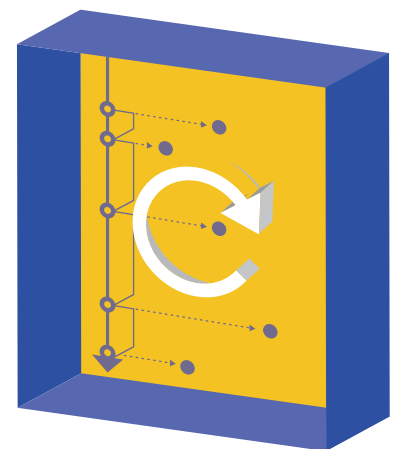
Version Control

A prerequisite for continuous integration and continuous delivery is that everything required to reproduce the production environment must be checked into version control, including:

◢ **Application code**
◢ **Application configurations**
◢ **System configurations**
◢ **Tests and deployment scripts that are used to validate software and deploy it to test, staging and production environments**

Comprehensive version control enables us to cheaply create testing environments on demand and communicate the desired state of our systems effectively across teams, as well as providing the foundation of a reliable, repeatable, low-risk process for performing deployments and restoring service in the event of a failure.

Top 5 Predictors of IT Performance

▲ **Peer-reviewed change approval process.** We found that when external approval (e.g., change approval boards) was required in order to deploy to production, IT performance decreased. But when the technical team held itself accountable for the quality of its code through peer review, performance increased. Surprisingly, the use of external change approval processes had no impact on restore times, and had only a negligible effect on reducing failed changes. In other words, external change approval boards had a big negative impact on throughput, with negligible impact on stability.

▲ **Version control for *all* production artifacts.** Version control provides a single source of truth for all changes. That means when a change fails, it's easy to pinpoint the cause of failure and roll back to the last good state, reducing the time to recover. Version control also promotes greater collaboration between teams. The benefits of version control shouldn't be limited to application code; in fact, our analysis shows that organizations using version control for both system and application configurations have higher IT performance.

▲ **Proactive monitoring.** Teams that practice proactive monitoring are able to diagnose and solve problems faster, and have a high degree of accountability. When failures are primarily reported by an external source, such as the network operations center (NOC) — or worse, by customers — IT performance suffers.

▲ **High-trust organizational culture.** One of the pillars of DevOps is culture, and we were pleased to prove what we already knew anecdotally: Culture matters. In fact, organizational culture was highly predictive of both IT performance and overall organizational performance. No one should be surprised to hear that high-trust cultures lead to greater performance, while bureaucratic and fear-based cultures are destructive to performance.

▲ **Win-win relationship between dev and ops.** It's not dev versus ops, it's dev plus ops. When the outcome of a dev and ops interaction is win-win, IT performance wins.

The Virtuous Circle of IT Performance & Organizational Performance

As we analyzed our survey data, we discovered something interesting: While a high-performing IT team improves the entire organization's performance as a business, organizational performance itself plays a role in improving IT stability.

A high-performing company is not going to make sudden, unreasonable demands on its IT team, but will work closely with IT to plan the initiatives that IT must support, or that will affect IT. There's a virtuous circle at work here: As stability improves, IT performance improves. This improved performance helps to create a better-functioning business that can pay attention to the communications and processes that enhance and improve stability. We're delighted with this finding, which underlines the importance of continuous learning and improvement.

THE CULTURE
OF HIGH
PERFORMANCE

Organizational culture is one of those things we feel is important, but it is difficult to quantify in terms of impact on business performance. Yet in our survey, we were able to measure culture quantitatively, and found it is one of the top predictors of organizational performance. Culture really does matter to the business.

A number of researchers have found that organizational culture has an impact on business performance. Our analysis of the survey data reinforces this: We found that culture is one of the top predictors of organizational performance.

To measure culture, we used a typology developed by Ron Westrum, a sociologist who found that organizational culture was predictive of safety in the health care industry. Table 1 shows Westrum's typology of organizational cultures. According to Westrum, hallmarks of a generative organization are good information flow, high cooperation and trust, bridging between teams, and conscious inquiry.

Table 1 Typology of Organizational Culture (Westrum, 2004)

Pathological *Power-oriented*	Bureaucratic *Rule-oriented*	Generative *Performance-oriented*
Low cooperation	Modest cooperation	High cooperation
Messengers shot	Messengers neglected	Messengers trained
Responsibilities shirked	Narrow responsibilities	Risks are shared
Bridging discouraged	Bridging tolerated	Bridging encouraged
Failure leads to scapegoating	Failure leads to justice	Failure leads to inquiry
Novelty crushed	Novelty leads to problems	Novelty implemented

You can easily map these attributes to DevOps practices, which reinforce the importance of collaboration between functions such as development, quality assurance, operations and information security; no-blame postmortems; a high-trust culture in which people are free to experiment; and a strong focus on continuous improvement.

Westrum on Organizational Culture

"The climate that provides good information flow is likely to support and encourage other kinds of cooperative and mission-enhancing behaviour, such as problem solving, innovations, and inter-departmental bridging. When things go wrong, pathological climates encourage finding a scapegoat, bureaucratic organizations seek justice, and the generative organization tries to discover the basic problems with the system."

The Culture of High Performance

Deployment scenarios by organization type

Pathological Organizations *Power-oriented*

In pathological organizations, developers spend months writing code for a release. There is little collaboration with test and IT operations during the development process to ensure the code will actually work in production. The release is integrated, thrown over the wall to test, and finally released.

Each of these stages is painful and involves a great deal of rework. The system often fails to meet user needs. Management finally decides to fire someone in IT operations, making everyone less likely to speak up about dysfunctions, or make recommendations. In pathological organizations, operations departments are usually overwhelmed with unplanned work, and incidents lead to finger-pointing exercises.

 IT is always on the critical path for new work, which is usually delivered late. The products and services IT is able to build under these conditions are likely to be of lower quality than the business needs or expects. Employees are disenfranchised and unmotivated. Such organizations will — sooner or later — be outpaced by higher-performing rivals.

Bureaucratic Organizations *Rule-oriented*

In bureaucratic organizations, there is enough discipline to ensure that IT can produce relatively predictable results. However, in this environment high throughput is usually seen as a threat to stability, and so it's difficult to change processes or break down organizational silos.

The reliance on process and centralized decision-making to manage complexity makes it hard to react quickly to the changing needs of customers; employees feel disempowered and become frustrated. Such organizations lose their ability to innovate, and find it hard to compete with higher-performing organizations.

Top Predictors of Organizational Culture

- Job satisfaction
- Climate for learning
- Win-win relationship between dev and ops
- Version control
- Automated testing

Puppet Labs ©2014   22   DEVOPS REPORT

Generative Organizations *Performance-oriented*

In a generative environment, it is understood that continuous improvement leads to ever-higher levels of throughput and stability. Development, test and operations work together throughout the delivery process, and collaborate on ways to reduce the cost and risk of making changes. Everybody is encouraged to run experiments to learn how to improve both processes and the products and services they build. Failure is treated as a learning opportunity. The flow of information and feedback is fast, because it's built into the system, from continuous integration to automated tests to monitoring of production environments. The result is a business that can pull ahead of the competition because it can quickly detect and respond to new market opportunities, unburdened by an external decision-making process. Just as important, generative workplaces have more engaged employees who can express their intelligence and creativity, lending the company an important competitive advantage.

JOB
SATISFACTION

While we suspected job satisfaction affects organizational performance, we were surprised to find that job satisfaction was the No. 1 predictor of organizational performance.

How Does Job Satisfaction Impact Organizational Performance?

We mentioned the virtuous circle earlier in reference to IT performance, and we see it at work here, too: People who feel supported by their employers, who have the tools and resources to do their work, and who feel their judgment is valued, turn out better work. Better work results in higher IT performance, which results in a higher level of organizational performance.

This cycle of continuous improvement and learning is what sets successful companies apart, enabling them to innovate, get ahead of the competition — and win.

How Does DevOps Contribute to Job Satisfaction?

Although DevOps is first and foremost about culture, it's important to note that job satisfaction depends strongly on having the right tools and resources to do your work. Tools are an important component of DevOps practices, and many of these tools provide automation.

Automation matters because it gives over to computers the things computers are good at: rote tasks that require no thinking, that in fact are done better when you don't think too much about them. Turning these tasks over to computers allows people to focus on the things they're good at: weighing the evidence, thinking through problems, making decisions. The ability to apply one's judgment and experience to challenging problems is a big part of what makes people satisfied with their work.

Looking at the measures that correlate strongly with job satisfaction, we see some commonalities. Proactive monitoring, version control and automated testing all automate menial tasks, and require people to make decisions based on a feedback loop. Instead of managing tasks, people get to make decisions, employing their skills, experience and judgment.

Top predictors of job satisfaction

- High-trust organizational culture
- Climate of learning
- Win-win relationships between ops, dev and infosec teams
- Proactive monitoring and autoscaling
- Use of version control for all production artifacts
- Automated testing

RECOMMENDATIONS
FOR
IMPROVEMENT

Organizational improvement isn't simple or easy, and it can be difficult to measure whether individual actions are having an impact. We've identified some concrete actions individuals can take to positively change their team. To be clear, there's no secret recipe (or magic DevOps wand) that will fix your organization. Our goal is to uncover the underlying structures behind your daily pain points and problems.

If you're trying to institute change, don't forget you must make time and resources available for improvement work. Creating change takes time, and people also need time to adjust to the changes, as you build practices such as automation and continuous integration into your delivery process. On top of that, improving process is itself a skill that needs to be learned; teams that routinely work on improvement get better at it over time.

We've split our recommendations between practitioners and managers, because they have different methods and opportunities available. There is some overlap, especially when it comes to encouraging new ideas or working with other teams.

	Practitioners	Managers
Cross-Functional Collaboration	◢ **Work with other teams, and find ways to build empathy.** Building bridges between teams will increase your understanding of the challenges at every point in the lifecycle. As a developer, try to put yourself in the shoes of the operations team: How will they monitor and deploy your software? As an ops person, think about how to help devs get feedback on whether their software will work in production. ◢ **Make invisible work visible.** Record what you and your colleagues do to support cross-functional collaboration. If members of the dev and ops teams work together to solve a problem in the development environment, make sure to record and recognize what made that possible: an ops colleague taking an extra on-call shift, or an assistant ordering food for a working session. These are non-trivial contributions, and may be required for successful collaboration.	◢ **Build trust with your counterparts on other teams.** Building trust between teams is the most important thing you can do, and it must be built over time. Trust is built on kept promises, open communication, and behaving predictably even in stressful situations. Your teams will be able to work more effectively, and the relationship will signal to the organization that cross-functional collaboration is valued. ◢ **Encourage practitioners to move between departments.** An admin or engineer may find, as they build their skills, that they're interested in a role in a different department. This sort of lateral move can be incredibly valuable to both teams. Practitioners bring valuable information about processes and challenges to their new team, and members of the previous team have a natural point person when reaching out to collaborate. ◢ **Actively seek, encourage and reward work that facilitates collaboration.** Make sure success is reproducible and pay attention to latent factors that make collaboration easier.

	Practitioners	Managers
Climate of Learning	▴ **Learn by sharing knowledge.** Hone your skills by teaching them to someone else — by pairing on difficult problems, submitting talks at conferences, writing blog posts, or just reaching out when you see someone struggling with a task you've mastered. ▴ **Always bring back what you learned.** Give a presentation, share your notes, or offer tutorials on new skills. When you've been given the opportunity to get training, don't come back empty-handed. ▴ **Prepare for postmortems.** An important part of a learning climate is effective, blameless postmortems. This type of post-event analysis identifies the actions you and your team can take to improve, and to incrementally learn from failures. Prioritize root cause analysis after an outage, and make sure to provide a detailed log of actions taken and effects observed, without fear of punishment or retribution. Learn how to participate in the postmortem without taking it personally, and don't level personal criticism at anyone. Remember, postmortems make your service better.	▴ **Create a training budget, and advocate for it internally.** Emphasize how much the organization values a climate of learning by putting resources behind formal education opportunities. ▴ **Create a climate of learning.** Learning often happens outside of formal education. Ensure that your team has the resources to engage in informal learning, and the space to explore ideas. Some companies, like 3M and Google, have famously set aside a portion of time (15 percent and 20 percent, respectively) for focused free-thinking and exploration of side projects. ▴ **Make it safe to fail.** If failure is punished, people won't try new things. Treating failures as opportunities to learn, and holding blameless post-mortems to work out how to improve processes and systems, help people feel comfortable taking (reasonable) risks, and help create a culture of innovation. ▴ **Create opportunities and spaces to share information.** Whether you create weekly lightning talks or offer resources for monthly lunch-and-learns, set up a regular cadence of opportunities for employees to share their knowledge.

	Practitioners	Managers
Tools	◤ **Evolve your skill set to solve your most important problems.** Focus on learning new skills that help you overcome the big challenges. If you don't know how to program, make it a priority to learn. If you already have some skills, learn a new language, framework or library. Spend your time writing code to solve problems, instead of just gluing together vendor solutions with fragile scripts. Make sure you understand the theory behind the software you're using before you implement it. ◤ **Learn foundational concepts.** Refresh or learn some of your high school mathematics, such as basic calculus and probability. A quick statistics lesson can drastically improve your understanding of monitoring output and performance analysis. ◤ **Automate the things that are painful.** The IT predictors almost all rely on automation: version control, automated testing, monitoring, and more. One of the best ways to see results from your team is to simplify and automate high value and repetitive tasks.	◤ **Make sure your team can choose their tools.** Unless there's a good reason, practitioners should choose their own tools. If they can build infrastructure and applications the way they want, they're much more likely to be invested in their work. This is backed up in the data: One of the major contributors to job satisfaction is whether employees feel they have the tools and resources to do their job. If your organization must standardize tools, ensure that procurement and finance are acting in the interests of teams, *not* the other way around. ◤ **Make monitoring a priority.** Refine your infrastructure and application monitoring system, and make sure you're collecting information on the right services, and putting that information to good use. The visibility and transparency yielded by effective monitoring are invaluable. Proactive monitoring was strongly related to performance and job satisfaction in our survey, and it is a key part of a strong technical foundation.

REPORT
CONTRIBUTORS

Nicole Forsgren Velasquez

Nicole Forsgren Velasquez is a professor at the Huntsman School of Business at Utah State University, with expertise in IT impacts, knowledge management and user experience, and a background in enterprise storage and system administration.
Nicole holds a Ph.D. in management information systems and a masters in accounting from the University of Arizona.

Gene Kim

Gene Kim is a multiple award-winning CTO, researcher and author. He founded Tripwire and was its CTO for 13 years, and has researched high-performing IT organizations for almost as long. He has written three books, including *The Visible Ops Handbook* and *The Phoenix Project: A Novel About IT, DevOps, and Helping Your Business Win.*

Nigel Kersten

Nigel is the CIO at Puppet Labs, responsible for technical operations and business optimization.He came to Puppet Labs from Google in Mountain View, California, where he was responsible for the design and implementation of one of the largest Puppet deployments in the world. He's been a sysadmin for Linux and Mac deployments since before Mac OS became a UNIX.

Jez Humble

Jez Humble is a principal at ThoughtWorks Studios, a lecturer at University of California at Berkeley, and co-author of the Jolt Award-winning *Continuous Delivery*, published in Martin Fowler's Signature Series (Addison Wesley, 2010), and *Lean Enterprise*, in Eric Ries' Lean series (O'Reilly, 2014). He has worked with a variety of platforms and technologies, consulting for nonprofits, telecoms, financial services, and online retail companies.

Special Thanks

We'd like to thank everyone who contributed to the making of the 2014 State of DevOps Report. First and foremost, our thanks goes to the 9,200+ people who took the survey and provided us with this treasure trove of data. Next, the good people of the DevOps Survey of Practice Google Group who helped shape the survey questions. Last but not least, we want to thank all of the people involved in the analysis and development of the report: Alanna Brown, Michelle Carroll, Sze Wa Cheung, Aliza Earnshaw, Peter Irby, Tiffany Longworth, Pamela Ju and Wally Zabaglio.

WORKS CITED

Westrum, R. A typology of organisational cultures. *Qual. Saf. Health Care* 13, (2004), 22-27.
Widener, S.K. An empirical analysis of the levers of control framework. *Accounting, Organizations and Society* 32:7-8, (2007), 757–788.