puppet labs®

2015 STATE OF DevOps REPORT

In Partnership with

IT REVOLUTION

Sponsored by pwc

Table of Contents

Chapter 1
Executive Summary

Chapter 2
Who Took the Survey

Chapter 3
IT Performance

Chapter 4
The Impact of Lean Management & Continuous Delivery on Culture & Performance

Chapter 5
Application Architecture & Developer Productivity

Chapter 6
How IT Managers Can Help Their Teams Win

Chapter 7
Burnout

Chapter 8
Methodology

1. Executive Summary

The fourth annual State of DevOps Survey confirms that IT performance provides real business value. High-performing IT organizations have a strong and positive impact on the overall performance of the organizations they serve. This year's report also tells us:

- High-performing IT organizations deploy 30x more frequently with 200x shorter lead times; they have 60x fewer failures and recover 168x faster.
- Lean management and continuous delivery practices create the conditions for delivering value faster, sustainably.
- High performance is achievable whether your apps are greenfield, brownfield or legacy.
- IT managers play a critical role in any DevOps transformation.
- Diversity matters.
- Deployment pain can tell you a lot about your IT performance.
- Burnout can be prevented, and DevOps can help.

This is our fourth annual State of DevOps Report. As in prior years, we have deepened our understanding of how DevOps enables IT performance and organizational performance, based on responses from more than 20,000 technical professionals we've surveyed over the past four years.

Last year we linked IT performance to organizational performance, proving that IT is not just a cost center — it provides real business value. We also confirmed that DevOps practices lead to better IT and organizational performance. This year we were surprised to find that while throughput for the high-performing group was no different than for last year's high performers, stability was significantly better. This supports the widely held belief that DevOps practices equip organizations to embrace more and more change, rather than to fear it. We also found that lean management and continuous delivery practices contribute to both throughput and stability. That in turn improves organizational performance.

So how do you actually achieve higher performance? This year's research shows that IT managers carry a great deal of responsibility for getting there, especially those in the middle layer who are responsible for connecting on-the-ground execution to the strategic objectives of the business. This year's report provides guidance to IT managers for improving the performance of their teams and leading their organizations through a DevOps transformation.

Key findings include:

- **High-performing IT organizations experience 60 times fewer failures and recover from failure 168 times faster than their lower-performing peers. They also deploy 30 times more frequently with 200 times shorter lead times.** Failures are unavoidable, but how quickly you detect and recover from failure can mean the difference between leading the market and struggling to catch up with the competition.

- **Lean management and continuous delivery practices create the conditions for delivering value faster, sustainably.** Manufacturing was revolutionized by the application of lean principles in the 1980s. Today, it's IT's turn to go lean. When you apply lean management and continuous delivery practices to software delivery, you get the same results — higher quality, shorter cycle times with quicker feedback loops, and lower costs. And the benefits don't stop there: These practices also contribute to creating a culture of learning and continuous improvement, lower levels of burnout, and higher organizational performance overall.

High performance is achievable if you architect with testability & deployability in mind.

- **It doesn't matter if your apps are greenfield, brownfield or legacy — as long as they are architected with testability and deployability in mind, high performance is achievable.** We were surprised to find that the type of system — whether it was a system of engagement or a system of record, packaged or custom, legacy or greenfield — is not significant. Continuous delivery can be applied to any system, provided it is architected correctly. We also found that high performers are more likely to use a microservices architecture, and less likely to outsource software development or run their software on mainframes.

- **IT managers play a critical role in any DevOps transformation.** This year's report shows us how IT managers can help their teams win and lead their organizations through a DevOps transformation. Managers play a critical role in connecting the strategic objectives of the business to the work their teams do. Managers can do a lot to improve their team's performance by ensuring work is not wasted and by investing in developing the capabilities of their people.

- **Diversity matters.** Research shows that teams with more women members have higher collective intelligence and achieve better business outcomes. Our survey shows that few teams are truly diverse with regard to gender. We recommend that teams wanting to achieve high performance do their best to recruit and retain more women, and improve diversity in other areas, too.

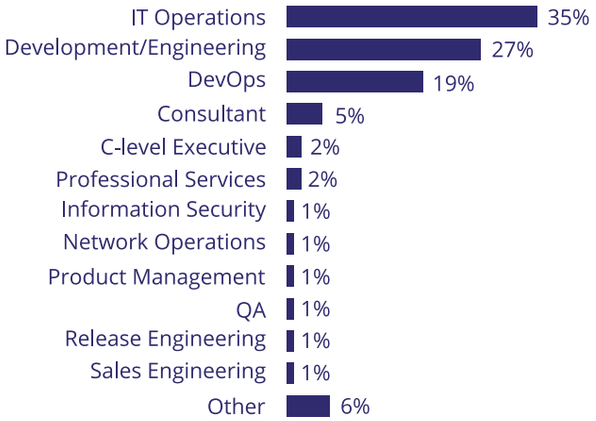
- **Deployment pain can tell you a lot about your IT performance.** Do you want to know how your team is doing? All you have to do is ask one simple question: "How painful are deployments?" We found that where code deployments are most painful, you'll find the poorest IT performance, organizational performance and culture.

- **Burnout can be prevented, and DevOps can help.** Burnout is associated with pathological cultures and unproductive, wasteful work. The consequences of burnout are huge, both for individuals and for organizations. Organizations can fix the conditions that lead to burnout by fostering a supportive work environment and ensuring work is meaningful, and that employees understand how their own work ties to strategic objectives.

2. Who Took the Survey

While we saw a similar distribution of respondents by geography, industry, company size and infrastructure size, compared to last year, we noticed an increase in responses from people working in DevOps departments. This year, nearly one in five respondents came from DevOps departments, compared to fewer than one in six. The proportion of female respondents this year was, on the other hand, lower than we'd hoped and expected. Why aren't organizations working harder to recruit women and people from other underrepresented groups, especially since research has shown that the presence of women in leadership positions is correlated to stronger financial performance and higher levels of group intelligence?

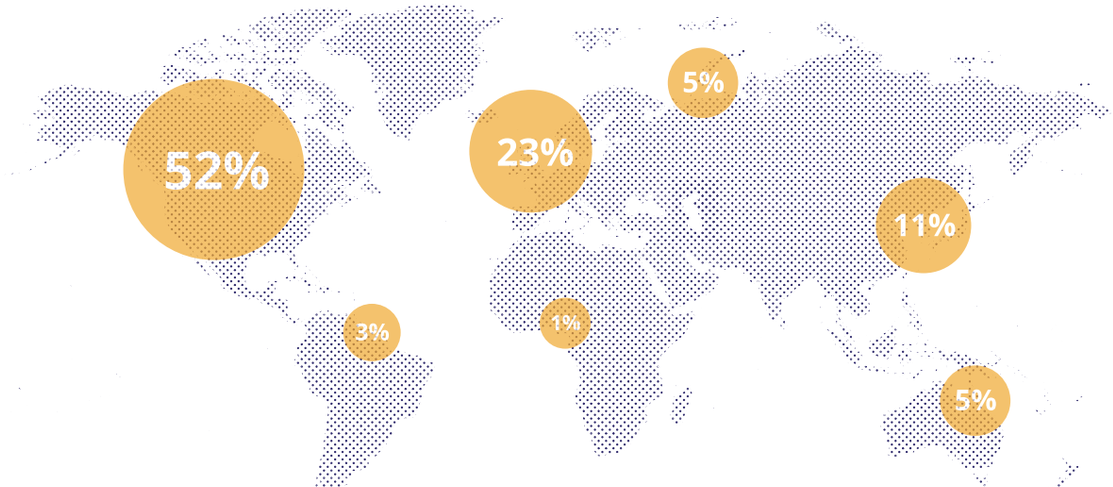
Departments



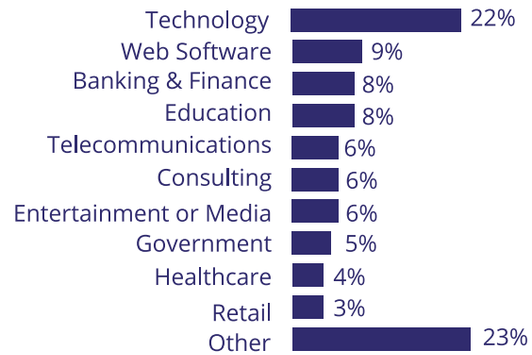
One notable difference this year was an increase in DevOps departments. This year, 19 percent of respondents were part of a DevOps department, up from 16 percent last year.

This year, 4,976 respondents completed the 2015 State of DevOps Survey. Compared to last year, we saw similar distributions across geographies, company size, industries and size of infrastructure.

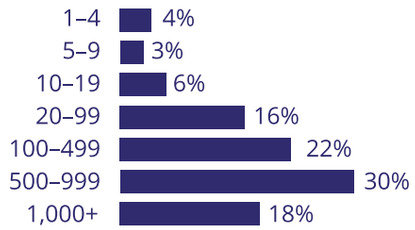
Geography



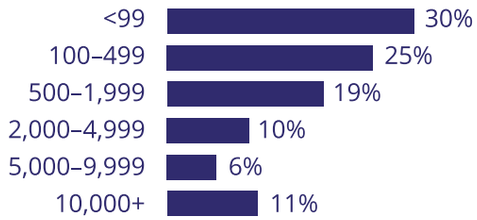
Industries



Size of Infrastructure by Number of Servers



Company Size

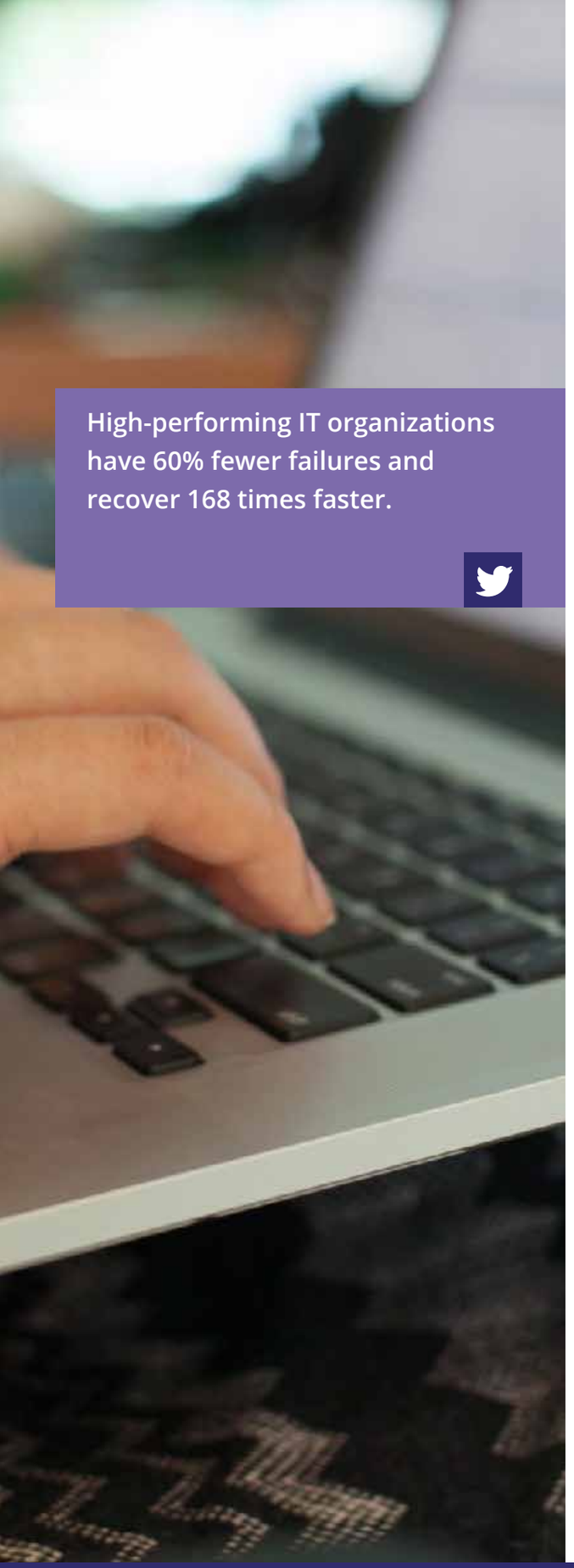


Gender

See page 29 for more about women in tech.

3. IT Performance

We asked about critical throughput measures — deployment frequency and deployment lead time — and discovered that high-performing IT teams deploy code 30 times more frequently than their peers, and 200 times faster (measured in the time required to go from "code committed" to code successfully running in production). We also found that high-performing IT teams achieve far better stability than lower-performing peers, with 60 percent fewer failed deployments and a mean time to recover (MTTR) that's 168 times faster. It's their use of DevOps practices that sets these top performers apart from the pack.



High-performing IT organizations have 60% fewer failures and recover 168 times faster.

One of the most exciting outcomes of our research was coming up with a quantitative definition of IT performance. This breakthrough allowed us to show the relationships between DevOps practices, IT performance and organizational performance.

We have debunked the myth that we need to choose between speed and reliability. We found that high-performing IT organizations deploy code 30 times more frequently and 200 times faster (deployment lead time, defined below) than their lower-performing peers. They also have 60 percent fewer failures and recover 168 times faster. High performers are able to achieve higher levels of both throughput and stability through the use of DevOps practices — a key reason the movement has gained so much traction.

Our definition of IT performance includes two throughput metrics — deployment frequency and deployment lead time — and one stability metric, mean time to recover (MTTR).

Throughput Measures

- **Deployment frequency.** How frequently the organization deploys code.
- **Deployment lead time.** Time required for changes to go from "code committed" to code successfully running in production.

Stability Measures

- **Mean time to recover (MTTR).** Time required to restore service when a service incident occurs (e.g., unplanned outage, service impairment, etc.).

While change fail rate — the percentage of changes that fail when rolled out — is not part of our IT performance construct, we did analyze it because it's such an important measure of IT stability. We found that high-performing IT organizations have the lowest failure rates when they roll out changes, and low-performing IT organizations have the highest change failure rates.

We used the same statistical methods as last year to validate IT performance and divide the population into high, medium and low performers. Surprisingly, for high performers, throughput did not change compared to last year, but stability increased. There are many possible reasons throughput has not increased, and some of these may have nothing to do with effective, efficient deployment practices. Business leaders of the organization may not be able to make decisions about what to deploy any faster than they could before. Growth projections may not justify further investment in faster change at this point. IT doesn't exist in a vacuum, after all; it's there to serve the business.

In some cases, once you've reached a certain level of throughput (including more frequent releases) you're going to get more economic benefit from investing in improved stability. The fact that stability increased in our high-performing group suggests that quality is shifting to the left — that is, it's being built into the software earlier in the development process. If you think of the software delivery process as a manufacturing assembly line, the far left is the developer's laptop where the code originates, and the far right is the production environment where this code eventually ends up. Ensuring that quality is built into each stage of the process implies:

- Better code quality.
- Better testing.
- Building apps with testability and deployability in mind.
- Creating a culture of continuous improvement.

Quality isn't just the responsibility of one team; it's the shared responsibility of everyone involved in the software delivery lifecycle. High-performing organizations know this and build quality into the entire process.

As we discuss in the next section, there's no secret to achieving both speed and reliability, and delivering higher-quality products and services at lower cost. Our research shows this can be achieved with the right practices in place.

Figure 1

Comparison of IT performance metrics between high[1] and low performers

	2015 *(Super High vs. Low)*	2014 *(High vs. Low)*
Deployment Frequency	**30x**	**30x**
Deployment Lead Time	**200x**	**200x**
Mean Time to Recover (MTTR)	**168x**	**48x**
Change Success Rate	**60x**	**3x**

[1] We used the same hierarchical clustering technique as we did last year, but to dig into the larger high-performing group that emerged this year, we did additional analyses and discovered a group of super-high performers.

The graphs below show the distribution of the IT performance answers, for each of the clusters. Each bar represents 100 percent of the cluster members — the 50 percent mark represents the median of each group.

Figure 2

Distribution of deployment frequency by performance cluster

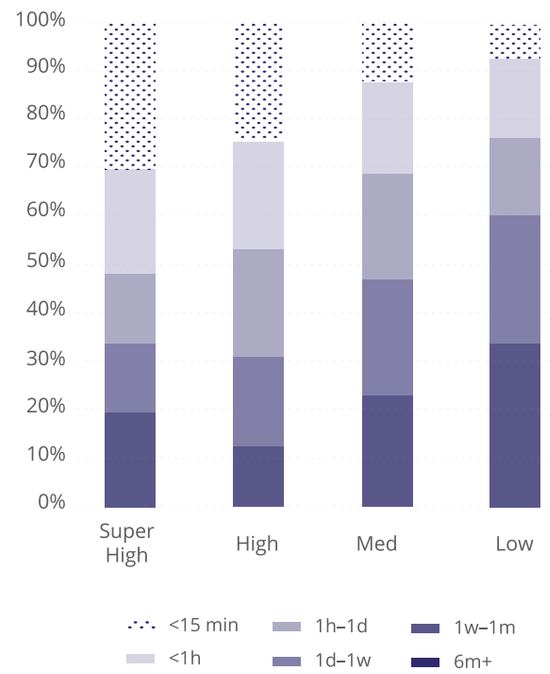


Figure 3

Distribution of deployment lead time by performance cluster

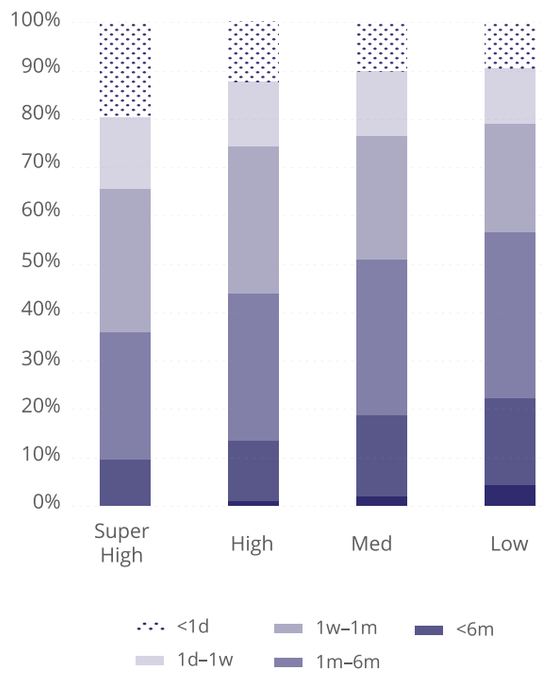
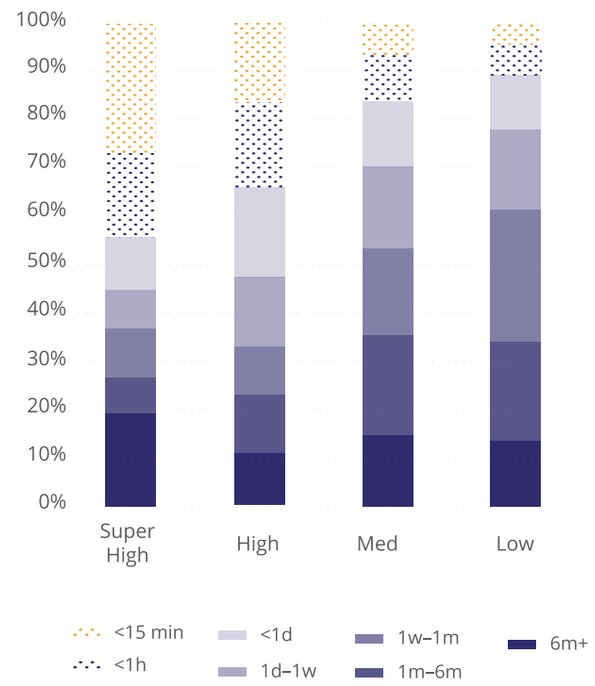


Figure 4

Distribution of mean time to recover (MTTR) by performance cluster



DevOps in Practice

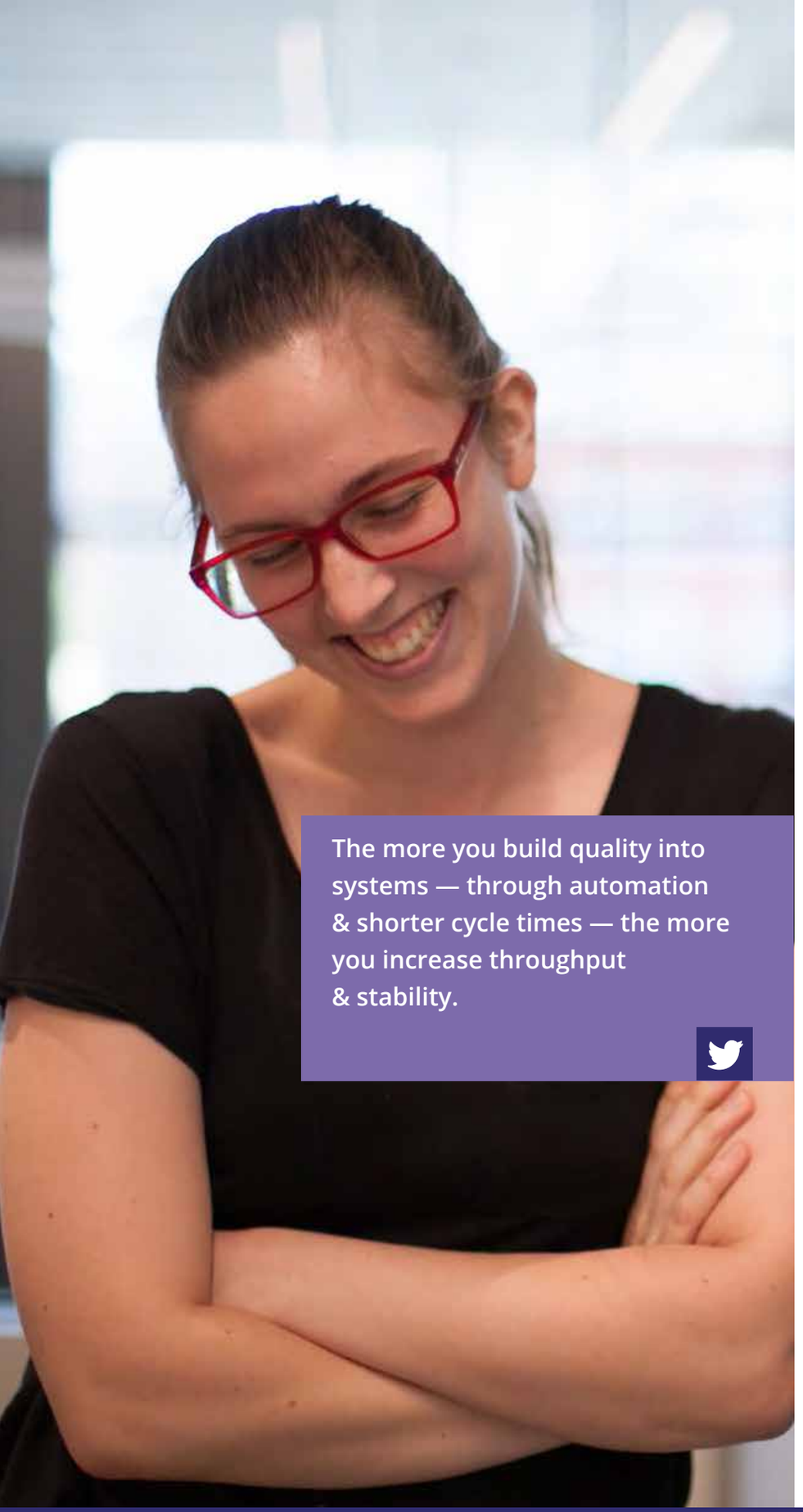
Read the full story >

"The number of issues we had from production emergencies that were triggered by an ops change essentially went to zero. Because we were able to roll changes out in an automated fashion, and then test those changes in the various environments, by the time code got to production, it had been through three other environments — dev, integration, customer test — before it got to production."

— Jez Miller

4. The Impact of Lean Management & Continuous Delivery on Culture & Performance

Manufacturing was revolutionized by the application of lean principles in the 1980s. Today, it's IT's turn to go lean. When you apply lean management practices to technology — limiting work in process (WIP); introducing visual displays to monitor quality, productivity and WIP; and using monitoring data to help inform decisions — you get results. The culture becomes more generative and performance-oriented; people experience less stress in the workplace; and IT performance improves. Our survey analysis shows that both culture and IT performance predict organizational performance, so improvements in these areas lead to better outcomes — including higher financial performance — for the overall organization.



The more you build quality into systems — through automation & shorter cycle times — the more you increase throughput & stability.

In the previous section, we reported that this year's high-performing group had significantly higher stability than last year's high performers, suggesting that fewer things break because issues have been resolved earlier in the software development process. (We talk about this as "shifting left.")

The reason so many people are interested in continuous delivery is because, executed well, it does exactly this: It shifts quality to the left. The set of practices associated with continuous delivery are continuous integration, automated testing, deployment automation, and version control for all production artifacts. These practices work in concert with lean management practices, which include limiting work in process (WIP), use of visual displays, and use of monitoring tools to make business decisions. Used together, continuous delivery and lean management practices amplify each other, and enable ever-improving delivery of better and better software.

This year, we created two new theoretical models[2] for how continuous delivery and lean management practices affect IT performance and organizational performance. As we had hoped and expected, these practices predict IT performance — and IT performance predicts organizational performance. The more you build quality into the system — through automation, reducing batch sizes and shortening cycle times — and the more effectively you manage your team's workload and visualize work queues, defects and bottlenecks, the more you increase throughput and stability.

The diagrams on the next page show how these constructs relate to each other.

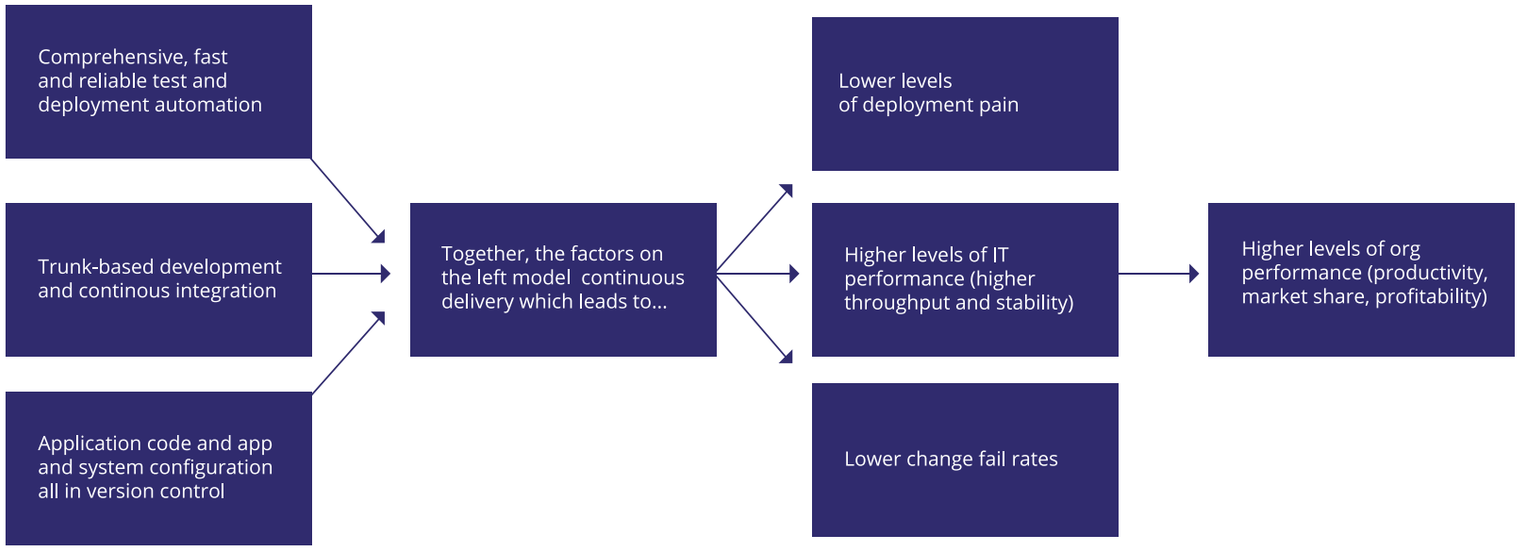
[2] A statistical technique called structural equation modelling (SEM) was used to test the models this year (see Methodology section).

Continuous Delivery Practices

We found that the practices that make up continuous delivery — deployment automation and automated testing, continuous integration, and version control for all production artifacts — have a significant predictive relationship to deployment pain, IT performance and change fail rate. In turn, IT performance predicts organizational performance, as measured by productivity, market share and profitability.[3]

Figure 5

Path diagram showing relationships between continuous delivery practices, IT performance, deployment pain, change failure rate, and organizational performance



The practices that make up continuous delivery have a significant predictive relationship to IT performance.

[3] In a follow-up survey last year, we gathered stock ticker data and 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 people were from 355 companies, and they 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. This year, we hoped to validate this preliminary finding, but our sample size of stock ticker symbols was too small to conduct a meaningful analysis. However, we did find that this year's high performers were 1.5 times more likely than their peers to exceed their organization's' profitability, market share and productivity goals, compared to last year's high performers, which were 1.9 times more likely to exceed goals.

Lean Management Practices

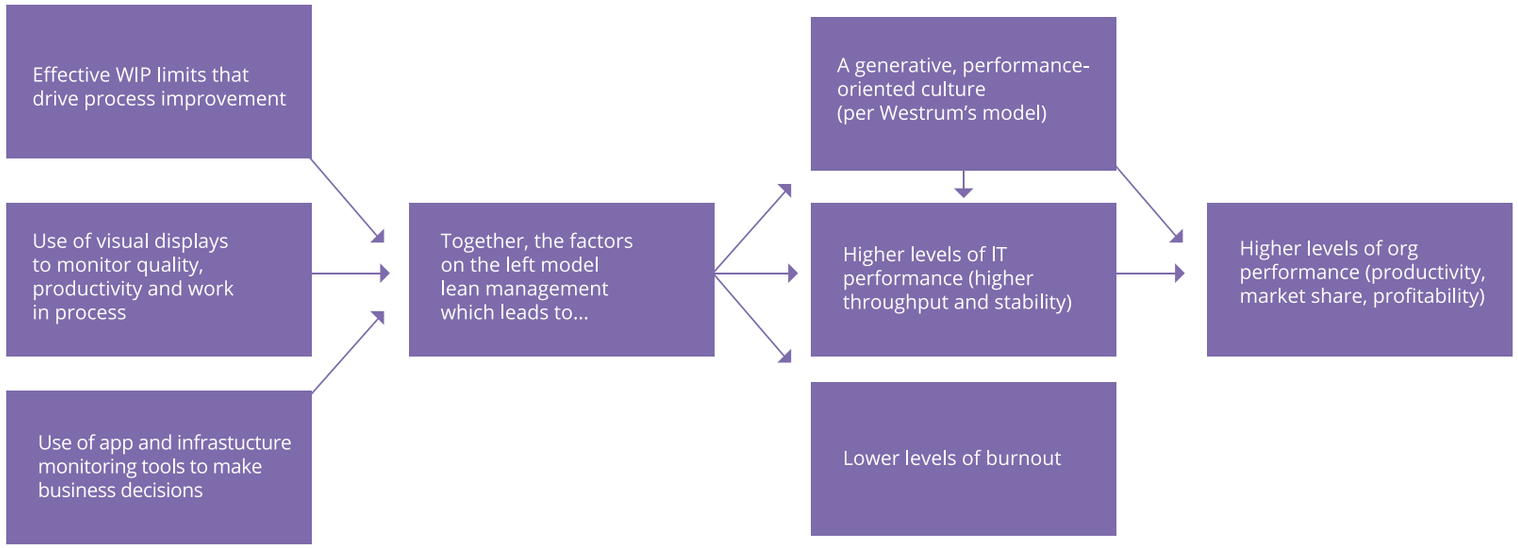
This year's report tackled another question: We wanted to understand the effect of lean management practices on both organizational culture and performance. This year we used two new constructs to measure the impact of lean management:

- The ability of teams to limit work in process (WIP) and use these limits to drive process improvement, which increases throughput.
- The extent to which teams created and maintained visual displays[4] showing key quality and productivity metrics and the current status of work (including defects), and were able to align these metrics with their operational goals.

We included in this year's model a construct from last year: the extent to which teams use data from application performance and infrastructure monitoring tools to make business decisions on a daily basis. So this year, incorporating our new constructs, the model includes limiting WIP, use of visual displays and use of monitoring tools to make business decisions. Survey data supports the model, and shows that all of these practices have an impact on organizational culture, IT performance, and levels of burnout. It's important to note that both organizational culture and IT performance are predictive of organizational performance.

Figure 6

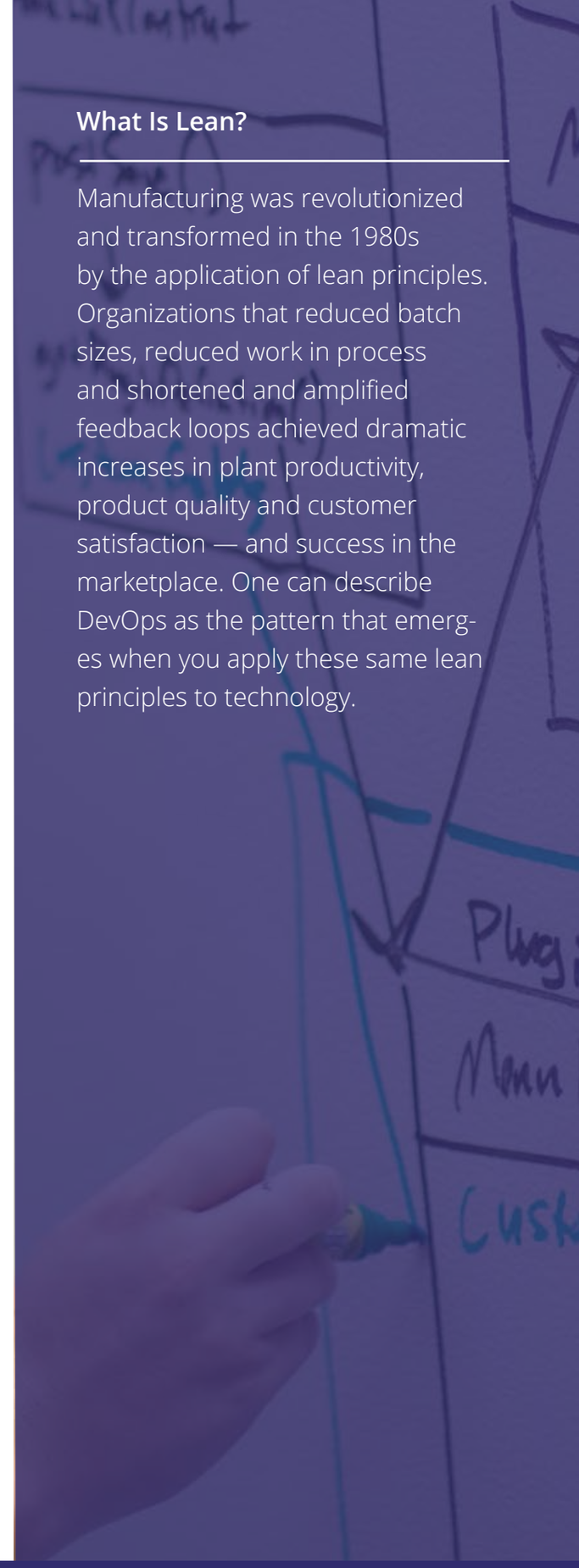
Path diagram showing relationships between lean management practices, IT performance, culture, burnout, and organizational performance



[4] Fullerton, R. R., Kennedy, F. A., and Widener, S. K. "Lean manufacturing and firm performance: The incremental contribution of lean management accounting practices," *Journal of Operations Management* 32:7, (2014), 414–428.

What Is Lean?

Manufacturing was revolutionized and transformed in the 1980s by the application of lean principles. Organizations that reduced batch sizes, reduced work in process and shortened and amplified feedback loops achieved dramatic increases in plant productivity, product quality and customer satisfaction — and success in the marketplace. One can describe DevOps as the pattern that emerges when you apply these same lean principles to technology.

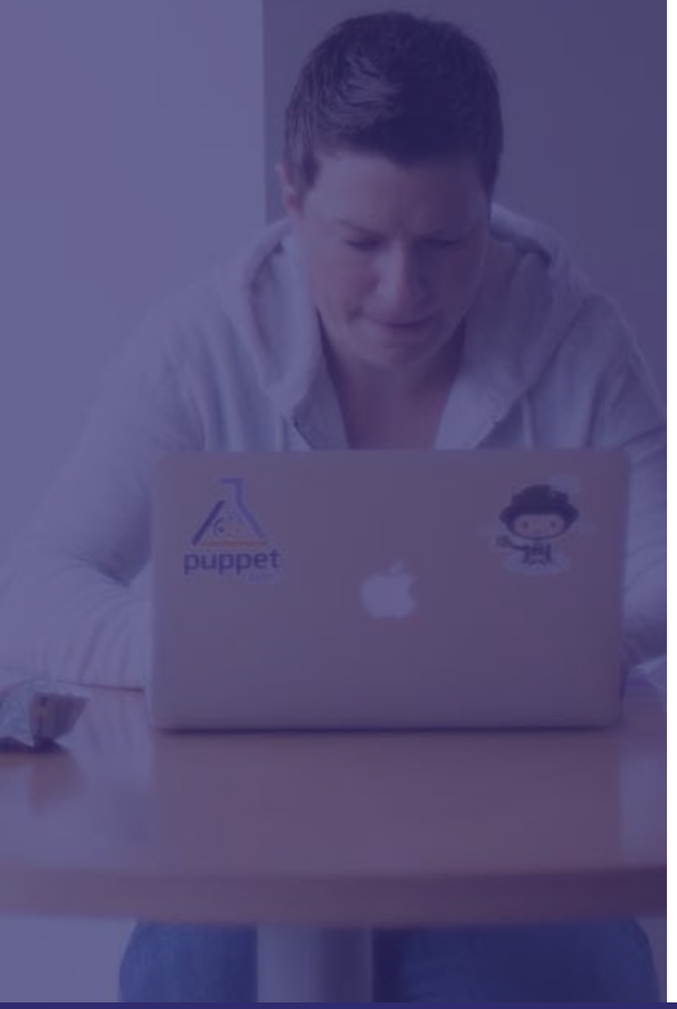


5. Application Architecture & Developer Productivity

Anyone who's spent time studying or thinking about design won't be surprised to hear that architecture has a powerful impact on software quality and developer productivity. Our survey analysis shows that specific architectural characteristics correlate with high IT performance: ability to test without an integrated environment; ability for developers to get comprehensive feedback from automated tests; ability to deploy an application independent of other services it depends on; and use of a microservices architecture. These characteristics are also what we see in enterprise architecture designed for continuous delivery, so it's not surprising that architectures like this deliver better IT performance and more deployments per developer per day.

Effective Use of Cloud & Containers Requires a Modular Architecture

This finding is of particular importance for teams that want to leverage cloud computing and containers. Effective use of these technologies requires a modular architecture consisting of loosely coupled, well-encapsulated components with clearly defined and enforced interfaces.



A major criticism of DevOps is that it can be applied only to greenfield projects. Our research shows it doesn't matter how old or new your systems are; high performance can be achieved if the application is architected for testability and deployability. So if you think you can't implement DevOps practices because your app runs on a mainframe, think again. Don't focus on the type of system you have: Instead, focus on re-architecting for testability and deployability.

We found that certain architectural characteristics correlated with high performance. A larger proportion of high-performing teams answered the following questions in the affirmative, compared to medium- and low-performing teams, both for the primary application or service they were working on and for the services they had to interact with:

- We can do most of our testing without requiring an integrated environment.
- We can and do deploy/release our application independently of other applications/services it depends on.
- It is custom software that uses a microservices architecture.

This validates our belief that achieving high levels of throughput and stability requires an enterprise architecture designed with continuous delivery in mind. Such an enterprise architecture must be designed to give developers comprehensive feedback from automated tests without relying on complex, integrated environments. And it must be possible to deploy systems independent of any other services they interact with.

Any service-oriented architecture worth its name should have these characteristics, though sadly, many do not. These requirements have been emphasized with the recent rise of microservices, and our data indicates that this kind of architecture is correlated with higher levels of IT performance.

DevOps in Practice

"Starting our DevOps journey was largely delayed by the sheer scope and size of the challenge ahead. Trying to effect process, people, technology and cultural changes across the entire application portfolio, in a globally dispersed team and with a lot of associated technical debt, is an epic challenge. If you think about deployment, testing, moving to Agile, spinning up new environments, instituting new version control strategies, etc., it's just too much to do in one hit."

— Jonathan Fletcher

Read the full story >

In contrast, low performers were more likely to say that the software they were building — or the set of services they had to interact with — was "custom software developed by another company (e.g., an outsourcing partner)." This emphasizes the importance of making IT a strategic, core focus of your business if you need to move fast while also ensuring stability. Low performers were also more likely to be working on mainframe systems — although interestingly, having to integrate against mainframe systems was not a statistically significant indicator of performance.

We also asked about a number of other candidate architectures, and obtained results that were not statistically significant in terms of performance, regardless of whether you were building or integrating against them:

- Packaged commercial software / COTS.
- Systems of record (systems that log transaction information and keep information in order).
- Systems of engagement (systems that leverage mobile, social, cloud, and big data to deliver apps and smart products to end users).
- New, not-yet-deployed systems.
- Software with an embedded component that runs on a manufactured hardware device (e.g.,, a printer firmware).
- Software requiring a user-installed component that runs on the user's machine (including mobile apps).

These results were surprising — we had expected teams working on packaged software, systems of record, or embedded systems to perform worse, and teams working on systems of engagement and greenfield systems to perform better. This reinforces the importance of focusing on architecture rather than other characteristics of the systems you work with. Even packaged software and systems of record can be evolved and operated using DevOps principles and practices, provided sufficient attention has been paid to their design and to the enterprise architecture of the ecosystem where they live.

Figure 7

Application architecture by IT performance clusters

Results are derived from the question:

Which of the following apply to the architecture of the primary application or service you are working on? (Select all that apply)

Note: Only items that are significantly different between performance clusters are shown.

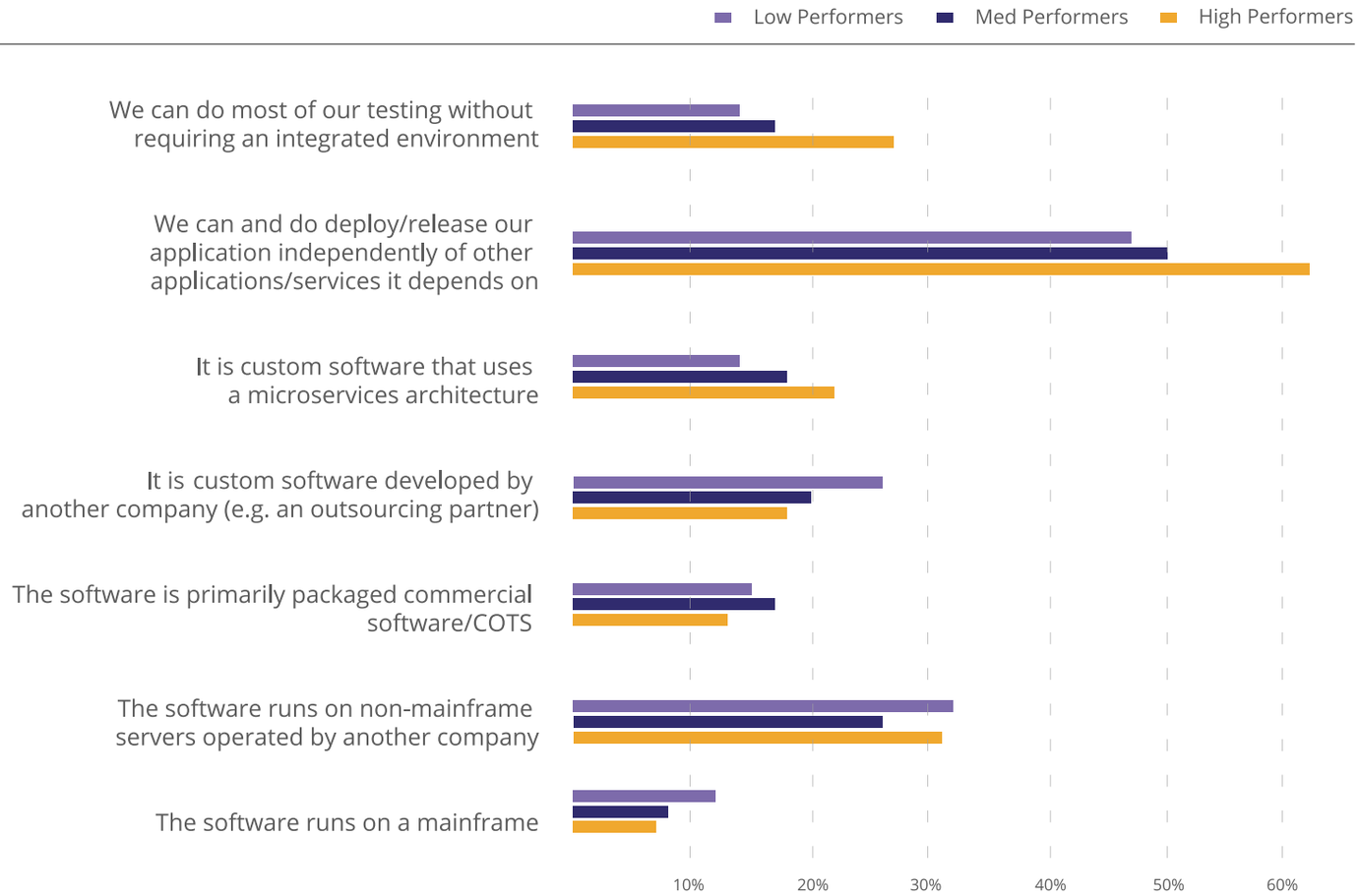


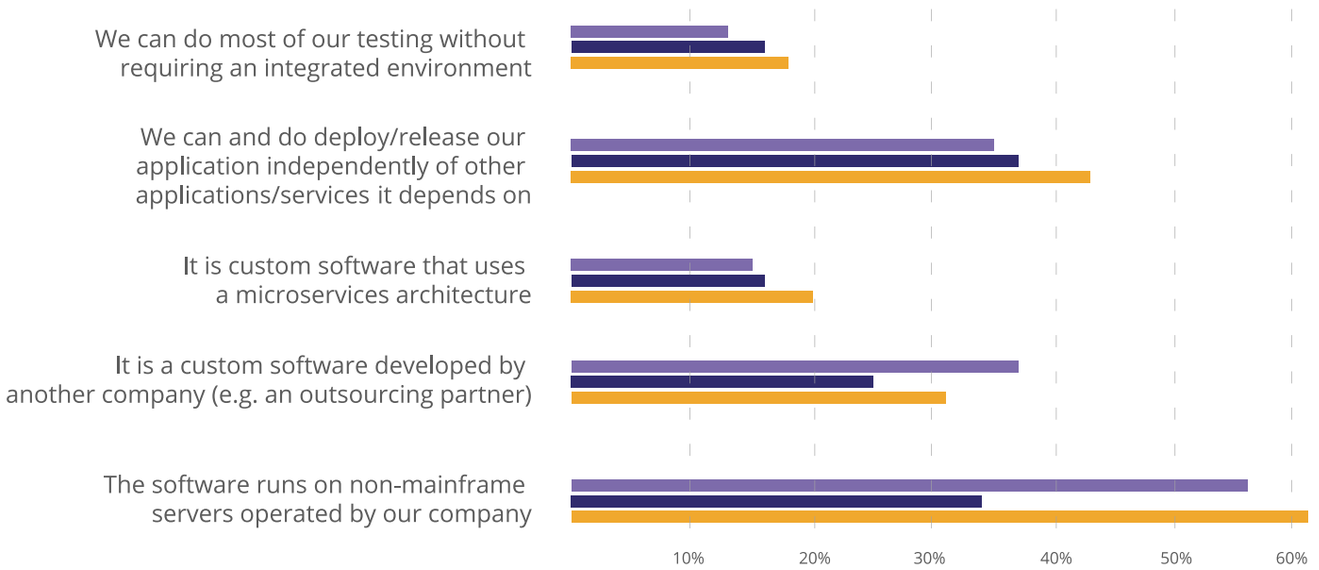
Figure 8

Services architecture by IT performance clusters

Results are derived from the question:

For services that your team has to interact with (new and existing), which of the following architecture attributes apply? (Select all that apply)

Note: Only items that are significantly different between performance clusters are shown.



The Link Between Architecture & Developer Productivity

The orthodox view of scaling software development teams states that while adding developers to a team may increase overall productivity, individual developer productivity will in fact decrease due to communication and integration overheads. A particularly painful case was highlighted in the famous book by Frederick Brook, *Mythical Man-Month.*[5] When projects are late, adding more developers not only decreases individual developer productivity, but also decreases overall productivity.

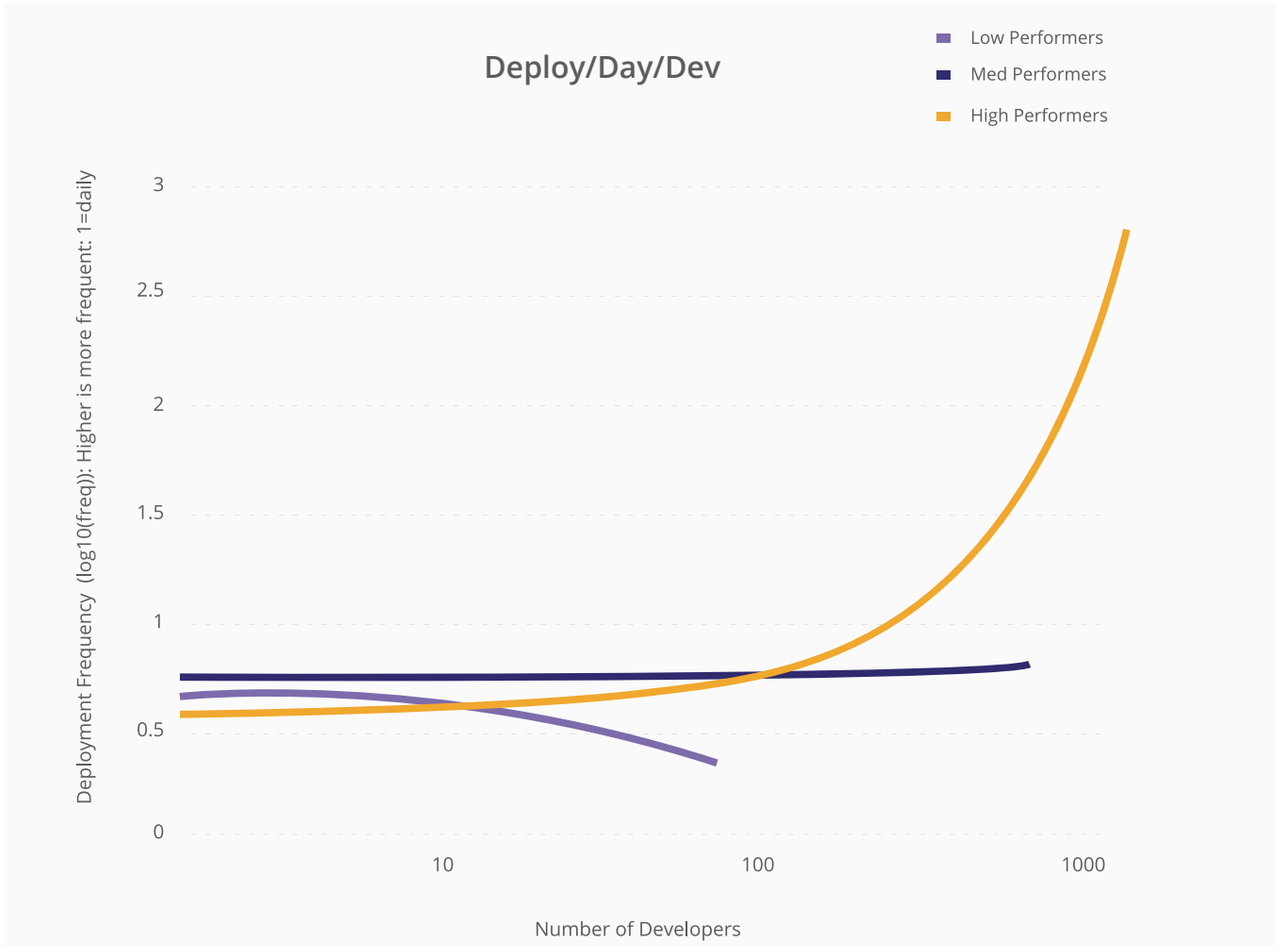
Given the type of modular architecture described in the previous section in which developers and operations work together to continuously integrate and deploy code and environments, checking their changes into source control trunk at least daily, small changes can quickly be independently developed, integrated, tested and deployed into production without causing global chaos and disruption.

As we looked at the data showing that development productivity can scale as you add more developers, we wondered whether the important variable was not just "deploys per day," but rather "deploys per day per developer." We tested for this measurement in this year's survey. The graph below represents only those respondents who deploy at least once per day. We were interested in investigating whether these teams exhibited the same characteristics as large WebOps shops. As the number of developers increases, we found:

- Low performers (light purple line) deploy with decreasing frequency.
- Medium performers (dark purple line) deploy at a constant frequency.
- High performers (yellow line) deploy at a significantly increasing frequency.

Figure 9

Number of deployments per day per developer



In other words, to scale deployments per day per developer, we need to focus on all the factors that predict high IT performance: a goal-oriented generative culture, a modular architecture, the engineering practices that enable continuous delivery, and effective leadership.

[5] Parts of this section are excerpted from the upcoming *DevOps Cookbook* by Gene Kim, Patrick Debois, John Willis, and Jez Humble, (IT Revolution, 2015).

6. How IT Managers Can Help Their Teams Win

This year's survey results show that the top three predictors of organizational performance are culture, investment in DevOps and IT performance. All of these affect each other, too. IT leaders can do a lot to improve their organization's performance by paying attention to all three of these areas. When it comes to culture, leaders can improve matters by enabling specific DevOps practices and by visibly investing in DevOps and in their employees' professional development. Managers can also facilitate big improvements in IT performance by taking measures to make deployments less painful. Last but not least, IT managers should make performance metrics visible and take pains to align these with organizational goals, and should delegate more authority to their employees. Knowledge is power, and you should give power to those who have the knowledge.

Why Culture Matters

Almost everyone agrees that culture is the most important ingredient of DevOps. The challenge for most IT leaders is defining and communicating a vision of beneficial culture for their organizations, and then facilitating the changes needed to achieve that.

Because culture is hard to define, measure and discuss, let alone change, many organizations instead default to implementing frameworks such as ITIL and COBIT, and adopting the tools that are buzzword-compliant with those frameworks. But adopting a framework wholesale, without reference to the particular issues of the workplace, often makes life more difficult for everyone. The pace of throughput slows down, productivity declines, and the morale of the team is undermined.

We felt there must be a better way, so in last year's report we measured both culture and organizational performance, and demonstrated a strong link between the two. In order to measure culture, we used a model developed by Dr. Ron Westrum, a sociologist who studied the effects of organizational culture on safety in healthcare and aviation. Westrum defines culture thusly:

The organisation's pattern of response to the problems and opportunities it encounters. Three dominant types — pathological, bureaucratic, and generative — are described. These types are shaped by the preoccupations of the unit's leaders. The workforce then responds to these priorities, creating the culture.

In other words, team leaders shape the culture according to their own proclivities, by creating incentive structures that reward certain behaviors. These incentive structures also affect how team members process and share information, cooperate and collaborate.

Westrum's model doesn't deal with all aspects of culture; it is limited to factors that influence the flow of information through the organization. The table below shows the three types of organizations.

Figure 10

Typology of Organizational Culture (Westrum, 1994)

Pathological *Power-oriented*	**Bureaucratic** *Rule-oriented*	**Generative** *Performance-oriented*
Low cooperation	Modest cooperation	High cooperation
Messengers shot	Messengers neglected	Messengers trained
Responsibility 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

Figure 11

How to create a generative culture

Characteristics of a Generative Culture	DevOps Practices
High cooperation	**Cross-functional teams.** Many organizations create cross-functional teams that include representatives from each functional area of the software delivery process (business analysts, developers, quality engineers, ops, security, etc.). This allows everyone to share the responsibility for building, deploying and maintaining a product.
Messengers trained	**Blameless postmortems.** By removing blame, you remove fear; and by removing fear, you enable teams to more effectively surface problems and solve them. Mistakes happen. Holding blameless postmortems is a valuable way to learn from mistakes.
Risks are shared	**Shared responsibilities.** Quality, availability, reliability and security are *everyone's* job. One way to improve the quality of your services is to ensure that devs share responsibility for maintaining their code in production. The improvement in collaboration that comes from sharing responsibility inherently reduces risk: With more eyes on the software delivery process, it's a given that some errors in process or planning will be avoided. Automation also reduces risk, and with the right tool choice, can enable collaboration.
Bridging encouraged	**Breaking down silos.** In addition to creating cross-functional teams, techniques for breaking down silos can include co-locating ops with the dev team; including ops in planning throughout the software delivery lifecycle; and implementing ChatOps.[6]
Failure leads to inquiry	**Blameless postmortems.** Our response to failure shapes the culture of an organization. The more you focus on the conditions in which failures happen, as opposed to blaming individuals for failures, the closer you'll get to creating a generative culture.
Novelty implemented	**Experimentation time.** Giving employees freedom to explore new ideas can lead to great outcomes. Some companies give engineers time each week for experimentation. Others host internal hack days or mini-conferences to share ideas and collaborate. This is how many new features and products have originated, and it shows how much value employees can generate for an organization when they're released from habitual pathways and repetitive tasks.

[6] Newland, Jesse. "ChatOps at GitHub," Rubyfuza presentation, March 16, 2013, https://www.youtube.com/watch?v=NST3u-GjjFw.

DevOps in Practice

"Our brand of DevOps meshes with the collaborative culture of the company. [Our] culture is all about candor, collaboration, creative challenges, and courage to move the needle. It's about initiating new concepts, new ideas, and new compelling stories we want to tell... We have to promote positive disruption, so our business doesn't get stuck, and can move into the future."

— Jason Cox

Read the full story >

Westrum's model gives us the language to define and measure culture. Perhaps most interesting, Westrum's model also predicts IT performance. This shows that information flow isn't just essential to safety, it's also a critical success factor for rapidly building and evolving resilient systems at scale.

Building on last year's findings, this year we wanted to investigate how organizations can move towards a high-performance, generative culture — especially large, regulated, slow-moving organizations that have complex, heterogeneous systems.

We discovered the top seven measures with the strongest correlation to organizational culture are:

1. Organizational investment in DevOps.
2. The experience and effectiveness of team leaders.
3. Continuous delivery practices.
4. The ability of development, operations, and infosec teams to achieve win-win outcomes.
5. Organizational performance.
6. Deployment pain.
7. Lean management practices (see previous section).

This list contains several constructs that are new in this year's survey: organizational investment in DevOps, effective leadership, deployment pain, and lean management practices. We'll discuss these constructs and their impact on organizational culture and performance in the rest of this section.

Organizational Investment in DevOps

One of the factors we wanted to understand was the level and type of investment in DevOps happening in organizations, and how that was related to the success and impact of DevOps practices. Organizational investment was measured based on answers to the following items:

- We plan to invest in tools that support DevOps initiatives.
- We plan to invest in training and development for technical staff.
- We already have training available for technical staff.
- DevOps is a high priority in my organization.

These statements may sound somewhat general, but the data shows they are surprisingly important. In fact, even *plans* to invest in DevOps can send a strong signal to teams throughout an organization.

We found that organizational investment in DevOps is strongly correlated with organizational culture; the ability of development, operations, and infosec teams to achieve win-win outcomes; lower levels of burnout; more effective leadership; and effective implementation of both continuous delivery and lean management practices. Organizational investment in DevOps is also predictive of organizational performance.

Why does organizational investment have such an outsize impact on outcomes? First of all, simply making an initiative high priority and communicating that fact to the organization is already a big deal. In organizational change expert John Kotter's book, *Leading Change*, he lists "establishing a sense of urgency" as the first and most important of eight steps required to create effective, lasting change.

Second, it's unrealistic to expect people to change they way they work without organizational support — and budget. Today, many organizations expect people to acquire new skills and adopt new tools without adequate funding, and even without allowing for any extra time, all while continuing to be productive.

In contrast, high-performing organizations commit to investments in training and development, in addition to tools. Our data shows these organizations are more likely to have a generative culture and achieve better outcomes.

There are a number of ways IT leaders can invest in their teams:

- Establish a dedicated training budget and make sure people know about it. Also, give your staff the latitude to choose training that interests them.
- Encourage staff to attend technical conferences at least once a year and summarize what they learned for the entire team.
- Set up internal hack days, where cross-functional teams can get together to work on a project.
- Hold regular internal DevOps mini-conferences.
- Give staff dedicated time, such as 20-percent time or several days after a release, to experiment with new tools and technologies. Allocate budget and infrastructure for special projects.

Deployment Pain

Do you and your staff dread deployments? Are you afraid changes will break everything, taking your services down, forcing people to work overtime, and attracting unwelcome attention from upper management?

You aren't alone. With traditional methods of software development, deployments are often infrequent, painful, and disruptive events. In organizations practicing DevOps, deployments are regular, pain-free and dependable. We wanted to understand how you could go from one extreme to the other, so we created a new construct called deployment pain, which consists of three items:

- Code deployments are not feared.
- Code deployments are extremely disruptive and my team and I fear them.[7]
- Code deployments are relatively pain-free.

Statistical analysis revealed a high correlation between IT performance and deployment pain: The more painful code deployments are, the poorer the IT performance, organizational performance and culture. The data also tells us that painful deployments result in higher change fail rates.

It was gratifying, though unsurprising, to find that deployment pain was predicted by whether the key continuous delivery practices had been implemented: comprehensive test and deployment automation, the use of continuous integration including trunk-based development, and version control of everything required to reproduce production environments. A generative organizational culture was also highly correlated with low deployment pain.

Quick temperature gauge: How painful are your deployments?
If you want to know how your team is doing, all you have to do is ask your team how painful deployments are and what specific things are causing that pain.

Common problems include:

- Changes that often result in failures and are difficult to diagnose and fix.
- Dev, test, and staging environments that are different from production environments, causing failures when builds are promoted across environments.
- Lots of manual work required to deploy.
- Lots of handoffs between teams, resulting in slow, inefficient deployments.

The countermeasures that should be implemented include:

- Do smaller deployments more frequently (i.e., decrease batch sizes).
- Automate more of the deployment steps.
- Treat your infrastructure as code, using a standard configuration management tool.
- Implement version control for all production artifacts.
- Implement automated testing for code and environments.
- Create common build mechanisms to build dev, test and production environments.

[7] This item was reverse coded for statistical analysis.

Create Leaders & Delegate Authority

Effective leadership plays a critical role in shaping the cultural environment of any organization. Respondents who reported strong agreement with the following statements also reported a generative organizational culture (according to Westrum's model, see Figure 11):

- My team leader is effective at what they do.
- My team leader has significant experience in the work my team does.
- My team leader is a recognized expert on the team.
- My team leader has led similar efforts in the past.

Effective leadership was also strongly correlated with:

- **Helping teams achieve win-win outcomes**. One common source of adversarial relationships between departments is when they are measured in ways that encourage them to focus on team-level goals — for example, "How fast can I declare my code dev complete?" — rather than organization-level or customer-focused goals such as, "How fast can we successfully resolve bugs?"

- **Creating feedback loops**. Teams with effective leadership use data from application and infrastructure monitoring tools to make business decisions daily. Looping from production back into what teams work on is critical to creating high-performance teams.

- **The use of key continuous delivery practices**. Continuous integration and the use of comprehensive configuration management and test automation were also highly correlated with effective leadership. That's consistent with our hypothesis that effective leaders encourage their teams to invest in ongoing improvement work. Effective leaders are those who demonstrate by their actions that the *improvement* of daily work is more important than the daily work.

Effective leaders encourage their teams to invest in ongoing improvement work.

Leadership, like culture, is also hard to define and measure. David Marquet, former U.S. Navy captain and author of *Turn the Ship Around!*, gives us this remarkable definition of leadership:

Leadership (n): Embedding the capacity for greatness in the people and practices of an organization, and decoupling it from the personality of the leader.

The secret to his methodology is simple: "Move the authority to where the information is." This is absolutely critical if you want to scale. People must have the authority to act, and they can act wisely only if they have the information they need to do so. In fact, it's the people on the ground who really know what needs to be done. The role of the leader should be to trust and enable those who know what needs to be done.

Here are some ways to delegate authority to where the information is:

- **Make metrics visible and actionable.** Many organizations claim to be data-driven, meaning they collect a lot of data, but very few actually make decisions based on that data. Do you regularly review metrics and take action on them, or are they primarily vanity metrics? To achieve organizational clarity, you need to do two things:
  - Make sure performance metrics are aligned with organizational goals, rather than team or functional goals.
  - Turn data into actionable, visible information that provides teams with feedback on key quality, performance, and productivity metrics.

- **Manage work in process**. Give teams control over their work in process, and the authority to limit it so they are not overburdened and can get work completed quickly and sustainably.

- **Support employees**. Effective leaders help people grow and learn as part of their work, and support their decisions even when they produce poor outcomes — so long as they learn from them.

DevOps in Practice

Read the full story >

"Everyone is responsible for quality and we're all trying to deliver the best solution for our customers... Constant discussion between software and infrastructure teams really helps us proactively find issues before deploying to production. We all benefit from working together more efficiently. Everyone is interested in making our services better, and everyone is thinking bigger scale. We're encouraging people to ask the right questions to understand what we need to deliver for the future. That's a big win for me."

— Reena Mathew

#WomenInTech

We introduced a few new questions on gender, which sparked some lively discussion in social media on the topic of women in tech. We heard everything from wholehearted support from many women and men in the DevOps community, to questions about why gender diversity in tech matters. Of the total respondents, 5 percent self-identified as female. This was much lower than we expected, given that women make up about 13 percent in systems administration[8] and 27 percent in computer and information management[9]. We were hoping to find more reassuring numbers of women working on technical teams, but we didn't. Among survey respondents:

- 33 percent report working on teams with no women.
- 56 percent report working on teams that are less than 10 percent female
- 81 percent report working on teams that are less than 25 percent female.

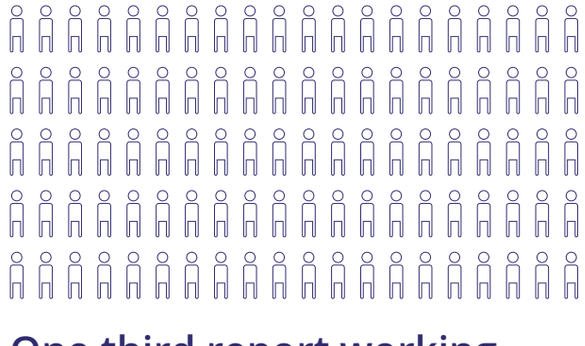
There's plenty of research linking the presence of women in leadership positions to higher financial performance,[10] stock market performance,[11] and hedge fund returns.[12] Furthermore, a study conducted by Anita Woolley and Thomas W. Malone measured group intelligence and found that teams with more women tended to fall above average on the collective intelligence scale.[13] Despite all of these clear advantages, organizations are failing to recruit and retain women in technical fields.

If there are no significant differences between men and women in terms of ability or aptitude in STEM (Science, Technology, Engineering and Mathematics) fields, what's keeping women and other underrepresented groups out of tech? Nothing more than the pervasive belief that some men are naturally more suited to technical work because they possess innate brilliance.[14]

We can do better. It's up to all of us to prioritize diversity and promote inclusive environments. It's good for your team and it's good for the business. Here are some resources to help you get started:

- Ada Initiative. Provides Allies Workshops, codes of conduct, and anti-harassment policies.
- Anita Borg Institute. Excellent tools for advancing women in technology.
- Geek Feminism. Great resources for supporting women in geek communities.

More than half report working on teams that are less than 10% female.



One third report working on teams with no women.

---

[8] SAGE Annual Salary Survey for 2007, The USENIX Association, https://www.usenix.org/system/files/lisa/surveys/sal2007_0.pdf.

[9] Diaz, Von and King, Jamilah. "How Tech Stays White," *Colorlines*, Oct 22, 2013, http://www.colorlines.com/articles/how-tech-stays-white.

[10] McGregor, Jena. "More women at the top, higher returns," *The Washington Post*, September 24, 2014, http://www.washingtonpost.com/blogs/on-leadership/wp/2014/09/24/more-women-at-the-top-higher-returns/.

[11] Covert, Bryce. "Companies With Female CEOs Beat The Stock Market," *ThinkProgress*, July 8, 2014, http://thinkprogress.org/economy/2014/07/08/3457859/women-ceos-beat-stock-market/.

[12] Covert, Bryce. "Returns for Women Hedge Fund Managers Beat Everyone Else's," *ThinkProgress*, January 15, 2014, http://thinkprogress.org/economy/2014/01/15/3168521/women-hedge-funds-returns/.

[13] Woolley, Anita and Malone, Thomas W. "Defend Your Research: What Makes a Team Smarter? More Women," *Harvard Business Review*, June 2011, https://hbr.org/2011/06/defend-your-research-what-makes-a-team-smarter-more-women.

[14] Leslie, Sarah-Jane, et al. "Expectations of brilliance underlie gender distributions across academic disciplines," Science 347:6219, (January 16, 2015), 262–265.

7. Burnout

Just as in any industry where the work is high tempo and high consequence, IT is plagued by employee burnout. IT managers, like so many other managers, often try to fix the person while ignoring the work environment, even though changing the environment is far more vital for long-term success. Managers who want to avert employee burnout should concentrate their attention and efforts on fostering a respectful, supportive work environment that emphasizes learning from failures rather than blaming; communicating a strong sense of purpose; investing in DevOps and employee development; asking employees what causes deployment pain and then fixing those things; and giving employees time and space to experiment and learn. Last but not least, employees must be given the authority to make decisions that affect their work and their jobs.

As in other high-tempo, high-consequence work, burnout is an important issue in IT, with serious repercussions for the mental and physical health of practitioners. Research shows that stressful jobs can be as bad for physical health as smoking and obesity. Symptoms of burnout include feeling exhausted, cynical or ineffective; feeling little or no sense of accomplishment in your work; and feelings about your work negatively affecting the rest of your life. In extreme cases, burnout can lead to family issues, severe clinical depression and even suicide.

Job stress also affects employers, costing the U.S. economy $300 billion per year in sick time, long-term disability and excessive job turnover.[15] Thus, employers have both a duty of care towards employees and a fiduciary obligation to ensure staff do not become burned out.

Christina Maslach, professor of psychology at the University of California at Berkeley and a pioneering researcher on job burnout, found six organizational risk factors that predict burnout:[16]

- **Work overload.** Job demands exceed human limits.
- **Lack of control.** Inability to influence decisions that affect your job.
- **Insufficient rewards.** Insufficient financial, institutional or social rewards.
- **Breakdown of community.** Unsupportive workplace environment.
- **Absence of fairness.** Lack of fairness in decision-making processes.
- **Value conflicts.** Mismatch in organizational values and the individual's values.

Maslach found that most organizations try to fix the person and ignore the work environment, even though data shows that fixing the environment has a higher likelihood of success. All of the risk factors above are things management has the power to change.

---

[15] Maslach, Christina. "Understanding Burnout," Thriving in Science Lecture, December 10, 2014, https://www.youtube.com/watch?v=4kLPyV8lBbs.
[16] Maslach, Christina and Leiter, Michael P. "Early predictors of job burnout and engagement," *Journal of Applied Psychology* 93:3, (2008), 498–512.

Our research also tells us which organizational factors are most strongly correlated with high levels of burnout, and suggests where to look for solutions. The five most highly correlated factors are:

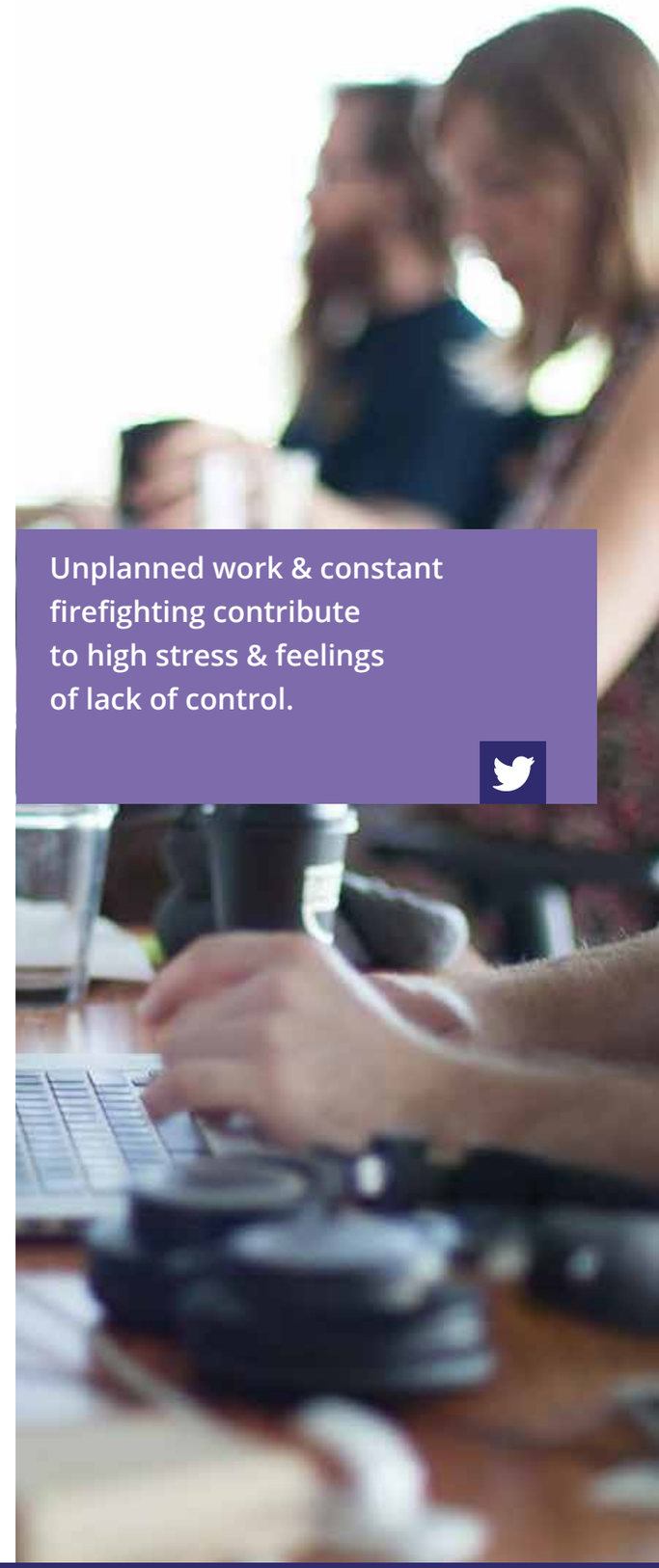
- **Organizational culture.** Strong feelings of burnout are found in organizations with a pathological, power-oriented culture. Managers are ultimately responsible for fostering a supportive and respectful work environment, and they can do so by creating a blame-free environment, striving to learn from failures, and communicating a shared sense of purpose.

- **Deployment pain.** Unplanned work and constant firefighting contribute to high stress and feelings of lack of control. With the right practices in place, deployments don't have to be painful events. Managers should be asking their teams how painful their deployments are, and fixing the things that hurt the most.

- **Effectiveness of team leader.** Responsibilities of a team leader include limiting work in process and eliminating roadblocks for the team so they can get their work done. It's not surprising that respondents with effective team leaders reported lower levels of burnout.

- **Organizational investment in DevOps.** Organizations that invest in developing the capabilities of their teams get better outcomes. Investing in training and providing people with the necessary support and resources (including time) to acquire new skills are critical to the successful adoption of DevOps.

- **Organizational performance.** Our data shows that lean management and continuous delivery practices predict IT performance, which in turn predicts organizational performance. At the heart of lean management is giving employees the necessary time and resources to improve their own work. This means creating a work environment that supports both experimentation and failure, and allowing employees to make decisions that affect their jobs.



Unplanned work & constant firefighting contribute to high stress & feelings of lack of control.

8. Methodology

The State of DevOps Survey has evolved over the past four years. Our current rigorous methodology was established last year,and has given us a rich data set that tells us a great deal about the relationships between IT performance, organizational performance, DevOps and lean practices. In this section, we describe how we enlisted survey respondents — a mix of IT practitioners and managers, developers and testers — how we designed our questions, models and constructs, and our analysis methods. We welcome any questions about our survey methodology, so please feel free to get in touch: devopssurvey@puppetlabs.com.

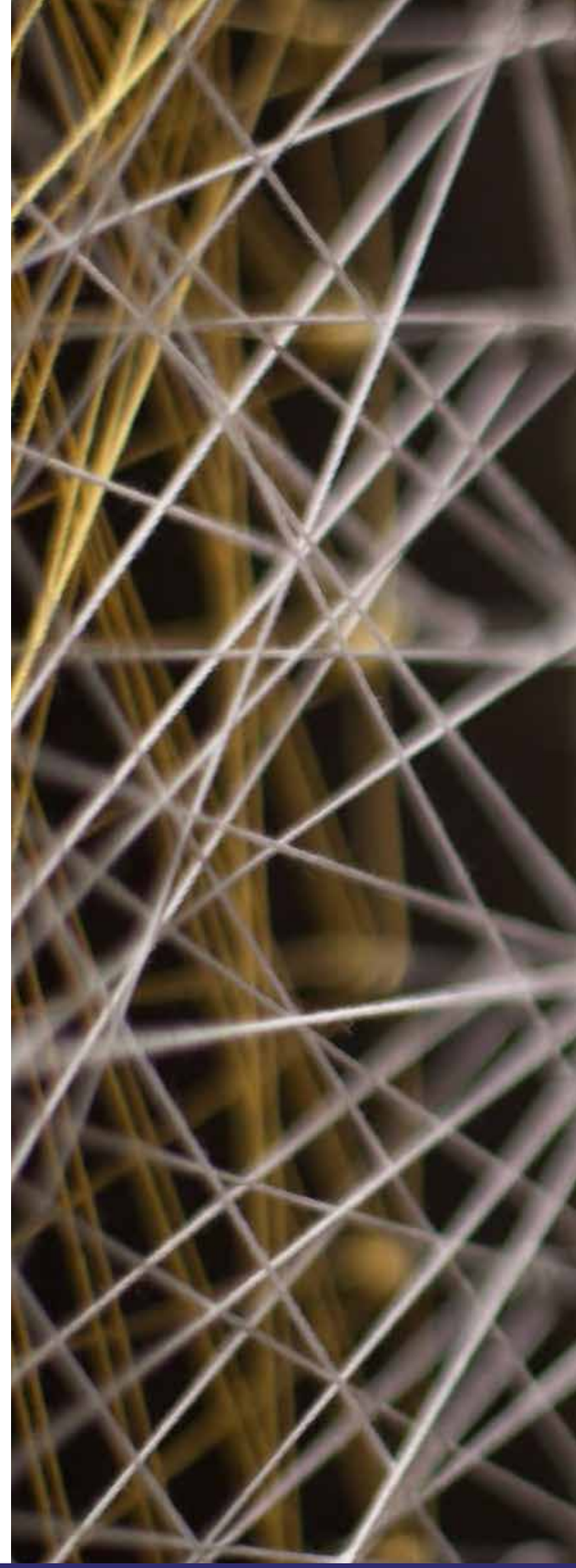
Target Population & Sampling Method

Our target population for this survey was practitioners and leaders working in (or closely with) IT, and especially those familiar with DevOps. Because we don't have a master list of these people — we can describe them, but we don't know exactly where they are, how to find them and how many of them exist — we used snowball sampling to obtain respondents. This means we promoted the survey via email lists, online promotions and social media, and also asked people to share with their networks, growing the sample like a snowball. Our sample is likely limited to organizations and teams that are familiar with DevOps, and as such, may be using some DevOps practices.

Creating Latent Constructs

Once the target population and sampling method were determined, we began the difficult work of determining which questions to include in the survey. To do that, we first had to figure out which hypotheses we wanted to test.

To add to the rigor of our study, we referenced existing research and theories. We formulated our hypotheses and constructs, using previously validated constructs wherever possible. When we needed to create new constructs, we wrote them very carefully based on theory, definitions and expert input. We then took additional steps to clarify intent and wording to ensure that data collected from the final survey would have a high likelihood of being reliable and valid.[17] To create constructs, we used Likert-type[18] questions, which provided shades of gray, rather than black-and-white, yes-or-no, true-or-false questions. Likert-type questions also make it possible to perform more advanced analysis.



---

[17] We used Churchill's methodology: Churchill Jr, G. A. "A paradigm for developing better measures of marketing constructs," *Journal of Marketing Research* 16:1, (1979), 64–73.

[18] "Likert scale," Wikipedia, last modified June 25, 2015, https://en.wikipedia.org/wiki/Likert_scale.

Methods

- **Measurement Model.** Prior to conducting any analysis using constructs — including correlations, regressions, and partial least squares (PLS)[19] analysis — the constructs were tested for validity and reliability. The constructs passed checks for convergent validity[20], discriminant validity[21], and reliability, therefore exhibiting good psychometric[22] properties.

- **Regression Analysis.** When predictions or impacts are cited and PLS is not explicitly mentioned, a simple linear regression[23] was used.

- **Structured Equation Modeling.** The structured equation models (SEMs)[24] were tested using PLS analysis, which is a correlation-based SEM well-suited to exploratory analysis. SmartPLS 3.2.0 was used. All paths shown in figures 5 and 6 are $p < .001$.

- **Study Design.** This study employs a cross-sectional, theory-based design.

---

[19] "Partial least squares regression," Wikipedia, last modified May 26, 2015, https://en.wikipedia.org/wiki/Partial_least_squares_regression.
[20] "Convergent validity," Wikipedia, last modified January 7, 2015, https://en.wikipedia.org/wiki/Convergent_validity.
[21] "Discriminant validity," Wikipedia, last modified June, 22, 2015, https://en.wikipedia.org/wiki/Discriminant_validity.
[22] "Psychometrics," Wikipedia, last modified June, 27, 2015, https://en.wikipedia.org/wiki/Psychometrics.
[23] "Linear regression," Wikipedia, last modified July 3, 2015, https://en.wikipedia.org/wiki/Linear_regression.
[24] "Structural equation modeling," Wikipedia, last modified May 26, 2015, https://en.wikipedia.org/wiki/Structural_equation_modeling.

Be the DevOps you want to see in the world.

About Puppet Labs

Puppet Labs, Inc. is the leader in IT automation. Puppet Labs software provides system administrators the operational agility, efficiency and insight they need to proactively manage dynamic infrastructure, scaling from tens of servers to thousands, both on premise and in the cloud. Thousands of the world's leading organizations use Puppet Labs software to configure and manage their IT infrastructure. Learn more about Puppet Labs at www.puppetlabs.com.

About IT Revolution

IT Revolution assembles technology leaders and practitioners through publishing, events, and research. Our goal is to elevate the state of technology work, quantify the economic and human costs associated with suboptimal IT performance, and to improve the lives of 1 million IT professionals by 2017. Learn more about IT Revolution at www.itrevolution.com.

Say hello

Thanks to all of the Puppet Labs employees whose photos are featured in this report. All photos were taken at our headquarters in Portland, OR.

Sponsored by

About PwC

PwC US helps organizations and individuals create the value they're looking for. We're a member of the PwC network of firms, which has firms in 157 countries with more than 195,000 people. We're committed to delivering quality in assurance, tax and advisory services. Find out more and tell us what matters to you by visiting us at www.pwc.com/us/highvelocityit.