DZone

DZONE'S 2019 GUIDE TO

DevOps

RESEARCH PARTNER SPOTLIGHT

WhiteSource

Key Research Findings

BY JORDAN BAKER
CONTENT COORDINATOR, DEVADA

Demographics

For this year's DZone Guide to DevOps, we surveyed developers, architects, and technologists to get their thoughts on all things DevOps. We received 983 total responses, with a 54% completion rating. Based on these numbers, we have calculated the margin of error for this report to be 3%. Below are some basic demographics of our respondents.

- The average respondent has 18 years' worth of experience in IT.
- 36% work for organizations headquartered in the USA and 36% for companies headquartered in Europe.
- 34% of respondents live in Europe, 27% live in the USA, and 11% live in South-Central Asia.
- 25% work for organizations sized 1,000-9,999 employees, 24% work for organizations sized 100-999, and 19% work for organizations sized 10,000+.
- 23% work for software vendors, 16% work in finance/banking, and 7% work in e-commerce.
- 29% work as developers, 24% as architects, and 19% as developer team leads.
- 86% develop web applications/services, 43% develop enterprise business applications, 33% are modernizing legacy applications, and 28% develop native mobile apps.
- 83% work in the Java ecosystem, 70% work in the client-side JavaScript ecosystem, 36% use the Node.js ecosystem, 34% develop with the Python ecosystem, and 28% use the C# ecosystem.

The Architecture of a DevOps Team

Despite the prominence of DevOps in the software industry, less than half of respondents (46%) told us that their organization has an officially designated DevOps team. This lack of an official DevOps team in many organizations manifests itself in the unequal distribution of code deployments. Whereas in DevOps, development and operations teams are meant to work cooperatively to create and release code, 57% of respondents reported that only development teams perform code deployments in their organization. 42% reported this as an operations function and 32% told us that release engineers handle code deployments. Despite these trends, 54% reported management as a DevOps enabler.

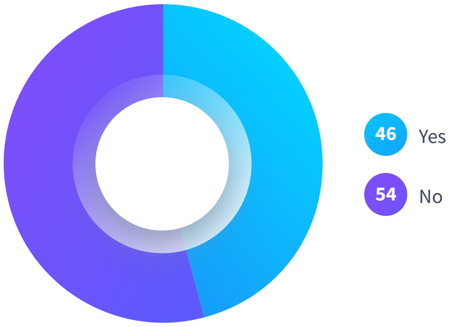
What does a modern DevOps team look like? According to respondents, there are four major aims of a development team when adopting DevOps practices: introducing automation across the SDLC (67%), helping the organization adopt the best CD tools (65%), increasing collaboration and breaking down silos between dev and ops (62%), and developing and delivering software across the entire stack (53%). The purpose behind teams adopting DevOps, thus, has two main considerations: improving software and improving culture. Let's first focus on the ways teams use DevOps to improve the software they produce.

To begin, the majority of respondents' organizations (57%) still rely on the development side of their DevOps pipeline to deploy code to production. This is up from 54% in our 2018 DevOps survey. 42% of respondents' organizations use operations to deploy their code, up from 39% in 2018. As the practice of DevOps continues to grow in influence, one would expect the percentage gap of deploys performed by development and operations to narrow; yet, in each of the last two years, development has performed 15% more deploys than operations among our respondents.

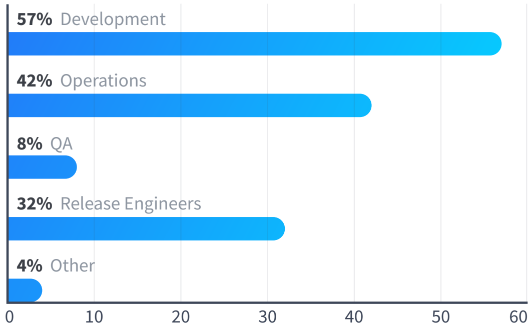
When it comes to the actual software delivery process, survey-takers reported five major processes: code quality checks (62%); breaking

SURVEY RESPONSES

Do you have an officially designated DevOps team in your organization?



Which division of your organization normally performs code deployments to production?



up the build into stages (62%); code reviews (58%); manual checks to proceed (52%); and code coverage checks (50%). Several of these delivery processes saw significant year-over-year growth when compared to our 2018 DevOps survey. The adoption rate of code quality checks among respondents' teams rose by 6%, as did breaking up the build into stages. Code coverage checks, however, witnessed the largest year-over-year growth, increasing by 7% among survey-takers (came in at 43% in 2018).

Given that DevOps processes are meant to increase the agility and efficiency of software development and operations teams, and thus the quality of software, one would expect a certain amount of developer autonomy to come with the adoption of DevOps. When we asked what tasks development teams have autonomy to do with limited or no manual approval from others, 76% of respondents told us making code changes. This proved the most popular answer by far. Respondents also reported a fair amount of autonomy when deploying testing environments (55%). While no other answer received more than 50% of responses, some other interesting responses included creating internal resources (41%), contributing improvements to tooling provided by other teams (35%), and deploying code changes to production (26%).

Now that we've seen how DevOps is used to improve development, let's look into a few ways it's being used to improve the culture around development. As noted earlier, 62% of respondents told us that one goal of their DevOps team is to increase collaboration and break down silos between dev and ops. This goal saw significant year-over-year growth. In our 2018 DevOps survey, 55% of respondents reported the breakdown of silos as a main goal. We also mentioned earlier that 54% of respondents view management as an enabler of DevOps principals. This, t` oo, grew from last year, when less than half (48%) of respondents reported thusly.

It seems that DevOps is helping break down some barriers and also has increasing buy-in from management. We see these two trends greatly influencing the ways in which teams share DevOps best practices and lessons learned within their organization. 36% of respondents reported they share their DevOps findings among individuals within the immediate team, and 29% share this information across teams within the organization.

Developing in DevOps

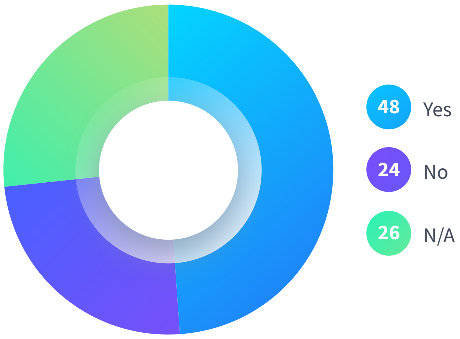
DevOps and microservices are a natural pair, given their impetus on continuous, component-based development rather than a monolithic-based approach. As such, we have seen a growth in the percentage of respondents using a microservices-based architecture for their applications in production environments. In our 2018 survey, 29% of respondents claimed to use microservices in production; in this year's survey, the adoption rate of microservices rose to 39%. We also saw a decrease in the number of respondents not using microservices. In 2018, 31% told us they were not using microservices but were considering them, and another 15% said they were not considering microservices at all. This year, these numbers fell to 27% and 13% respectively. Interestingly, when we compare this data to the two most popular types of developers reported in our Demographics section (web app and business enterprise app), we find web app developers are more likely to use microservices. 42% of respondents developing web applications reported using microservices in production, whereas 38% of business enterprise developers use microservices in production. Despite this slight disparity in microservices adoption in production environments, web app (19%) and business app (18%) developers proved almost equally likely to adopt microservices in development. If you'd like to explore microservices more, check out the 2018 DZone Guide to Microservices.

In order to maintain and commit changes to code, respondents reported two main tools: GitHub and GitLab. Interestingly, despite the enormous popularity of GitHub in the software industry, the percentage of respondents using GitHub fell slightly from 37% in 2018 to 34% for this year's survey. GitLab, however, saw a nice jump in adoption rates, rising from 23% in 2018 to 33% this year. When it comes to the tools used to deploy code, 64% report using the same deployment tool across all phases of the SDLC (development, QA, and production). This rose slightly from last year, when 61% of respondents reported using the same deployment tool across all phases of the SDLC.

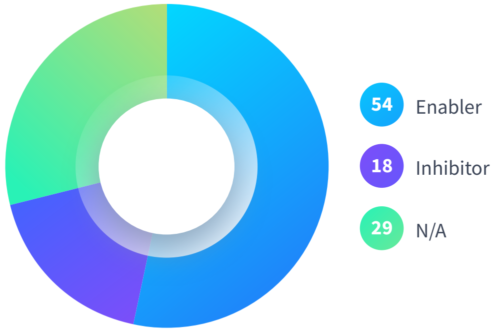
No matter the tools used, almost half (47%) reported their main project's build execution time to be less than 10 minutes. On a more granular level,

SURVEY RESPONSES

Is your Ops team involved in the design of your CD pipeline?



Is management an enabler or inhibitor of DevOps principles at your company?



25% reported a build execution time of two to five minutes, and 22% reported a build execution time of six to 10 minutes. When we compare these numbers to our stats on web app and business enterprise app developers, we find that 28% of web application developers report a slightly faster build execution time. 28% of web app developers told us they have a build execution time of two to five minutes, whereas 23% of enterprise business app developers claimed a two- to five-minute build execution time.

When things go wrong, however, there seems to be two main causes. 75% of respondents told us that application errors/issues were the leading cause for rollbacks and/or hotfixes in their organization, and 52% reported environmental errors or issues. The mean number of rollbacks required among our general survey, however, proved rather low. Averaging out all the responses received, we find that 15% of deployments need rollbacks or hotfixes. And, continuing this positive trend, we find that the typical time to restore service when something goes wrong in production has gone down since last year. In 2018, 24% of respondents reported a time window of less than one hour, and this year, 28% of respondents reported the same time frame.

CI and CD

CI/CD is at the core of DevOps, and both CI and CD constitute two of the most important processes in any DevOps pipeline. When we asked respondents if they believe their organization has achieved continuous integration, 31% said yes, 33% told us on some projects, and 37% said no. When we asked the same question about continuous delivery, only 14% of respondents said yes (down from 19% in 2018), 28% said on some projects, and 58% said no (up from 50% in 2018). Based on these numbers, it appears that approximately two-thirds of respondents feel their organization has achieved a level of comfort with CI processes, but less than half can say the same for CD.

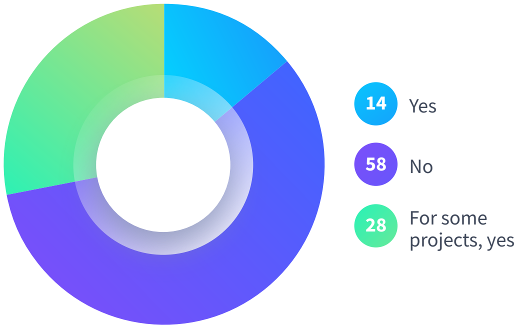
This disproportionate comfort level between CI and CD came through when we asked survey-takers if their CI processes extend into an automated CD pipeline. Of those surveyed, 58% said no. When looking into the pain points of the continuous delivery pipeline that could be behind this disconnect, we find that 51% of respondents report environment configuration setup to be an issue and 30% report user acceptance testing as a problem (up from 25% in 2019). While these were the two most oft-reported issues, automated testing (28%), coordination of team members and resources (27%), and deployment process in use (24%) also came in as popular answers. It's interesting to note that all of the issues listed, outside of automated testing, saw a significant percentage swing over 2018's responses. The number of respondents reporting environment configuration and set-up fell by 4%, the percentage of those reporting coordination of team members and resources fell by 6%, and those reporting deployment processes fell by 6%. While those are all positive signs, respondents who claimed user acceptance testing to be an issue rose by 5%.

There seems to be a few reasons why so many organizations are struggling with the adoption of continuous delivery. Much like we saw with our discussion of obstacles to DevOps adoption, the main barriers to adopting continuous delivery are more cultural than technological. The main hurdle reported by respondents was corporate culture (47%), specifically a lack of collaboration and/or DevOps practices within an organization. In fact, of the four largest barriers to CD adoption given by respondents, corporate culture was the only one that exhibited a year-over-year increase, with 45% of our 2018 survey respondents telling us culture was an issue. Due to these obstacles, team- and organization-wide adoption of CD practices tends to take anywhere from two months to a year. 26% of respondents told us their team's adoption of CD processes took two to six months to complete, and 24% said it took their organization six to 12 months to fully adopt continuous delivery. Compounding these issues, nearly half of survey-takers (42%) said their team/organization does not use any metrics to track their CI/CD processes.

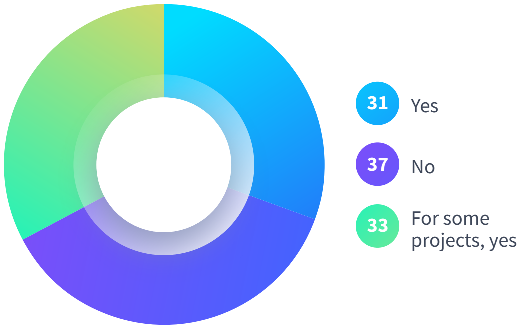
It should be noted, however, that it's not all gloom-and-doom on the CD side of things. The knowledge gap and distance between dev and ops in the CD pipeline are both shrinking. 48% of respondents reported that their ops team is involved in the design of their CD pipeline, up from 43% in 2018. And while 45% of our 2018 respondents said that engineers/ops did not have the right skillsets for working in a CD pipeline, only 36% espoused this belief in this year's survey.

SURVEY RESPONSES

Do you believe your organization has achieved continuous integration?



Do you have an officially designated DevOps team in your organization?



Diving Deeper Into DevOps

Twitter

@RealGeneKim

@sKriemhild

@bridgetkromhout

@jezhumble

@DonovanBrown

@lisacrispin

@editingemily

@testobsessed

@proudboffin

@HelenRanger4

Podcasts

Arrested DevOps
Get the knowledge to help you achieve understanding, develop good practices, and operate your team and organization for maximum DevOps awesomeness.

The New Stack Makers
Through featured speakers and interviews, learn about new software stacks that are changing the way we develop and deploy software.

DevOps Radio
Learn about effectively achieving software delivery through DevOps.

Zones

DevOps dzone.com/devops
DevOps is a cultural movement supported by exciting new tools that is aimed at encouraging close cooperation within cross-disciplinary teams of developers, and IT operations, and system admins. The DevOps Zone is your hot spot for news and resources about continuous delivery, Puppet, Chef, Jenkins, and more.

Agile dzone.com/agile
In the software development world, Agile methodology has overthrown older styles of workflow in almost every sector. Although there are a wide variety of interpretations and techniques, the core principles of the Agile Manifesto can help any organization in any industry improve their productivity and success.

Open Source dzone.com/opensource
The Open Source Zone offers practical advice regarding transitioning from a closed to an open project, creating an enforceable code of conduct, and making your first OSS contributions. This Zone encourages you to adopt an open-source mentality and shape the way open collaboration works.

Refcardz

Continuous Delivery
Minimize the time it takes to go from idea to usable software. Learn to use agile techniques and automate the entire software delivery system: build, deploy, test, release.

Compliant DevOps
With new data protection laws coming into play, and consumers more aware than ever before of how their privacy is being compromised, there is now a requirement for companies to adopt a compliant DevOps approach.

Shifting Left With Continuous Delivery
Learn how shifting left with CD makes software development faster and more reliable, why whole teams need to cooperate in effective left shifts, and how containers are important in shifting left with continuous delivery.

Courses

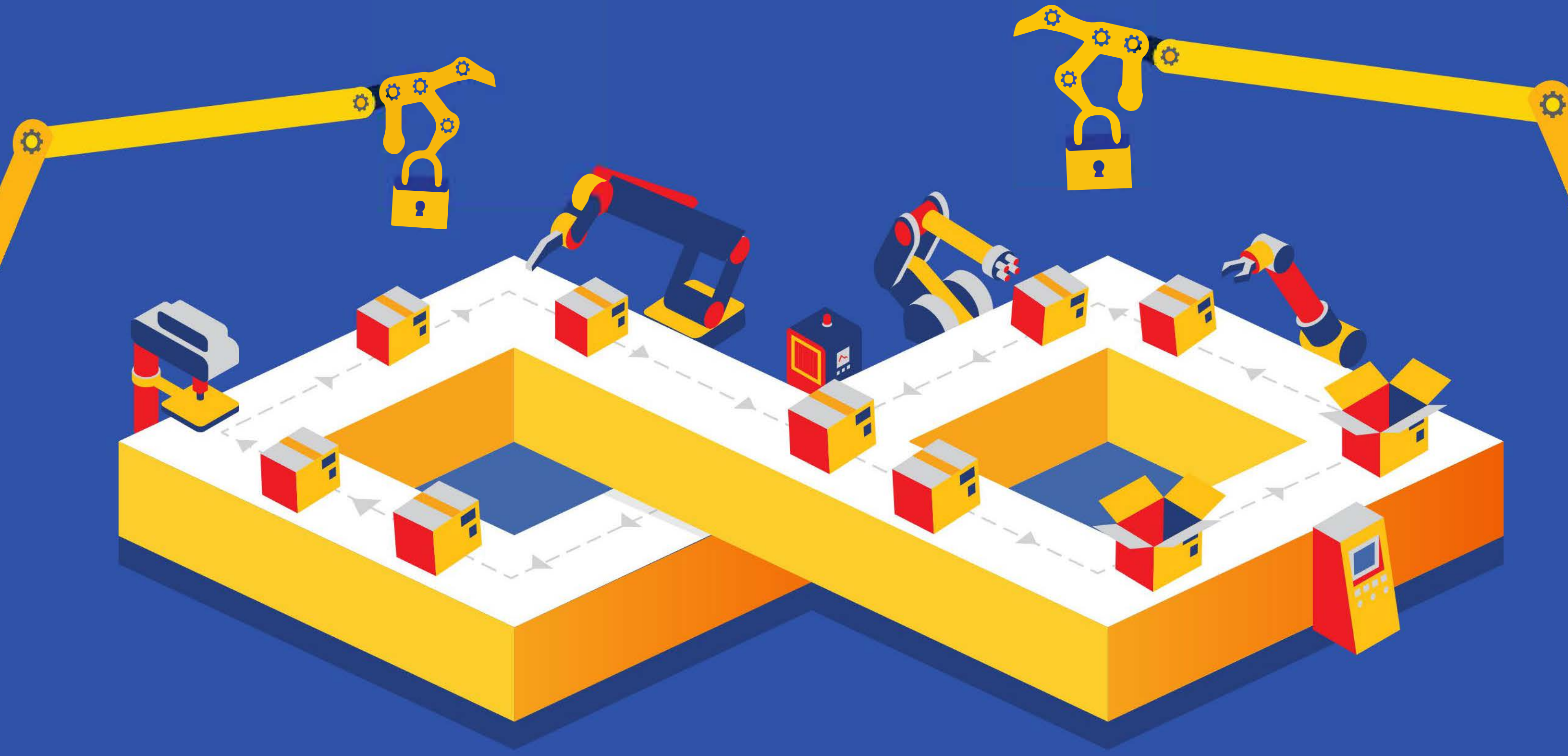
Master Jenkins CI for DevOps and Developers
Take your DevOps skills to the next level by learning how to build automated CI pipelines with Jenkins.

Learn DevOps: The Complete Kubernetes Course
Learn how to build, deploy, use, and maintain Kubernetes to run and manage your containerized apps.

DevOps for Developers: How to Get Started
Get a comprehensive definition of DevOps, an understanding of why you should do DevOps, and insight into how to get started with DevOps.

WhiteSource

Automate Security Into Your DevOps Pipeline



FIND OUT HOW

www.WhiteSourceSoftware.com

Automating Open Source Security In Your DevOps Pipeline

You're in a race against the clock. Time is your most valuable resource as you try to keep up with the rapid pace of development. It's all about innovation, and meeting release schedules. You can't afford to fall behind. You're about to make it to the finish line when you see a security flaw lurking right there. You're brought to an unnecessary halt.

According to WhiteSource's 2018 Annual Report, nearly all developers rely on open source components on a daily basis to keep up with tight deadlines without sacrificing on quality. The problem? Many organizations still treat open source security as an afterthought, despite the significant rise in open source security vulnerabilities over the past few years. This mistake could end up hurting your agility, flexibility, and — ultimately — your organization's products and customers.

"Security slows the process down" you may be thinking right now. But contrary to belief, this doesn't need to be the case. Open source vulnerabilities can be revealed from the get-go, by choosing the right automated tools and integrating automated open source security checkpoints in different phases of the DevOps pipeline. Developers can remediate security flaws in open source components early in the pipeline, saving time and most importantly, boosting agility.

Bottom line? Don't treat security as an afterthought. By continuously automating open source security measures throughout the development lifecycle, development and DevOps teams will be able to power through to meet tight release schedules, allowing all teams to make it to the finish line effortlessly.

WRITTEN BY RAMI SASS
CEO & CO-FOUNDER, WHITESOURCE

PARTNER SPOTLIGHT

WhiteSource

Automates and secures open source components used in your code by fully integrating into your software development lifecycle.

Category Application Security | Software Composition Analysis | DevOps | Secure Coding

Release Schedule Continuous

Open Source? Yes

Case Study Due to the sensitive regulatory environment Siemens Healthineers works in, Siemens H. needed assurance that they are remaining compliant and secure in their use of open source components.

"With open source software, usually the source code is available for all to see, including hackers," explained Code Clinic Lead, Neil Langmead. In order to avoid costly mistakes that can result from vulnerable or risky open source components being added to their products, Siemens Healthineers turned to WhiteSource.

"We chose WhiteSource because of its ease of use, its excellent data, and for the in-depth security vulnerability information that comes with the reporting engine." With WhiteSource, Siemens was offered the widest coverage of plugins and languages that they needed, as well as the continuous monitoring and policy enforcement safeguards they required in order to allow their team to code with confidence. Read more here

Strengths

1. Largest Vulnerabilities Database: Continuously aggregates information from multiple sources
2. Comprehensive Coverage: Supports 200+ programming languages and 20+ environments (incl. containers)
3. Pinpoint Accuracy: Proprietary algorithms guarantee no false positives
4. Automated Workflow: Enforce policies automatically at all stages of the SDLC
5. Easy Remediation: Pinpoints vulnerable methods affecting your products

Notable Customers

- Microsoft
- Comcast
- EllieMae
- IGT
- Spotify

Website
whitesourcesoftware.com

Twitter
@WhiteSourceSoft

Blog
resources.whitesourcesoftware.com