

WHITE PAPER

HOW DEVOPS IS REVOLUTIONIZING IT

An agile approach to software development and operations can drive innovation and speed up the delivery of new capabilities.



EXECUTIVE SUMMARY

Across industries, IT shops are adopting DevOps methodologies to improve app quality, decrease time to deployment and optimize organizational efficiency. But DevOps requires more than just new tools and processes. A successful initiative will emphasize not only the tools and skills needed to implement DevOps but also the culture change that necessarily comes with the upending of traditional development practices.

It is common for organizations to struggle with challenges including unrealistic expectations, insufficient collaboration between teams and unclear business outcomes. By overcoming these hurdles, organizations can implement more

agile processes that lead to real-time feedback and constant iteration. A number of tools can help streamline DevOps initiatives. One of the most crucial is containerization, a technology that allows IT teams to develop and maintain applications using standard units of software. Containers can be created and replicated rapidly, giving organizations a tool to accelerate their development practices. Also, containers are extremely portable among multiple computing environments, a benefit that allows organizations to create cloud-native applications.

An effective DevOps initiative requires a blend of solutions, strategies and services. Many organizations turn to a trusted third-party partner to guide their efforts and avoid mistakes.

DevOps Is Changing IT

It's no mystery why DevOps has swept through IT departments in virtually every industry around the world in recent years. According to Gartner, the approach "represents a change in IT culture, focusing on rapid IT service delivery through the adoption of agile, lean practices in the context of a system-oriented approach." That's a lot to unpack, but at its essence, DevOps is a key component in the move away from more cumbersome traditional development practices, which frequently lead to waste, delays and confusion, and toward more agile workflows that emphasize speed to market and continuous iteration.

To be successful, a DevOps approach requires more than a desire to break out of less effective development practices. An organization must also invest in its people, embrace culture changes and adopt the appropriate set of tools. In particular, containers have been instrumental in helping organizations create the environment necessary to achieve DevOps success.

What Is DevOps?

DevOps is a methodology that enables fast, frequent delivery of value to end users by decreasing the time it takes to produce and provide features and services they need or enjoy. This can be accomplished by breaking down historical barriers between development and operations departments, enabling a frictionless experience for both. In a DevOps environment, operations and development engineers participate and collaborate with one another throughout the entire service lifecycle — from design to development to production support. While the idea of breaking down organizational silos is not new, DevOps in particular has taken off in part because of its alignment with lean and agile workflows.

In the past, developers were typically seen exclusively as the makers or creators of an application, while operations engineers handled applications exclusively after they were deployed throughout an organization. Stated so plainly, it is easy to see why this approach could lead to inefficiencies. It's also easy to see how both groups of IT professionals (and, by extension, an

entire organization) can benefit from improved collaboration. Along with a more free-flowing exchange of ideas, one of the primary benefits of DevOps is the increased ability to iterate as new information comes in, with teams continually tweaking apps in response to user behavior. This ultimately leads to workflows that create the greatest possible value for an application's end users. At the center of DevOps is a unified, relentless pursuit of delighting customers with a great experience.

DevOps has sometimes been described as an extension or outgrowth of agile software development. This puts DevOps in contrast with traditional development methodologies such as waterfall development, a linear project management approach that creates a sequential plan for meeting project requirements. Agile prescribes the core concepts of close collaboration among product managers, developers, customers and quality assurance. DevOps goes a step further by recognizing that both service delivery and the way an application and systems interact with one another should be top concerns throughout application development and operations because of the outsized impact these factors have on the eventual product.

While DevOps was seen as cutting-edge only a few years ago, it is now solidly part of the IT mainstream — a proven practice that creates a significant advantage for organizations that implement it effectively. In particular, organizations are using DevOps practices to help them achieve benefits such as automation, detailed service measurement, orchestration, self-healing, configuration management and security.

According to [CITE Research](#), 99 percent of developers and IT decision-makers say that DevOps has had a positive impact on their organizations. Sixty-one percent say it has helped them produce higher-quality deliverables, 49 percent say it has improved their deployment frequency and 46 percent say it has resulted in better products and services.

What Are Containers?

Containers are standard units of software that allow applications to be moved easily between computing environments. They are lightweight and standalone yet include everything needed to

run an application, such as code, runtime, system tools, system libraries and settings. Because of this flexibility, containers have become central to many organizations' DevOps strategies. When used in a DevOps environment, containerization can greatly accelerate the development process, since containers can be created, replicated or destroyed in seconds. Rather than waiting for an IT operations team member to spin up a virtual machine, developers can rapidly create containers on their own and simply eliminate them when they are no longer needed. By writing, testing and deploying applications all inside containers, organizations create a consistent environment that makes it easier for different teams to collaborate.

Additionally, the portability of containers allows organizations to seamlessly take advantage of both public and private cloud resources during development and operations. Because containers are lightweight, they do not present the same compatibility concerns as other testing and development resources. Another key benefit: Containers simplify the rapid updates that are central to DevOps success. When an application is distributed across multiple microservices that are each hosted in their own containers, development teams can update one part of the app and restart the corresponding container without interrupting the rest of the app.

While containers allow organizations to work on innovative projects, they require cultural shifts, the establishment of

best practices and investments in staff training to truly create value. And one-time training isn't enough. As organizations adopt containers and implement DevOps practices, they need to continue to evolve and allow their employees to adapt to changing conditions over time. Container platforms are also becoming more packaged and ruggedized to ease their support and maintenance.

Strategies for DevOps Success

In the few years that DevOps has been a relatively mainstream IT practice, something of an old joke has already sprouted up. "A customer calls up a tech vendor to make a purchase — Order quantity: One DevOps." The implication, of course, is that IT and business leaders of ten wish they could implement DevOps in their organizations simply by swiping a credit card.

While DevOps implementation is far more complex than that, there are a number of tools, practices and strategies that can help organizations set themselves up for success.

Continuous Integration and Continuous Delivery

CI/CD describes the frequent delivery of apps to customers by integrating development teams with day-to-day data center operations. This is the fundamental concept that most distinctly sets DevOps apart from other development and deployment models. It requires organizations to break down silos so teams can work together to deploy code as quickly as possible — with minimal outages or within the error budget. Applications that cannot have any outages are expensive to operate and difficult to evolve. DevOps strategies aim to set acceptable volumes of small outages where users do not perceive the issue. Canary testing is an example in which an organization deploys a new application to a very small selection of users and confirms the operating behavior. If it performs as expected, the organization can choose to deploy it. This also implies that impacts can be effectively measured too.

While the "CD" in CI/CD stands for continuous delivery, organizations must also emphasize continuous deployment. The primary difference is a manual approval and promotion process between all the CI steps and deployment into a production environment. This usually comes down to teams trusting automation tools and their ability to roll back changes if something breaks.

Application Modernization

Closely associated with the use of containers, app modernization is the process of modernizing the platform infrastructure, internal architecture and other features of legacy applications. Legacy applications can still have an agile development process. The emphasis on application modernization is breaking up large monolithic applications into small decoupled microservices. Containers have become popular largely because they support this architectural pattern. This does add some complexity — for example, instead of having 100 virtual machines, you may have thousands of containers, each with a different piece of the application. Teams can be organized around these services, and development can happen independently. Teams should be able

Top DevOps Challenges

According to [Gartner](#), 75 percent of DevOps initiatives through 2022 will fail to meet expectations. The firm blames five common challenges for the gap:

Lack of attention to the customer value proposition: Clear goals that are tied to customer value will ultimately generate the right buy-in. Rapid experimentation can be a key enabler to get to user acceptance.

Insufficient change management: Rather than focusing their efforts exclusively on DevOps tools, organizations should carefully consider team culture and make a concerted effort to get staff on board with upcoming changes.

Lack of collaboration: DevOps efforts cannot be limited to infrastructure and operations teams. Instead, organizations should break down barriers between teams and coordinate efforts across silos.

Too much, too soon: Gartner warns that launching DevOps in a single step — a "big-bang approach" — comes with a high risk of failure. Instead, organizations should implement DevOps incrementally and iteratively.

Unrealistic expectations: Like other developments in the technology landscape, DevOps is sometimes seen as a cure-all. It's not. Instead of expecting a DevOps initiative to produce transformative results overnight, organizations should agree on objectives and metrics and then track success over time.

to own and update their services without relying on another team as a dependency.

Containers allow organizations to virtualize their applications, breaking them into individual pieces that can be delivered and patched faster than ever before. According to [Forrester](#), application modernization is associated with a number of benefits, including greater operational efficiency, improved developer productivity and reduced costs. Also, Forrester notes, organizations that successfully modernize their applications reduce the likelihood that additional staff will be needed to support applications in the future.

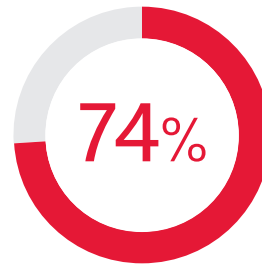
Infrastructure as Code

This is a foundational pillar of DevOps. Without Infrastructure as Code, teams must maintain the settings of individual deployment environments, which evolve over time into “snowflakes” – configurations so unique that they cannot be replicated.

IaC can help organizations eliminate this problem by allowing them to manage their IT infrastructure using configuration files. Since infrastructure configuration takes the form of simple code in IaC, it is easy for IT teams to edit, copy and distribute as they would with any other source code file. This stands in contrast to the more cumbersome (and more costly) practices associated with managing physical infrastructure. Even cloud infrastructure, with its high availability and scalability, cannot match IaC in terms of consistency. IaC also enables quick setup, but the top benefits are likely consistency and accountability.

Security

A core DevOps concept is “shifting left” on security. Shifting left refers to the timeline of the software development lifecycle.



The percentage of organizations that had adopted a DevOps approach within their IT environments by 2020 – up from 66 percent a year earlier¹

Moving from left to right, your process has planning, building, testing, deployment, and ongoing operations and maintenance. Traditional security is an operations and maintenance concern that typically results in security being bolted on instead of being integrated into the full lifecycle of an application.

Security controls, policy and governance should be considered at every phase of the software development lifecycle, hence shifting from right to left in the timeline. Security teams should be consulted during planning phases. Tools can be embedded directly into development environments to give instant feedback. Many types of scanning and

testing can be performed on an application and where code is stored, and security teams should put policies and governance in place, as well as the tooling necessary for visibility and controls in a cloud-native environment.

Cloud-Native Approach

A cloud-native approach to DevOps can deliver instant business insights and help organizations control costs by making it easier for IT leaders to monitor consumption and usage rates. But the best cloud tools also allow organizations to become infrastructure-agnostic, which increases flexibility and agility. Also, cloud-native applications allow organizations to control application lifecycles from inside the application rather than relying on external processes and monitors.

Technologies and Services That Support DevOps

While DevOps is primarily a set of practices (as opposed to a technology solution that can be purchased off the shelf), organizations will want to standardize on common processes and tools to drive the culture change needed for success. It is common for internal stakeholders to resist these changes. IT and business leaders must communicate effectively and transparently to show employees the value of adopting a new way of doing things.

While new technology tools are necessary to enable effective DevOps initiatives, they are not sufficient on their own. Often, organizations find they must rely on external partners for services that help them deploy and manage their new DevOps environments. Organizations also should be prepared to revisit past decisions. A tool or partner that is beneficial today might need to be changed later.

Solutions

Container management platforms: While containerization offers significant benefits, the approach does require management systems so that environments can continue to work effectively even after the number of containers in an organization becomes too large for a single IT team to handle. In fact, some of the advantages of containers (including the ease

What Is DevOps as a Service?

Some organizations that lack internal DevOps expertise opt for a DevOps as a Service model. In this approach, an external provider supplies an organization with the tools needed to enable DevOps and connects these tools together so they work as a cohesive unit. The model stands in contrast to a more typical best-of-breed approach, in which a DevOps team uses a collection of disconnected in-house tools.

This approach is a good fit for organizations that aren't able or inclined to hire professionals who are skilled in DevOps or to train existing employees. The model allows individuals and teams to use intuitive interfaces for DevOps processes without having to understand how an entire chain of disparate tools works. That helps improve collaboration, monitoring, management and reporting.

and speed with which they can be created or destroyed) are the same things that sometimes make them difficult to manage. While a single container can be easily manipulated by any user, the ability to spin up thousands or even millions of containers in very little time can lead to management challenges for obvious reasons. An effective container management platform can ease setup, simplify administration, provide continuous health checks and automate processes from load balancing to orchestration.

To help organizations solve their management challenges, Google created Kubernetes, an open-source platform for managing containerized workloads and services. Other open-source tools such as Docker also provide container management, and some vendors have begun creating their own management platforms (for example, VMware's Tanzu) to deliver the capabilities that organizations need.

Application modernization tools: Although containers are central to many application modernization efforts, organizations often require additional application modernization tools. Dedicated app modernization solutions can help to speed



The percentage of organizations that cite the need to modernize their application delivery environments as a top technology bottleneck²

Kubernetes enterprise adoption, simplify cloud migration and provide centralized management for clouds and apps. Data management tools are another important component of application modernization. According to IDC, data is the "glue" for application modernization efforts — a benchmark for measurement. "Data can help gain invaluable insights into the bottlenecks in application delivery; identify redundancies, infrastructure challenges, and security and compliance issues; or help developers roll back an update if necessary," IDC notes in a [2020 report](#). "It also informs how applications are used to improve customer experiences and application features."

Security solutions: To secure their DevOps processes, organizations must embed governance and cybersecurity practices in their efforts. This means investing in tools and processes such as identity and access management, privilege management, firewalling/unified threat management, code scanning and review, software composition analysis, configuration management and vulnerability management, just to name a few. Some vendors are even coming to market with tools specifically designed to improve security around DevOps, applying a consistent set of security checks throughout the build-to-release process.

Automated testing tools: Testing solutions help organizations with configuration, integration and delivery management, ultimately saving time for IT shops by automating the testing process.

Monitoring: Different monitoring tools can help organizations better manage their DevOps environments. Resource monitoring solutions gather data on how servers are running; network monitoring tools examine data coming in and out of the network; and application performance monitoring tools collect data on how a given service is performing.

Observability tools: Observability is a core site reliability engineering concept that changes the focus from creating thresholds and alerting (which can be somewhat noisy) to data points that give real context into application and service health and degradation of service performance, as well as telemetry into why issues may be occurring. Observability tools help cut through the noise and focus on what is important to end users: reliable services. Observability helps do this more proactively rather than reactively. It requires full-stack visibility and the ability to correlate data to provide quick insights into underlying causes.

Services

Advisory services: A third-party partner can help an organization see its current environment with fresh eyes to pinpoint the greatest areas of opportunity and the most pressing challenges that might hinder a DevOps initiative. When an organization lacks internal staffers with significant DevOps expertise, advisory services can bring in experts who have

Skills for DevOps Success

The capabilities necessary to ensure a successful implementation of DevOps go far beyond application development. Yet it is relatively rare for companies to offer formal DevOps upskilling programs.

- **45 percent** of IT and business leaders say that application development and design is a "must-have" DevOps skill, but even more say that IT operations knowledge (**52 percent**) and security practices (**52 percent**) are critical skills.
- Other technical skills cited as must-haves include knowledge on topics such as quality assurance (**32 percent**), disaster recovery (**31 percent**), testing (**31 percent**) and networking (**28 percent**).
- "Soft" or "human" skills are also seen as highly important to DevOps success; **88 percent** of respondents call collaboration and cooperation a must-have skill, for example. Further, **81 percent** cite interpersonal skills, **79 percent** cite problem-solving and **70 percent** cite flexibility and adaptability.
- Other important soft skills include leadership (**48 percent**), product thinking (**47 percent**) and risk-taking (**37 percent**).
- Only **31 percent** of companies have formal upskilling programs for employees engaged in DevOps.

Source: DevOps Institute, "[Upskilling 2020: Enterprise DevOps Skills Report](#)," March 2020

worked with dozens of organizations and have seen the real-world results of DevOps and containerization.

Implementation: Similarly, a partner can guide an organization through the implementation of its DevOps plans. This often includes not only the implementation of new tools but also help with the change management component of a DevOps initiative — frequently the most challenging part of such a big shift. Here again, an external set of eyes is essential to help leaders determine whether internal stakeholders are following best practices and embracing the processes that will lead to improved products and a nimbler IT ecosystem.

Security services: Engagements such as vulnerability assessments can give business and IT leaders a realistic view of how effective their current cybersecurity tools and practices are — and can lead to specific recommendations for improvement. Given the significant changes that a DevOps

initiative can bring to an IT environment, it is important to ensure these changes do not inadvertently create new security gaps.

Software design and consulting: These services supply organizations with the skills, knowledge transfer and real-world experience needed to manage their new software development lifecycles from end to end.

Application modernization: Finally, many organizations require assistance with their application modernization efforts. From help getting internal staffers up to speed with containerization to data management, an application modernization engagement can spell the difference between a highly successful DevOps initiative and a less successful effort. By getting this piece of their DevOps efforts right, organizations can ensure more nimble applications that reduce risk to the organization and meet the needs of the business.

CDW: We Get DevOps

With an industry-leading portfolio of partners, CDW's DevOps services team can help organizations bridge gaps between development and operations — and optimize tools and infrastructure across departments and disciplines.

Infrastructure as Code: Infrastructure as Code creates a number of business advantages that stem from automated provisioning and enhanced security. However, many organizations find that IaC can be a challenge to design and implement. CDW's DevOps team can provide complete training to shore up knowledge gaps and ensure that internal teams have the tools they need to become successful with IaC.

Containerization and app modernization: By modernizing applications and making them cloud-ready through containerization, organizations can deliver and patch solutions faster than ever. But an app modernization effort may require changes to existing IT practices, technologies and tools. CDW consultants can help internal stakeholders choose the most efficient containerization solution for their businesses, create agile development environments and get the most out of their hybrid and multicloud infrastructures.

Continuous integration and continuous delivery: DevOps initiatives require culture changes. CDW can help organizations develop the culture of collaboration necessary for continuous integration and continuous delivery, as well as deployment,

the big payout of CI/CD activity. All of this involves breaking down organizational silos so that teams can work together to deploy code as fast as possible, without interrupting business operations.

CDW Can Design, Orchestrate and Manage a Comprehensive Infrastructure Strategy

CDW's simple, smart, scalable and flexible services portfolio provides a fully automated and managed infrastructure across your entire network, whether on-premises, hybrid or in the cloud.



DESIGN for the Future

Consult with our team of technology experts to plan a solution that fits your unique needs and optimizes business impact.



ORCHESTRATE Progress

CDW Amplified™ Infrastructure services help you build and deploy your custom infrastructure utilizing best practices.



MANAGE Operations

Our world-class, certified staff monitors and manages your infrastructure 24/7/365 to ensure operational efficiency and security.

Sponsors



Learn more about how CDW can help your organization optimize its use of DevOps and containers.

CDW®, CDW-G® and PEOPLE WHO GET IT® are registered trademarks of CDW LLC. All other trademarks and registered trademarks are the sole property of their respective owners. Together we strive for perfection. ISO 9001:2000 certified MKT49855 – ©2021 CDW LLC

