

2019 Global Developer Report: DevSecOps



What's inside?

Introduction

DevSecOps 2019 — Mission Improvement

Development — Mission Acceleration

- » A day in the life
- » DevOps to the rescue
- » Security
- » Remote work(s)

Security — Mission Readiness

- » Security in action
- » A question of ownership
- » DevSecOps
- » Remote and secure

Operations — Mission Clarity

- » The choice of tools
- » Ops and DevOps
- » Staying secure
- » Why remote works for ops

Trajectory

Debrief

Introduction

Created to encourage conversation and collaboration, the Global Developer Report: DevSecOps dissects the cross-functional relationships of DevOps teams and offers insights into successful practices, problem areas, and potential solutions.

This year, over 4,000 respondents – across various industries, roles, and geographic locations – candidly shared their experiences, helping us uncover what software professionals require in order to innovate rapidly.

By uncovering best practices and unmet needs, the Global Developer Report: DevSecOps is one small step for software professionals to share their thoughts, and one giant leap for IT leaders to remove roadblocks to help teams thrive and offer the strongest contributions to software development. Using these insights as a guide, IT leaders can employ a solution-focused approach to creating a seamless software development lifecycle – from planning to monitoring.

All this guidance comes with a reminder that our results largely reflect the experiences of DevOps pioneers – developers from smaller companies primarily in the technology sector who are farther down this road than is typical. The data from our survey is both aspirational and reflective of the reality of DevOps as a challenging and time-consuming practice to achieve, even in smaller, tech-oriented companies. Your experiences may vary, however, and that should not be discouraging. Also, it's important to keep in mind that 60% of our survey respondents are GitLab users so it's not surprising they rate our tool highly.

In past years, our research focused on the culture, workflow, and tools that inspired developers to do their best work. Understanding the growing, critical importance of collaboration between teams to deliver software, we expanded our research this year to include operations and security groups. These additional insights allow us to take a comprehensive assessment of the entire software development lifecycle, providing recommendations on how teams can harmoniously deliver software and value to their organizations and customers. This analysis enables us to clear the launch pad for liftoff.

DevSecOps 2019

Mission Improvement

Through the survey earlier this year, we conducted a launch status check and carefully examined 4,071 development, operations, and security crewmembers to determine what teams need in order to lift off.

To help you create an individualized flight plan for your team, we determined that the overall mission objective for all software professionals today is improvement. Software professionals often hear about the imperative to undergo a digital transformation, the importance of writing secure code, the need to increase visibility, and the urgency to reduce cycle time. All these conversations trace back to the need to improve the way teams collaborate in order to deliver value to both their organizations and customers.

What improvement looks like depends on individual teams, processes, and workflows, but embarking on a journey to enhance the way your team delivers software helps hone your competitive edge.

Gender

91% Male



Gender	Percentage
Male	91%
Female	6%
Prefer not to say	3%
Non-binary/third gender	1%
Prefer to self describe	0.4%

6% Female

3% Prefer not to say

1% Non-binary/third gender

0.4% Prefer to self describe

Industry

46% Computer Hardware / Services / Software / SaaS



Industry	Percentage
Computer Hardware / Services / Software / SaaS	46%
Business Services / Consulting	8%
Education	7%

8% Business Services / Consulting

7% Education

6% Other



5% Banking / Financial Services



5% Media & Entertainment



4% Telecommunications



3% Healthcare



Role

50% Software Developer / Software Engineer



11% Development/Engineering Leadership



7% DevOps Engineer



7% Technology Executive - CIO / CTO



6% Software Architect



4% Other



3% DevOps Leadership



3% Systems Administrator



2% Product Manager



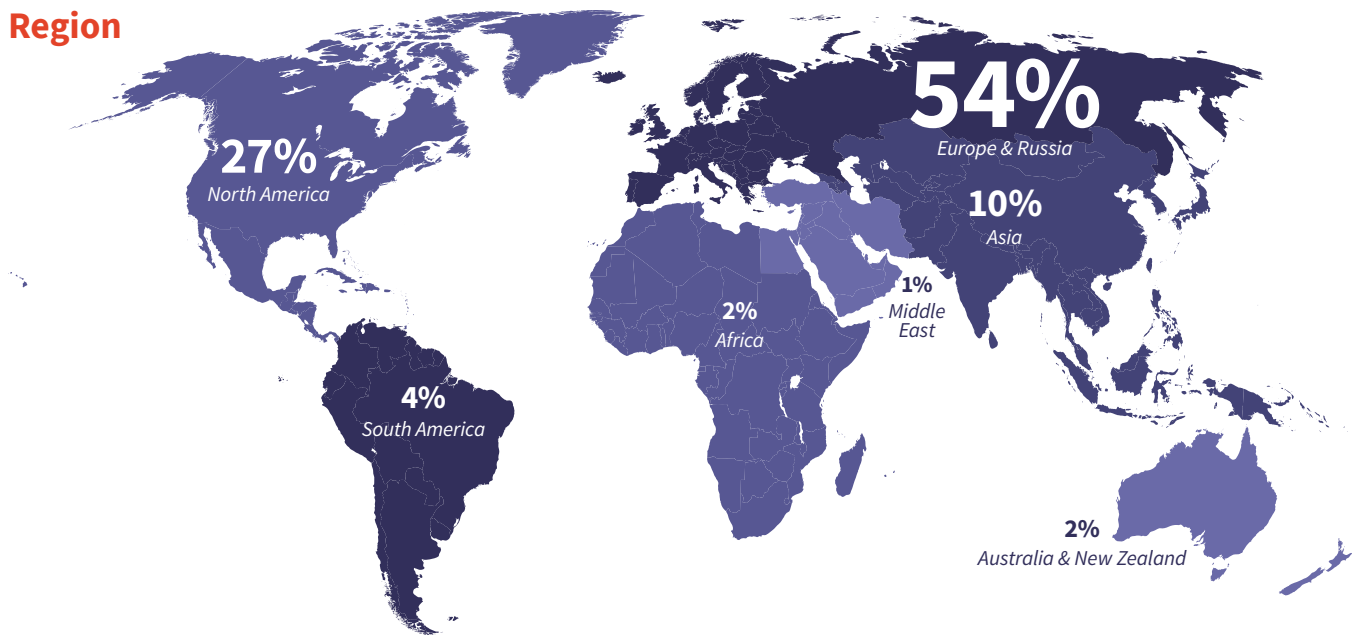
1% Systems Engineer / Network Engineer



1% Engineering Project Manager



Region



Number of employees

32% 11 - 100

24% 1 - 10

17% 101 - 500

12% 1,001 - 10,000

9% 10,000+

6% 501 - 1,000

55%
Most work at an office

25%

All work at an office

Remote or in-office

9%
Most work remotely

11%
All work remotely

2019 Global Developer Report: DevSecOps top findings

- 1. DevOps = better visibility:** Developers, operations team members, and security professionals are 89% more likely to have good insight into what their colleagues are working on when their DevOps model has been in place long term.
- 2. Continuous deployment is real:** Almost half — 45% — report continuous code deployment at least somewhere in their organization.
- 3. Security is also a work in progress:** 50% agree that security vulnerabilities are mostly discovered by the security team after code is merged and in a test environment.
- 4. Remote work supports efficiency:** Remote operations teams are 1.6x more likely to document their work than in-office counterparts.
- 5. Testing is still hard:** 49% of respondents encounter the most delays during the testing stage of the development lifecycle.

Mission description

Knowing that DevSecOps teams need to improve the way they collaborate in order to better deliver value, we developed specific missions for development, operations, and security crews. By using these missions as a guide, IT leaders can determine how to remove roadblocks and empower teams to work together.

We encourage you to review the missions of other teams to develop a stronger understanding of how you can better collaborate.

Now that you understand the mission objective, it's time to prepare for liftoff...

Launch your mission:



Development

Mission Acceleration

To help their organizations stay competitive in a rapidly changing market, development teams need to accelerate delivery. The primary focus in 2019 is to identify the biggest roadblocks to innovation.

Development top findings

- 1. DevOps sparks innovation:** Developers are 1.4x more likely to feel innovative with a mature rather than poor DevOps model.
- 2. Remote teams feel secure:** Developers with a mostly remote team rated the maturity of their organization's security practices 29% higher than those who work mostly in-office.

- 3. Continuous deployment is also on the rise:** 43% of developers report they deploy on demand or multiple times a day, and nearly the same percentage (41%) deploy between once a day and once a month.
- 4. CD = greater insight:** Those who continuously deploy code say project and product managers are 25% more likely to have better insight into developer capacity during the planning stages, compared with organizations that deploy between once per month and once every 6 months.
- 5. DevOps is taking hold:** 28% of developers feel good about the maturity of their DevOps practice although Scrum continues to be the dev methodology of choice for more than half of our respondents.
- 6. Testing is still hard:** 49% of respondents encounter the most delays during the testing stage of the development lifecycle.

For the most part, today's developers are poised to soar high and they're generally optimistic about their work now and in the future. In fact, 59% think their organization's development processes are designed to help them succeed and 63% say those same structures allow for innovation. They've got the right tools on board – 53% are happy with what they've got – and the vast majority of respondents use Scrum (50%), Kanban (37%), and DevOps (36%) versus waterfall (17%) to get the job done.

Most practiced development methodologies

54% *Scrum*



37% *Kanban*



36% *DevOps*



17% *Waterfall*



But that doesn't mean it's all smooth sailing. From constantly changing requirements to labor-intensive testing and ongoing concerns about security, developers see some room for improvement in their work lives.

How do developers rate their DevOps practices?

33% *fair*



28% *good*



17% *poor*



And while many organizations “do” DevOps, it’s clear from developer feedback just how much variation actually exists. In fact, just 28% called their DevOps implementations “good” while 33% opted for “fair” and 17% called them “poor.”

“Everything is done manually. No automation is used and Dev and Ops are fighting more than collaborating.”

“Most of us are knowledgeable in DevOps. It’s important nowadays. We want automation in everything.”

“(It’s a) very manual process. Need to figure out the right and efficient ways to automate processes (and) gain trust in DevOps.”

“In some projects, we use good DevOps practices, in other projects (we) don’t. It depends on the PM of the project. If he allows time to be spent on DevOps activities, then we do it, if not, we don’t.”

A day in the life

Take a closer look at modern development teams and it’s clear that no matter the state of DevOps or size of the company some things are universal:

Most commonly used forms of source control

95% *Git*



2% *Subversion*



1% *Team Foundation Server*



0.5% Mercurial



0.4% CVS



A decisive (and unsurprising) 95% of developers use Git for source code management (the next closest choice is Subversion at 2%).

Most used CI and build tools

61% GitLab



36% Jenkins



12% Travis CI



10% Don't use CI or build tools



Fully 61% of companies say GitLab is their most used tool for CI and build, followed by Jenkins (36%) and Travis CI (12%). (A reminder, though, that 60% of our survey takers are GitLab users.)

Half of those surveyed called out testing as the biggest source of delay in the development process, reflecting an industry-wide struggle to balance the benefits of manual testing with the need for automation. At the same time, 42% of developers said planning slowed things down, while 37% said code development was a hindrance and 27% said either code review or deploying to production.

Only 12% of developers are happy with their organization's accessibility testing solution – perhaps because 73% don't use any accessibility testing tools at all.

Nearly two-thirds of survey respondents (61%) collect user feedback data on their applications, most often using Google Analytics (46%), though more than a quarter (27%) don't use any tools to collect feedback. The most common feedback collected from users is customer satisfaction level (68%), user preferences (55%), and usability analysis (50%).

Nearly half of all developers (44%) aren't on call, while 31% get automated alerts from a monitoring system.

Despite the collaborative advances promised by DevOps, developers remain skeptical about their Ops counterparts: Only about a third of developers (36%) believe Operations team members are able to



quantify and document their work, while less than half (43%) think Ops gets sufficient advance notice to support development efforts in their organizations.

DevOps to the rescue


Continuous delivery – a cornerstone of DevOps – is an area developers see as critical.

Code deployment frequency

43% *Continuous deployment (on demand, multiple deploys per day)*




41% *Between once per day and once per month*



13% *Between once per month and once every 6 months*



3% *Don't know*



Of those surveyed, 43% said their organizations continuously deploy (meaning on-demand deployment and/or multiple deployments a day) and 41% said deployments happen between once a day and once a month. Just 13% reported deployments occurring between once a month and every six months. (These are definitely aspirational results – the percentage of continuous deployment across all industries and company sizes is between 5% and 20% according to [the latest DevOps Hype Cycle](#) from market research firm Gartner Group.)

The benefits of continuous delivery are clear: Developers say product/project managers are 25% more likely to have a better sense of dev team capacity in a CD organization than in a company that deploys between once a month and once every six months. And 47% agree those same managers are in a better position to accurately plan and scope features in a CD environment.

In a seasoned DevOps environment, developers are 1.4x more likely to feel innovative and they're also more connected to other teams: 89% say they're more likely to have visibility into what others are working on.

Having mature DevOps practices is critical to developer morale, since 88% of developers who work at organizations with very poor DevOps maturity don't feel that their development processes are designed to help them succeed. Furthermore, teams with long-standing DevOps models are 83% more likely to feel that when planning sprints, Project/Product Managers have a good sense of the capacity of different developers.

Security

Security is perhaps the most polarizing aspect of software development today and that's clear from the results of our survey. Nearly 70% of developers said they are expected to write secure code, but it's clear from the comments below that in most organizations, the mechanisms to make that happen remain elusive.

“It’s a mess, no standardization, most of my work has never had a security scan.”

“We’re starting to care about it, just now.”

“We have made various efforts over the past year, and continue to strive toward improved security.”

“We don’t have clear guidelines about security, so the different services present different levels of security.”

How do developers rate their security practices?

30% *fair*



25% *good*



23% *poor*



In fact, just 25% of developers rated security at their organization as “good,” while 30% said it was “fair” and 23% said it was “poor.”

Those poor grades on security may also reflect the high percentage of respondents working in small companies where a dedicated security team may not be an option. There may be one bright spot however: 45% of developers said they receive and are able to address security feedback during the development process.

Remote work(s)

You might think we're biased (because we're an all remote company), but working remotely really works for software developers. For starters, they're not out of the loop. In fact, developers say they're

23% more likely to be aware of what other colleagues are working on despite not being physically present in an office. Their managers aren't out of the loop either – 21% of developers said their project/product managers understand the capacity of the different developers working for them and because of that are 27% more likely to plan and scope features. And they're 24% more likely to provision their own testing environments than when working with in-office teams, no doubt because they can't count on someone else to do it for them. And perhaps because remote work brings with it the need for repeatable practices and better documentation, developers who work remotely rated the maturity of their organization's security policies 29% higher.

Security

Mission Readiness

When it comes to security, everyone – devs, ops, and security pros – is ready for more. But because security is a complicated and multi-layer endeavor, involving an entire organization, solutions are often complex and piecemeal and, as such, they can be elusive.

Security top findings

- 1. DevOps FTW:** Sec teams are 3x more likely to discover bugs before code is merged with a good DevOps practice in place and they are 90% more likely to test between 91% and 100% of code than in an organization with early stage DevOps.
- 2. Developers aren't always on board:** Nearly half of security pros surveyed (49%) said they struggle to get developers to make remediation of vulnerabilities a priority.
- 3. Bug catching:** Half of security professionals said bugs were most often found by them after code is merged in a test environment.
- 4. Remote, but still secure:** Mostly remote teams are 23% more likely to have mature security practices than primarily office-based teams.
- 5. Does security matter?** Of course, but only 44% report that security vulnerabilities are a performance metric for developers in their organizations.

The vast majority (84%) of those who responded to our survey questions about security were CIOs and CTOs, so their responses may be less likely to reflect those with hands-on responsibility. But despite the challenges inherent in security, this is a group that is generally optimistic about today and the future. Over half of them don't see red tape as a problem that's getting in the way of fixing potential security issues. And 74% believe they're able to innovate.

Security in action

Our survey respondents use a variety of application security methods to identify problems.

Dependency scanning is the most popular at 56%, followed by cloud security (42%), container security (41%), SAST (35%), license compliance (29%) and DAST at 22%. All told, 12% of security teams test between 61-75% of code.

Application security methods

56% *Dependency scanning*



42% *Cloud security*



41% *Container security*



35% *SAST*



29% *License compliance*



22% *DAST*



Like every other step in software development, automation is critical to successful application security testing but it continues to be a challenge to implement for many companies.

How do you automate application security testing within your software development pipeline?

34% *Security testing results are included in the pipeline report used by developers*



33% *CI/CD automatically kicks off SAST scan*



27% *Developers use spell-check-like function for lite scan as they code*



25% *Don't know*



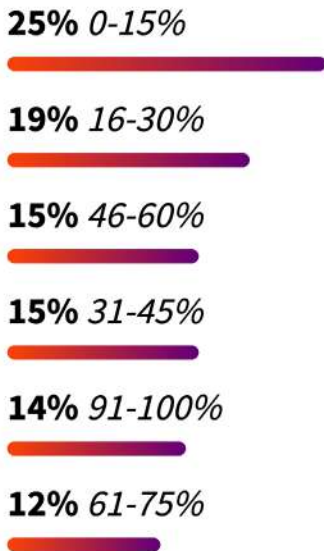
20% *CI/CD automatically kicks off DAST and/or IAST scan*



Just over one-third of companies (34%) said security testing results are included in the developer pipeline report, while 33% said CI automatically kicks off a SAST scan. Other options include developers using a spell-check-like function to scan as they code (27%), an automated DAST or IAST scan via CI/CD (20%), and 12% either do manual or no testing at all. A full 25% of respondents said

they didn't actually know how security testing was automated during development, a percentage that may reflect the large number of CIOs/CTOs answering questions in this area.

Percentage of code that's tested using application security methods

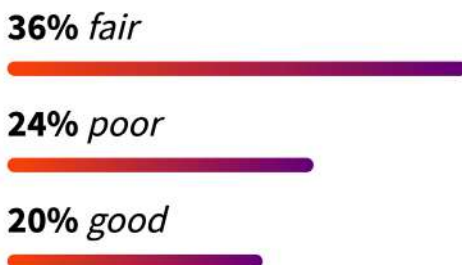


Like developers and operations professionals, security teams also believe testing is what slows down development the most (50% cited this), followed by planning (39%), code review (33%), code development (31%), and deploying to production (26%).

A question of ownership

No part of software development raises more issues about ownership than the subject of security. The idea that “everyone is responsible for security” might be the ideal but it can also be part of the problem as “everyone” can easily turn into “no one.” Security professionals often complain about being on the outside, while developers and operations teams can resent being told how to prioritize their work. Our survey responses indicate developers are taking more responsibility for security, but of course a lot of work remains. In fact, only 20% of those surveyed rated their organization’s security efforts as good, while 36% said they were fair and 24% said they were poor.

How do security professionals rate their security practices?



Half of those surveyed said coders receive and address security feedback during the development process and 44% report that security vulnerabilities are a performance metric for developers in their organizations.

That doesn't mean everything always goes smoothly. In fact, nearly half of those surveyed (49%) said they struggle to get developers to make remediation of vulnerabilities a priority. And that's true even if developers are held accountable – 55% who said security is a developer performance metric reported it is still hard to get coders to focus on fixing vulnerabilities.

Top 4 most important security metrics for tracking the handling of vulnerabilities:

1. Severity of vulnerabilities
2. Time elapsed since identification
3. Mean time to resolution
4. Number of vulnerabilities spotted

Half of security professionals said bugs were most often found by them after code is merged in a test environment. Developers are also much less likely to find bugs later in the lifecycle. Security professionals said just 12% of developers find between 76-100% of late-stage bugs.

The ongoing frustration comes through loud and clear from security professionals:

“It looks like my organization doesn't care about security. They are only interested in having something functional. Investing time in security, which usually is not visible to the user, is a waste of time (and money) for them.”

“We have an established security organisation, but our development teams are not as aware of secure development as they could be.”

“Security is a key point indicator for management, however the development team lacks formation and integration around secure code and secure by design.”

“Security is based on several coding practices, which often are hard to sell to the clients (in terms of velocity), which makes it harder for a small company to win in a competitive field. On the other hand, our ability with security analysis also got us some contracts to fix them and/or extend the client’s projects.”

“We have poor security processes in software development. Our IT department takes security seriously but lags behind in implementing reliable security monitoring for all incoming requirements.”

DevSecOps

The DevOps process can be helpful for security: 34% said their organization’s DevOps practices are good, 36% said they were fair and 13% said they were very good. Security leadership at organizations with strong DevOps models are 23% more likely to get an accurate view of security team performance.

How do security professionals rate their DevOps practices?

36% *fair*



34% *good*



13% *very good*



A mature DevOps model means teams are 3x more likely to discover bugs before code is merged, and they are 90% more likely to test between 91% and 100% of code than in an organization with early-stage DevOps. Not doing DevOps well leaves security teams 2.6x more likely to have to deal with red tape before finding or fixing bugs.

Based on the comments, DevSecOps means things run smoothly:

“I’m obsessive about documentation and streamlined CD.”

“We got project templates in GitLab which we can copy and change as needed. Everything is automated and integrated with our bare-metal Kubernetes cloud. Changes are automatically tested against our

requirements and deployed to staging for feature branches and to production on master. Only thing missing is automatic smoke testing after the deployment and rollback on failure.”

Remote and secure

Like their developer and operations colleagues, security pros who work remotely report they're better able to get the job done. Mostly remote teams are 23% more likely to have mature security practices than primarily office-based teams, and leadership is 57% more likely to see performance clearly. And at least when it comes to red tape, being in the office is a negative: In-office teams are 39% more likely to encounter red tape when they're trying to fix a bug quickly.

Operations

Mission Clarity

For operations teams, having more defined processes and workflows helps keep releases on track. The primary focus for ops teams in 2019 is to bring transparency to processes.

Operations top findings

- 1. Operations teams see the benefits of DevOps:** They're 2.6x more likely to believe they get sufficient notice to support developers when their DevOps practice is very good.
- 2. Working at home = efficiency:** All-remote ops teams are 1.6x more likely to quantify and document their work than in-office teams.
- 3. Security is still a work in progress:** The majority of ops pros (34%) rated their organization's security efforts as fair.
- 4. DevOps is the first choice of methodologies:** 70% practice it, versus 61% using Scrum.
- 5. Take a closer look:** Teams with a well-developed DevOps model are 58% more likely to have good insight into what colleagues on other teams are working on.

In the complex and sometimes underappreciated world of operations, DevOps can bring much needed clarity to processes and collaboration.

Top development methodologies

70% *DevOps*



61% *Scrum*



43% *Kanban*



18% *Other Agile (e.g. Extreme Programming)*



In fact, DevOps is the clear winner from our operations respondents. When DevOps works, it makes a dramatic difference. Ops teams are 1.8x more likely to believe they get sufficient notice to support the developer side when their DevOps practice is very good.

In the smaller, tech-focused companies our survey data highlights, operations pros have very clear priorities to structure their work. What matters most to them is the product roadmap timeline followed by ROI, the current workload of individual developers, and the estimated cost of development. They judge their success most often by looking at the number of incidents, followed by uptime trends over time and mean time to resolution of issues. This is a stark contrast with larger, more “modern” operations teams that have largely moved away from using incident data as a standalone metric, preferring instead to consider incident consolidation ratios or monitoring-initiated incident coverage or customer satisfaction.

The choice of tools

It takes a variety of solutions to manage operations, but ops pros certainly think they’re making good choices – 61% said the tools their engineering organization uses are the best for the job. And ops’ opinions matter: 59% said their recommendations for best practices and tools are usually followed.

Top tools for monitoring

42% *Grafana*



30% *Nagios*



30% *Kibana*



29% *Prometheus*



When it comes to monitoring, 42% of operations team members choose Grafana, 30% use Nagios and Kibana, while 29% reach for Prometheus. Nearly half of them (46%) use AWS as a cloud provider, while 33% use other providers such as DigitalOcean and OVH. Google Cloud Platform was chosen by 27%. Project management tool choices were dominated by Jira (34%) and GitLab (29%), while 11% use other tools including Redmine, ServiceNow, and Azure DevOps. Microsoft Project is the choice of 6.4%, which is the same percentage of operations professionals who reported using no project management tools at all, and 6% use Trello.

Most used CI and build tools

65% *GitLab*



39% *Jenkins*

14% *Don't use CI or build tools*

9% *Travis CI*

For continuous integration and build, 65% say they use GitLab followed by Jenkins at 39% and Travis CI at 9%. A full 14% of respondents said they don't use CI or build tools.

Code deployment frequency

51% *Continuous deployment (on demand, multiple deploys per day)*



35% *Between once per day and once per month*

11% *Between once per month and once every 6 months*

3% *Don't know*

About 75% don't run production applications using Serverless/FaaS architecture, with the majority - 36% - stating that there is no use case to use such architecture.

Ops and DevOps

It's clear our survey respondents see their departments and their success with DevOps as a work in progress. Like their coding counterparts, operations team members point to testing as the factor most likely to slow down developments. Other areas that can get in the way include planning (34%), code development (30%), deploying to production (29%), and code review (24%).

Operations teams are happier with their organizations' DevOps practices than developers are – 34% rated their experience as good (compared to just 28% of developers). But a full third of both ops and devs said DevOps was just “fair” in their companies, and 16% of operations team members rated it “poor.”

How do operations professionals rate their DevOps practices?

33% *fair*



34% *good*



16% *poor*



DevOps continues to be a process teams have to work at and that's made obvious by the comments below:

“DevOps is in its infancy. This is a topic that I am working on to elaborate. This I do by trying to demo DevOps with concrete examples. Still there is no time and there is no knowledge in higher management. This impedes budget and time for this.”

“We have good DevOps momentum, but the processes are not repeated across organizations and sometimes even between nearby teams.”

“We are planning our DevOps practices, mostly training our team and spreading the word about the benefits of DevOps culture.”

“The tools and processes are there. [The] main issue is getting developers to use the tools and understand what happens under the hood so that they're able to debug their own deployments.”

“My specific area is attempting to bring a cultural change to the organization but there is an incredible resistance to any change at all.”

Teams with a well-developed DevOps model are 58% more likely to have good insight into what colleagues on other teams are working on when compared to their counterparts who encounter immature DevOps practices. Furthermore, in the development lifecycle, organizations are 2.5x more likely to encounter the most delays during the planning stage if their DevOps model is very poor.

Staying secure

Like developers, operations professionals have mixed views on their organizations' security efforts.

How do operations professionals rate their security practices?

34% *fair*



29% *good*



20% *poor*



The majority – 34 percent – rated security as fair, while 29 percent said good and 20 percent called it poor. Based on the comments, standardization and documentation would probably help move the process along.

“We are about to document all measures for the security implementations. We’re missing automated testing that might come after the documentation phase - until then this will be ad hoc.”

“We are now in the stage where we just started the documentation of security processes in our organization.”

“[There is a] lack of automation/standardization in server installation and management, [and a] lack of centralized authentication across all our tools (although it also could be a strength: each system being an isolated autonomous island).”

Why remote works for ops

Like their developer counterparts, operations pros generally find remote work suits them and it makes them more efficient. All-remote teams are 1.6x more likely to quantify and document their work than in-office teams. And mostly remote workers are 17% more likely to have insight into what other teams are working on. All-remote operations professionals are also 2.6x more likely to be given sufficient notice to support developers compared to their in-office peers.

Trajectory

Not surprisingly, all of our survey respondents reported ambitious plans for 2019. Almost two-thirds want to invest in infrastructure to support continuous integration, deployment, and delivery. About half hope to improve automation, while 44% will increase use of containers and 43% will double down on DevOps. And just over one-third plan to expand their use of the cloud.

Developers and security pros hope to invest more in continuous integration, deployment, and delivery as well as amping up automation and container use. Operations teams are on the CI/CD and automation bandwagons as well, but they're also looking to deepen their commitment to DevOps.

Debrief

Over 4,000 developers, operations, and security professionals have spoken and their message about the importance of continuous improvement in DevOps should resonate no matter the stage of your company's journey. "Continuous" is a word that's used frequently when talking about DevOps and with good reason: Successful organizations continually iterate. It's not easy or straightforward but the changes DevOps will bring to software development – performance, visibility, collaboration, and innovation – make it worth it.

Start where you are, share this survey, and have the conversations. It really is as easy as an ops team member shared in our survey: "We are planning our DevOps practices, training our team, and spreading the word about the benefits of DevOps culture."



GitLab