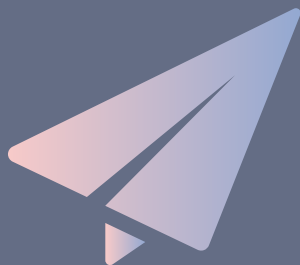


YOUR GUIDE TO

DevOps

IN GOVERNMENT



CONTENTS

**EXECUTIVE
SUMMARY**

2.

**HELPING
GOVERNMENT THRIVE
IN THE APPLICATION
ECONOMY**

7.

**UNDERSTAND
THE PROBLEM**

4.

**LEARN ABOUT
DEVOPS**

8.

**CALIFORNIA
TAKES ON
DEVOPS**

10.

**DIFFERENTIATE
FROM OTHER
METHODS**

12.

**OPTIMIZING &
ACCELERATING
YOUR DEVOPS
PROCESS**

15.

**TEXAS:
WATERFALL TO
AGILE**

16.

**DECIDE
WHEN TO USE
DEVOPS**

18.

TS

**DEFINE WHAT
YOU NEED**

20.

**BUILD YOUR
TEAM**

22.

**USING
CULTURE & TECH
FOR DEVOPS**

25.

**GET STARTED
WITH DEVOPS**

26.

**U.S. CITIZEN &
IMMIGRATION
SERVICES**

28.

**MEASURE
YOUR SUCCESS**

30.

**DEVOPS
WORKSHEETS**

32.

RESOURCES

34.

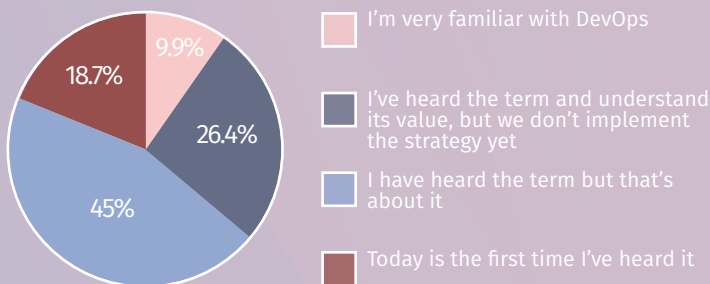
**ABOUT &
ACKNOWLEDGMENTS**

35.

EXECUTIVE SUMMARY

If you work on digital projects, you have probably heard the term DevOps. Even if you don't, that word is likely floating around the halls of your agency. But if you're like the 45 percent of government workers who took our recent survey, you don't really know what the term means or why it's important.

Q: How would you describe your current level of knowledge of DevOps?



Have no fear. We're here to answer all of your DevOps questions, including "What is it?" and "Why does it matter?" This guide will provide tactical instruction, plus advice from DevOps experts in government, to help you:



Understand the problem



Learn what DevOps is



Define what you need



Build your DevOps team



Get started with DevOps



Measure your success

Our Government EXPERTS



CHRIS CRUZ

Chief Deputy Director of Operations

DEPARTMENT OF TECHNOLOGY

STATE OF CALIFORNIA

PETER EICHORN

Director of Technology

TEXAS.GOV

STATE OF TEXAS



JIM TUNNESSEN

Chief Technology Officer

FOOD SAFETY AND INSPECTION SERVICE

UNITED STATES DEPARTMENT OF AGRICULTURE

DIEGO LAPIDUZ

Software Engineer

18F

GENERAL SERVICES ADMINISTRATION



UNDERS

the problem



In traditional application development processes, developers create a product without a full understanding of operational requirements and processes.

In any organization that creates its own technology applications or software, there are two broad teams that complete that task: developers and operations staff.

Traditionally, these teams follow a linear development process. Developers create code for the service, based on broad parameters provided by the organization's leadership. Once the developers build the product, they send it to the operations team to deploy and manage.

This handoff is often referred to as “tossing it over the wall” because operations and development staff pass the product off without truly understanding how each team is using the application. It's like there is a wall between them that prevents transparency or real communication.

That wall creates all sorts of problems. Because developers don't have a clear view of how operations will use the service, the product they create often doesn't stand the test of real-world deployment. The new service can disrupt existing workflows, overload or damage current IT systems, compromise security, or even fail regulatory requirements because developers didn't have enough information during development.

“This is particularly bad in government, because the steps in the [development] process are a lot stiffer than what they would be in the private sector,” said Diego Lapiduz, a Software Engineer at GSA's 18F. “In the private sector, you don't have the same compliance, security, or privacy [requirements]. But in government, we have to abide by laws and regulations throughout software deployment.”

TAND



In some instances, operations teams create workarounds to use the service. More often, they have to toss it back over the wall so that developers can fix bugs, increase security, or alter other features.

When errors are significant, this reversal can be especially disruptive to other coding projects as developers focus on immediate fixes to the already-deployed service. On the other side of the wall, operations personnel are left waiting for their needed applications to be improved and re-deployed. Both sides lose time and resources while the product is tossed back and forth. End users suffer, too, because they have to wait longer to access a completed online government service.

That's a big problem.

So what do we do about it?

Don't worry; we've got you covered. Keep reading.

CA Solutions to Support Your Agency's Digital Transformation

Agile Management

- Agile and Transformation Consulting
- Project and Portfolio Management
- Agile Application Lifecycle Management

Security

- Identity and Access Management
- API Management

DevOps

- Continuous Delivery
- Application Performance Management
- Unified Infrastructure Management



Government,
rewritten by software

Helping Government Thrive in the Application Economy

An interview with Steve Mazzuca, Director of Application Delivery at CA Technologies

The application economy – where everything is delivered as a service to drive professional and personal behavior – has overtaken the private sector. Now, government is getting into the service game. “Every CIO, every director, every leader in an agency has to think about how they deliver services and in an application economy the question is how do I deliver everything digitally, faster, securely, and with better quality” said Steve Mazzuca of CA Technologies, a leading software and solutions provider to the public sector.

To create those digital solutions, many government organizations are turning to agile workflows infused with the collaborative techniques of DevOps. However, the public sector faces unique challenges to adopting this process, particularly as it relates to privacy and security concerns within and among agencies. How can an organization embrace a transparent, collaborative development process while protecting sensitive information in a lengthy and rigid procurement structure?

From CA Technologies’ perspective, the key is to provide a better integrated tool chain beginning with a graphical logical business process map of the requirements linked to the data for test data management. This allows the team to visually map all test scenarios and then automate the test creation, optimize the number of test cases needed, and provision the data needed to test from either masked production data or synthetically generated data.

The second piece of this puzzle is to create a model or simulation of a dependent system or application. This allows development and test teams to run integration, regression and even performance tests against “virtual services.”

The final area is automation. You must build in as much automation as possible all along this tool chain. This means looking at release orchestration tools and moving away from heavily scripted environments.

Creating Real Test Environments

“The Intelligence Community is moving quickly to embrace all of this. They have the need to test in unclassified, lower environments against production like data and systems,” Mazzuca explained. “They need to simulate systems, produce synthetic data, and automate the release and deployment process.”

“In addition, health agencies and DoD agencies are also using technology to test against simulated versions of systems not available to them for testing. This is critical to the future to interoperability of electronic health records,” he added.

By using synthetic or masked sensitive information, developers and operations staff can collaboratively work with life-like code and data. That allows teams to test and understand the real implications of their development decisions, rather than working in an approximate environment that doesn’t show the full impact of potential changes.

That creates more reliable, secure projects. “We build security all the way through the process. It’s not at the end of the road that we do security,” said Mazzuca.

Reducing System Complexity

Every day we see examples of applications or services in production failing due to lack of proper testing. In most cases, this is because the test teams did not have the proper data available to test. CA Technologies’ approach is to create a test data warehouse where testers are able to find and reserve data on demand.

“This concept is very new and exciting to our government customers. Traditionally they have relied on DBAs to create test data in a very slow and methodical process. Now as business requirements change, teams can quickly pivot and set up automated tests with the correct data providing a greater amount of test coverage. This allows agencies a huge amount of risk reduction” said Mazzuca

Delivering Better Services

Ultimately, this technology-enabled approach to DevOps allows teams to create the best solutions possible for both the organization and end user.

Model-driven development with fake data, emulated environments, and automated processes allow for faster throughput, lower risk, less security hassles, and better services to the war fighter and US citizen.

As government workers and contractors focus less on overcoming internal obstacles, they can focus on identifying and remediated service errors – ultimately creating the best service possible for their users. Mazzuca impressed that this emphasis on service elevation, rather than simple delivery, is key if the public sector is going to keep up with the expectations of citizens in an application economy.

“From our perspective, the application economy is not simply a new buzzword but truly a transformational movement in the way services are delivered around the world, both to us as consumers in the commercial sense and as citizens, veterans, warfighters in the public sector sense,” said Mazzuca. DevOps, empowered with the innovative tools and services of CA Technologies, allows organizations to delve into that service-focused movement and truly serve citizens.

LEARN

about DevOps



DevOps is a project management methodology that solves the problematic product handoff between developers and operations staff. Instead of operating in silos, both teams collaborate on a project from start to finish.

They're using the same platforms, speaking the same language and sharing everything they do on a project in real time. That's where DevOps gets its name — it's the melding together of developers and operations staff operationally, culturally, and technically.

"DevOps is about taking down that wall and working together and making sure that the requirements are not just a piece of paper, but something that is living," explained Lapiduz.

Why pursue DevOps?

Collaboration — the very core of DevOps — is always a good thing. Bringing more minds to the table engenders more diverse ideas and helps focus teams on a common mission. Plus, it increases transparency across an organization.

But in addition to the basic benefits of collaboration, there are a ton of other reasons to use the DevOps process. It can help your agency achieve:

Quicker development

For many DevOpsers, this is the most important benefit of the process. In fact, Lapiduz said, "the end goal is to develop and shape code more efficiently and faster, and with more iterations."

Because developers and operations teams don't have to waste time throwing code back and forth over the metaphorical wall, they can release iterations of their product sooner. And when they need to make changes or improvements to those applications, they can do so faster because both teams have a shared understanding of the project's functions and goals.



Greater quality assurance

But even as workers deliver more services faster, they don't have to sacrifice quality with DevOps. Actually, the process minimizes errors through two distinct attributes.

First, DevOps allows both teams to see the full scope of a project. That means they have a better understanding of how any single change — say, a change in a line of code or an update to a hosting platform — will affect the rest of the application. In traditional development processes, one change could derail an entire project. That's far less likely because of the collaborative workflow of DevOps.

Second, automation replaces many human processes within DevOps. We'll dive into that in greater detail later in this guide. But for now, know that DevOps reduces human error by automating many core development and deployment functions.

Better end products

With DevOps, "we see higher-quality solutions, and at the same time we see flexibility," said Texas.gov Director of Technology Peter Eichorn.

When they use DevOps, both developers and operations workers gain a better understanding of the product they're working on. Developers get to know the real-world conditions in which their code will operate. Operations teams learn what makes their services tick and why they were built that way. Together, both teams can make better decisions about how to build and deploy services, and that results in a better product overall.

And because there are minimal barriers between the development and operations environments, employees can make changes with minimal stress to ensure that applications keep pace with evolving needs and scenarios.

Happier end users

Ultimately, DevOps helps government agencies better serve citizens in the Digital Age. As users increasingly expect government resources to be provided with the same mobility, agility, and utility as private sector services, agencies seek ways to quickly meet those demands. When they turn to DevOps, they're able to deliver those services and meet citizen expectations.

Read the case study on the next page to see how these benefits came to fruition when the state of California pursued DevOps.

CALIFORNIA

takes on DevOps

When the California Department of Health Care Services began creating a new medical eligibility data system within the state's health care portal, then-Chief Information Officer of Enterprise Innovation and Technology Services Chris Cruz saw an immediate problem: "We were tasked with doing something that probably had a five-year system development lifecycle in 18 months."

The existing development process would not support that deadline. "It took weeks to set up a test and development environment, due to our normal operations," explained Cruz. "We'd have to buy servers, we'd have to wrap and stack them, and then configure them. And when we stood up services within a shared services contract, we found that they were charging us additional amounts to stand up those environments in a short period of time."

Earlier versions of the process were already failing because of previous sacrifices. To expedite the process, Cruz said the department skipped user and performance testing to roll out the program. As a result, functionality was deferred, defects went unaddressed, and when citizens used the online portal, they couldn't receive coverage in a timely manner.

The portal, along with the process that supported its users, needed a quick and effective revamp. "I was thinking about

what kind of tools we could have in place up front that would help us repurpose development in test environments, so we could do comprehensive testing," recalled Cruz. "So we didn't find ourselves in a backlog situation, where we're making changes that are very costly to an older mainframe legacy application."

To remedy these problems and create a working system by deadline, the department turned to DevOps. "I realized that having a DevOps process in place would help us streamline and get a test environment up and running in an expedient manner," he said.

DevOps allowed software developers and operations teams to continually test and deploy changes to the portal, rather than follow a linear process of static development. The collaborative process was quicker and more effective, and the resulting data system was more user-friendly than previous versions.

DevOps also allowed the department to retain ownership of the project's development, rather than turning to a service contract. "When we have sustained repeatable solutions that we can all use, it's easier for knowledge transfer of staff, and to maintain that institutional knowledge in the state," said Cruz. "It puts the dependency back in the state's hand to own some of these projects."

Cruz and his team used DevOps to streamline not only the health care eligibility system, but also nearly all of the department's software development process. "We used this tool to develop an entire health care innovation strategy...built on velocity, mobility, and agility," he said.

Now Cruz is Chief Deputy Director of Operations at the California Department of Technology. When he took on this new role, he brought DevOps with him to help achieve new goals. "We're trying to build standardization across our whole system development lifecycle within the state, and that's where I'm taking our DevOps approach at the Department of Health Care Services and trying to expand that at an enterprise level," he said.

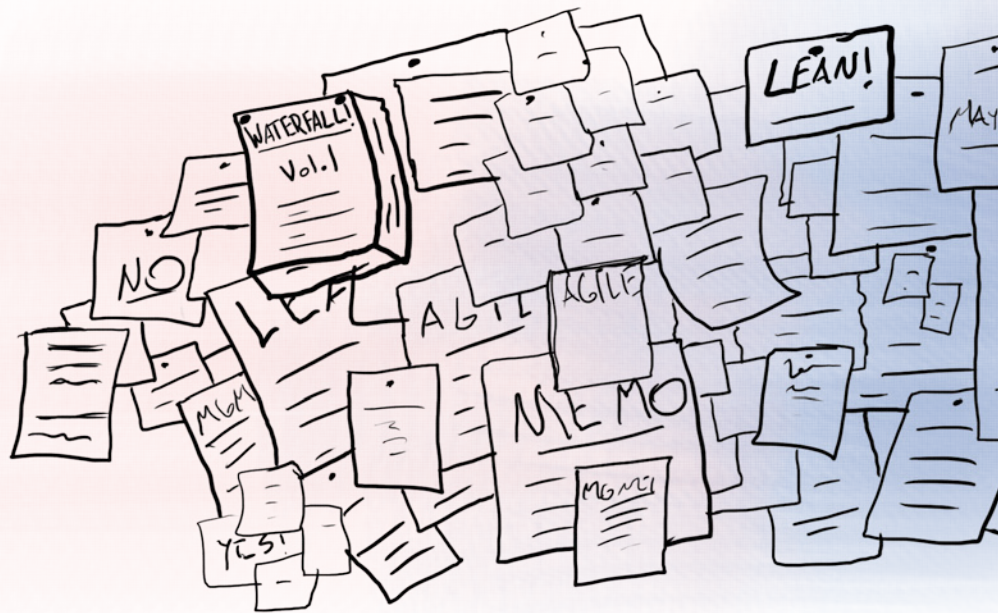
Specifically, Cruz wants to standardize initiatives such as the state's cloud portfolio, CalCloud, across state organizations to reduce costs and streamline operations. DevOps will help facilitate a common understanding of those projects by including multiple teams and their tools in a single, collaborative workflow.

ERNIA



DIFFER

DevOps from other



Within government, there is a raging debate over what sort of project management methods are best suited to deliver new government services and how to implement them.

This debate isn't new, however. It has been ongoing since government entered the digital services arena.

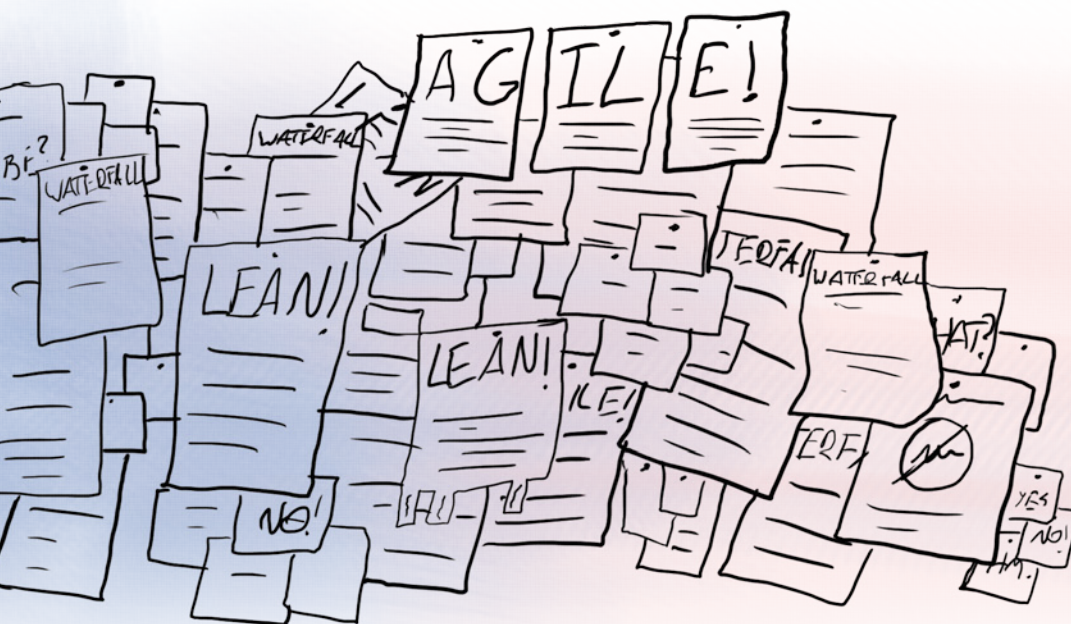
In the 1990s and early 2000s, [lean project management](#) was in vogue as the major challenger to more traditional approaches to development. Many heralded the fast-paced process as a way to cut costs without sacrificing quality. Today, you'll hear less about lean but a lot more about Agile, an iterative software development model that many suggest is the way forward for digital government.

As a challenge to Agile and other alternative processes, many public sector leaders still believe that traditional waterfall methods are the best way to ensure control over project budgets, timelines, and outcomes.

The fact is, there is no single right answer to this debate, as we pointed out in a recent guide, "[Agile, DevOps, & More: How to Succeed at Government Project Management](#)." Some processes are better suited for certain projects than others. For instance, you probably want to build a naval ship via a traditional, linear process, while software development calls for a more iterative approach.

DIFFERENTIATE

Agile from other methods



But that still begs the question: Where does DevOps fit into all of this?

Let's put it this way: If DevOps were a superhero, it would be the supportive, team-enhancing Robin to Agile's Batman.

Many people mistakenly frame DevOps as an alternative to Agile project management. In fact, as USDA CTO Jim Tunnessen said, "Agile is required." DevOps is not an alternative to Agile project management, but a complement to that iterative approach.

Agile development requires software to be incrementally created, released, and updated based on stakeholder feedback

and testing. But it does not necessarily require developers and operations teams to work together. The wall between the two teams could still exist, even if each group is constantly tossing the product back and forth for improvement.

If you add in DevOps, that wall comes down. Not only are teams iteratively testing their products with end users, but they are also collaboratively working with each other to better understand and produce those products. Ultimately, DevOps enables Agile workflows to happen even faster, with more communication and accuracy. Read the next case study to learn how DevOps can help an organization hone its Agile processes.

WHEREVER YOU GO YOUR DATA IS READY. **WELCOME TO DATA FABRIC**

Manage your data seamlessly across
any environment and move it to
where you need it most. The future
of data management is here.

netapp.com/datafabric



INDUSTRY SPOTLIGHT

Optimizing & Accelerating Your DevOps Process

An interview with Mike Dye, Solutions Infrastructure Architect and Chief Technology Officer at NetApp

There are a multitude of benefits to using DevOps: stronger collaboration and teamwork, better transparency, increased productivity, and, best of all, better quality of products and deliverables. But while it may seem like DevOps couldn't get any better, there are certain strategies your organization can apply to fully reap the process' benefits.

In an interview with GovLoop, Mike Dye, Solutions Infrastructure Architect and Chief Technology Officer of NetApp, emphasized that automation and flexibility to use and maintain multiple datacenters are critical to optimizing and accelerating your DevOps process.

NetApp offers a number of software tools, systems, and services to help you manage and store your data – all of which can help you make the most of your DevOps process.

Automation and Your DevOps Process

Automation is especially important to an incremental process like DevOps, because continual small enhancements to your product can incur significant manual labor costs. Whether you need to add more memory or storage to your software, it's important to automate those changes so that you can automatically redeploy and monitor how your changes affect the overall outcomes.

"The biggest value of the DevOps process is that it allows you to make incremental improvements as you operate the system," Dye said. "But by being able to automate those incremental changes, this allows you to make tweaks or re-architect everything if necessary. So it's important to think about those tools in conjunction with DevOps."

NetApp offers a number of tools that work well with a variety of automation frameworks and services. For example, NetApp's

automation tools are designed to work in conjunction with Chef, a configuration management tool that uses domain-specific language for writing system configuration "recipes" and streamlines the task of configuring and maintaining an organization's servers. This means if you decide to add an ounce of memory here or a tablespoon of storage there for your recipe, you can automate it and keep track of the changes.

Just like with any recipe, however, it's important not to introduce too many changes at once. This incremental approach helps you keep track of what's being affected and what changes you've made in your software environment.

It also gives other teams greater visibility into your workflow and adjustments. Dye emphasized that automation is not just about consistent management of changes in your software, but also increasing trust between your organization's teams since automation allows you to keep track of who made what changes.

"With DevOps and automation, it's crucial you pick the right tools and understand who's doing what," Dye said. "It's about making sure that administration and operations trust and work in concert with the development teams."

Automation is key to helping you accelerate and optimize your DevOps process. Like many government entities, however, you may be worried about maintaining security and control over your data with modern automated processes and systems. What if there's a glitch that compromises important data? That's where managing your Data Fabric comes in.

Managing Your Data Fabric

It can be especially difficult managing DevOps in a government setting. While it is

important to maintain control and security over your organization's data, you also need to be flexible enough to enable the innovation needed for DevOps. To manage this balance, NetApp offers Data Fabric, a tool that allows you to optimize a mix of private and public cloud services.

"With NetApp's Data Fabric, we can run our data services everywhere," Dye said. "As an agency, you have your own datacenter with your own private cloud. This allows you to utilize one of the colocation service providers that have some equipment in their datacenter."

The NetApp Data Fabric gives your organization the ability to run a variety of data services in different environments, maintaining data both in the cloud and in your own datacenter.

Because DevOps is iterative, you will need greater performance and equipment at some times and less at others. Data Fabric lets you upscale and downscale your capacity as needed. Regardless of where you are in the development process, whether you have some data in a public cloud, private cloud, or on-premise, you can be sure your data is streamlined and secure with Data Fabric.

The combination of automation and capabilities like NetApp's Data Fabric allow you to fully modernize your data infrastructure and processes all while ensuring your data remains safe. Using this combination, your organization can better take advantage of the benefits of DevOps. These tools allow you greater flexibility, ability to adapt to change more rapidly, security, and enhanced trust between teams, and, most importantly, the opportunity to take your DevOps to a whole new level.

TEXAS

waterfall to agile

How effective can DevOps and Agile development really be? Just ask Peter Eichorn, Director of Technology at Texas NICUSA, LLC, a subsidiary of NIC Inc. that operates Texas.gov, the official website for the state of Texas. His team decided to apply a more Agile approach to software development for a new, large state agency digital service and saw tremendous results with DevOps.

“DevOps really ended up meaning two things for us,” Eichorn said. “The first is Agile development for developing software with automation around the environment, deployment, and operation. The second is reduced risk in producing high-quality products.”

Eichorn and his team started out using traditional waterfall development, a more sequential process in which progress steadily flows downward. Although that process can occasionally be effective in software development, Eichorn and his team decided to try an Agile approach when they faced a particularly tight time frame on a project but didn't want to sacrifice quality.

“Oftentimes, software projects can run quite long in a typical waterfall approach,” Eichorn said. “We found that incrementally delivering through Agile development has a lot of value on the quality side because it really improves our engagement, flexibility, and product quality.”

As the team transitioned from basic to more advanced scales of Agile development, it saw that using DevOps was highly beneficial. Teamwork, collaboration and end-user engagement drastically improved. “Because we get good engagement, meaning we don't go away to work on the project for 18 to 24 months, we actually get to see the project and better evolve it in a shorter time frame,” Eichorn said.


Building the type of software required for this new, large online service would normally take 18 months for Eichorn and his team. But after applying DevOps, the team managed to do it in eight.

It almost sounds too good to be true. Better collaboration, shortened time frames, and better products? What's the catch? The benefits of DevOps and Agile seem all too

clear, but there are certainly challenges to implementing DevOps in a government setting. Eichorn cited cultural change and getting used to the incremental project development process as two of the most difficult challenges.

“Both the strategic and conceptual sides of change can be really hard — particularly getting your organization to shift its mindset toward DevOps,” Eichorn said. “Often, there's the desire to figure it all out up front, like with the waterfall approach. When your team is used to having more risk mitigation plans, it can be difficult to adjust to the Agile process, where you're incrementally working and learning as you go.”

Tackling mindset and strategy changes in an organization using DevOps can be especially challenging if your agency has silos. Fortunately, there are many ways to overcome that. Eichorn and his team persisted through such challenges and managed to evolve from basic scrum methodology — in which teams meet briefly every day to share progress on projects — to a more sophisticated, scaled Agile framework of constant iteration.



Based on the successful experience of the approach implemented with Texas.gov, Eichorn offers these best practices for starting DevOps for an Agile project at your organization:

Hire an Agile coach

A coach can guide your organization through the painful process of change from waterfall development to more Agile processes like those in DevOps. Not only can a coach guide you through the technical components, but also help your team navigate the drastic cultural shift DevOps brings.

Provide open demos

DevOps does not just benefit your development and operations teams. Every two weeks, Eichorn and his team provide open demos of products for anyone on the team to see. This helps with cultural visibility, transparency, and better product quality. Including everyone in the organization on software development projects can eliminate silos between teams and departments.

Start with an important project

It seems counterintuitive, but it can be detrimental to an organization to start implementing DevOps for minor side projects, Eichorn said. Using it on critical projects, however, can help focus your team and motivate everyone to succeed. When your team runs into challenges, more people are likely to rally behind the project and adjust goals as necessary. With side projects, people may be tempted to just drop the DevOps approach and the project all together if something is not going well.

Eichorn and his team evolved their techniques, going from waterfall development to basic scrum to scaled Agile development and DevOps. With the new online service, they not only improved their processes and products while shortening their time frames, but they also navigated the inevitable challenges that occur when implementing DevOps in a rigid government setting. Now they can continue to grow with more seamless communication, transparency, collaboration, and excellent product development.

DECIDE

when to use DevOps

By now, we hope you are jazzed to start implementing this collaborative approach to service development. However, planning and implementing DevOps is a journey that can take a long time and considerable resources. And that journey is only worth it only if DevOps fits the unique goals and direction of your agency.

When determining if DevOps is right for your organization and its project management needs, consider these factors:

YOUR APPLICATION INFRASTRUCTURE

Self-service infrastructure is key to productivity and autonomy of engineers across all disciplines. You also need an elastic infrastructure that can turn on, scale up, scale down, and turn off quickly and automatically. Cloud-based platforms and software, for example, leave you better prepared to quickly adopt new processes and products.

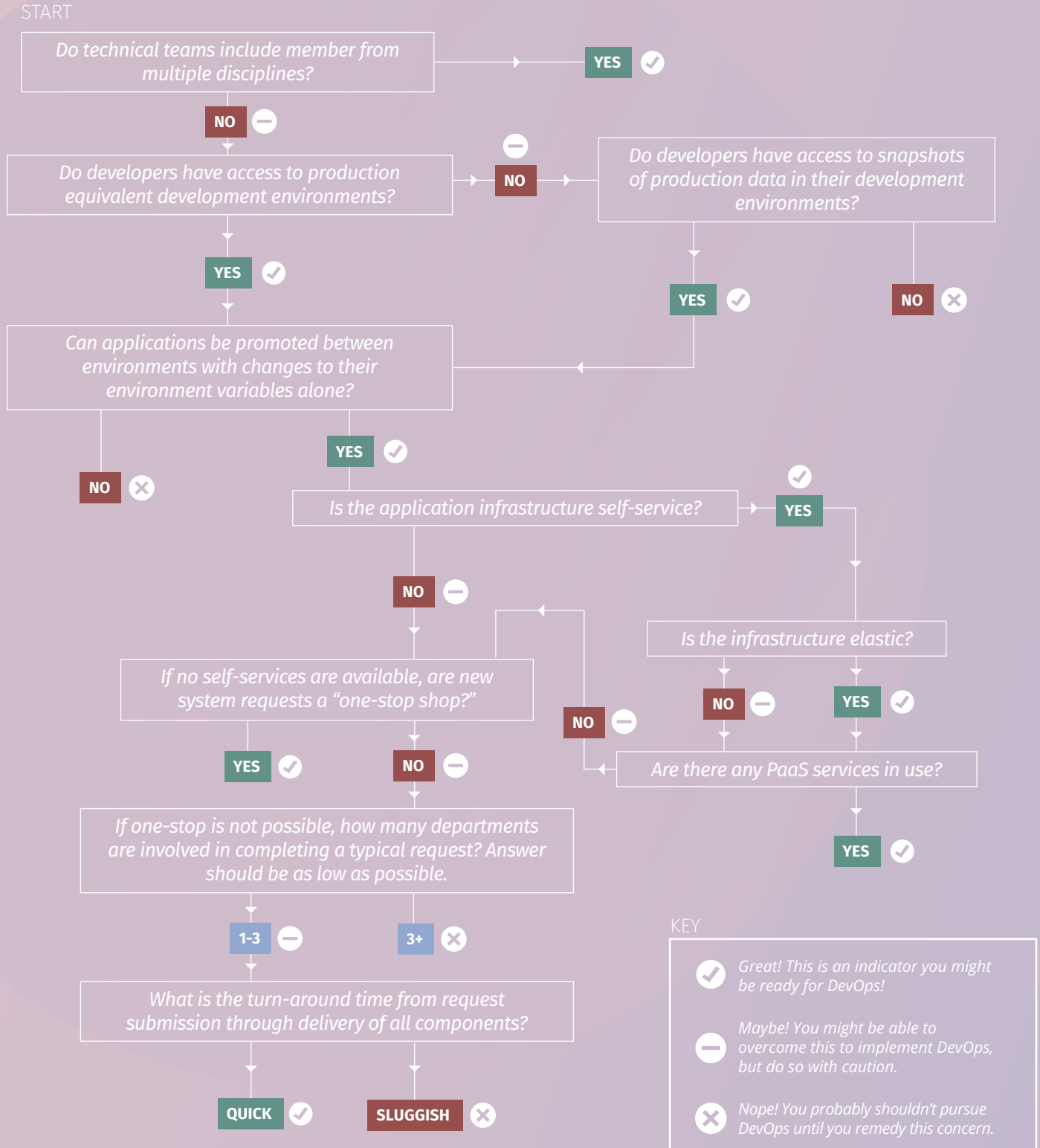
YOUR PRODUCTION ENVIRONMENT

It's important to have functionally identical environments and data for reliable development and testing. Being able to promote applications between environments implies secure, seamless application design.

YOUR ORGANIZATIONAL STRUCTURE

Technical teams need members from multiple disciplines to make products that appeal to a broad range of users. Multidisciplinary teams also set up your organization for better collaboration and information sharing.

Use this flowchart, based on the [18F DevOps Assessment Guide](#), to guide you through these and other considerations:



DEFINE

what you need

Because most people disagree on the perfect recipe for DevOps, it's difficult to define the ingredient list.

To keep things simple, let's focus on CAMS, the definition John Willis and Damon Edwards created at a 2010 DevOps conference in Silicon Valley.

CAMS stands for culture, automation, measurement and sharing — the four principles that Edwards and Willis assert are the core tenets to any DevOps process. Others have expounded on the list, so don't be surprised if you see longer acronyms like [CALMS](#) and [CALMSS](#) floating around DevOps discussions. However, to date, very few DevOps theorists have deducted from the CAMS list, so let's start there.

To practice DevOps, you'll need:

Culture

Especially in traditional government settings, a cultural shift is mandatory for DevOps to take root and succeed. Although waterfall approaches to service development require linear handoffs, discrete ownership and a certainty of outcomes before a project ever begins, DevOps requires collaboration, flexibility, and uncertainty.

"You have risk mitigation plans in waterfall techniques. You are going to study the whole project, figure out all of the risks, mitigate all of them, and then get started," Peter Eichorn explained. "But in Agile, you're incrementally working through those things and learning and discovering as you go." And with DevOps added to the mix, you have to face those risks collaboratively, rather than as distinct teams with separate responsibilities.

To meet the cultural requirements of DevOps, our experts impressed the need

to focus on people. After all, "people are a big part of your culture; they pretty much are your culture," said Eichorn.

Cruz suggested starting your cultural transformation by spreading awareness of what changes you envision and then training people to meet new requirements. "That's the area that we need to provide the most amount of care and feeding moving forward," he said.

Of course, that sort of training and support requires resources. Therefore, when you're thinking about the people aspect of DevOps, start with leadership. You'll want their buy-in as you reorganize and prepare your colleagues for DevOps.

Automation

We just said that DevOps is about people, and it is. But it's also about automation. Confused yet?

Many people consider automated processes as a replacement for people-run processes. In fact, automation is a tool to help humans complete software development quicker and with far greater accuracy. Think of it as a complement to the people part of DevOps, rather than a replacement.

18F's Lapiduz offered an example: "Instead of a privacy team having to review each piece of software manually and exchange it manually, they can implement tools that can assure you that your software is compliant."

Automated processes such as privacy checking are critical to DevOps because they allow developers and operations staff

to rapidly iterate designs without sacrificing quality. Automated tools can constantly check that new code is secure, compliant, and compatible with current systems, removing the chance of human error while speeding up the quality assurance process.

As our own software engineer, Josh Hill, pointed out, automation can also help you train new employees in DevOps. “Automation is especially critical when you’re onboarding new staff. It helps them get up to speed fast, without having to have another developer constantly check their work for small mistakes,” he said. “And it’s faster for the new recruits, because any changes they make, it can be checked with an automated system rather than having to copy and paste it into multiple systems for review.”

Measurement

Measuring the progress and outcomes of your projects is as important in DevOps as it is in any other project management methodology. We’ll get into how to measure progress later in this guide. For now, it’s important to understand why measurement is a crucial ingredient to DevOps.

First, you have to make sure any production process fits within the constraints of a service order or development contract, especially in government settings. But it’s easy for iterative processes to exceed their timelines and budgets if you don’t set concrete milestones for progress. You’ll need to establish expected outcomes and a timeline for achieving them before you begin DevOps.

Second, concrete metrics can help build intangible value. As the [U.K. Digital Service points out](#), “data can be incredibly powerful for implementing change, especially when it’s used to get people from different groups involved in the quality of the end-to-end service delivery. Collecting information from different teams and being able to compare it across former silos can implement change on its own.”

Sharing

You might think DevOps sharing is all about sharing code between teams. Of course, that’s a crucial part of the process, but that’s not what we’re talking about here.

To truly get the most out of any DevOps experiment, you have to share your experience of the process.

That’s the whole point of iterative development, after all. Just as you get the best final product by continually testing it with other users, you’ll create the best DevOps cycle by constantly sharing your practices with other teams, organizations, and sectors. Through sharing, you can understand how your processes affect others, learn from other teams’ best practices, and even gain more support or resources.

What about tools?

You’ll notice that there is no room for tools in the CAMS framework. That’s because most DevOpsers ascribe to Cruz’s belief that people and processes enable good technology.

Moreover, because you can practice a rudimentary form of the process simply by having developer and operations teams talk to each other regularly, you don’t technically need technologies to get started. “There are tools that help, certainly, but you can do this in a traditional data center,” said Tunnessen.

Nevertheless, if you decide to really dive into DevOps, there are a few tools and technologies that can help. In fact, there are more than a few and they can range from free online platforms to expensive, custom-built solutions. Rather than focusing on particular tools, consider what you want to achieve with them. The most common three functions that DevOps tools enable are:

- **Continuous Integration (CI) & Delivery**
Whether someone in development or operations makes a change to the service, you want that change to immediately be accessible to everyone else on the team. That way, everyone is on the same page about what’s happening in a single operating environment. CI also enables automated testing of code changes and deployments. Tunnessen recommended adopting cloud environments that give you complete control over an infrastructure from any part of it.
- **Configuration Management.**
Unless your new service is going to operate in a vacuum (it probably will

not), you’re going to want to know how it will interface with your organization’s larger infrastructure of tools, technologies and processes. This sounds obvious, but many developers work on a product in isolation because it allows them to exclusively focus on that project’s needs. To ease their transition to a more context-mindful operating procedure, acquire a configuration management solution that automatically considers system dependencies when code is tested. Lapiduz pointed out that this functionality also creates empathy for the operations team.

- **Analytics**

You want to create software or applications that people will actually use. The easiest way to track how users are interacting with your service — and share those results with others — is through data analytics. This capability can be achieved with custom-built applications, but you can also use basic web analytics platforms to get started.

Depending on your budget and team needs, decide how ambitious you want to be in matching tools to these capabilities. For some teams, simple solutions will be the route to get collaboration off the ground. For others, custom development environments may be required to achieve more sophisticated development tasks.

Build your team

Remember, DevOps is about culture. That means it's about people. But what kind of people?

For DevOps to work, you'll obviously need developers and IT operations staff. These are the core of your DevOps team. But what is a DevOps engineer, coder, or IT specialist? Here's a sample job description for a DevOps Engineer from GSA's 18F:

DEVOPS ENGINEER

Experience serving as the engineer of complex technology implementations in a product-centric environment. Comfortable with bridging the gap between legacy development or operations teams and working toward a shared culture and vision. Works tirelessly to arm developers with the best tools and ensuring system uptime and performance.

Primarily responsible for:

- Deploying and configuring services using infrastructure as a service providers (e.g., Amazon Web Services, Microsoft Azure, Google Compute Engine, RackSpace/OpenStack)
- Configuring and managing Linux-based servers to serve a dynamic website
- Debugging cluster-based computing architectures
- Using scripting or basic programming skills to solve problems
- Installation and management of open source monitoring tools
- Configuration management tools (e.g., Puppet, Chef, Ansible, Salt)
- Architecture for continuous integration and deployment, and continuous monitoring
- Containerization technologies (e.g., LXC, Docker, Rocket)

That looks like a typical engineer job description, right? The only difference is — you guessed it — culture.

DevOps team members will need the same technical skills that they would for a non-DevOps role. However, they also should be willing to embrace the collaborative, iterative approach of DevOps.

"DevOps is a practice," said 18F's Lapiduz. "It's not something that you are, it's something that you do. That is about breaking down silos. It's about collaboration between teams."

At 18F, there are no operations teams: "We have a policy where if you're a developer and you build software, you run it yourself. We don't have a dedicated operations team. Instead, we ensure developers have access to all the tools that they need to deploy their software. This empowers developers to focus on innovation rather than on paperwork," Lapiduz said.

This doesn't mean that you should get rid of your operations team. It just means giving your teams, whether they are developers, operations staff members, or both, the tools they need. That's why 18F's job description specifies someone who is "comfortable with bridging the gap between legacy development or operations teams and working toward a shared culture and vision."

***Let's look at the ideal persons
for your DevOps team members***



Developer Dan

As a Developer, Dan is somewhat of a generalist. This makes him open to working with others and especially keen on engaging in projects that incorporate Agile development. With DevOps, he doesn't have to worry about repeatedly fixing mistakes because operations is sending his code back over the wall. Instead, he works directly with operations to identify pain points, problem areas, and various breakdowns to ensure that he takes their operational concerns into account at the beginning of the design process. With DevOps, Developer Dan avoids redundancies, fulfills operations requirements early on and improves how he designs.



Operations Olivia

Olivia's aim is to ensure that the organization is running as smoothly and efficiently as possible, while meeting user needs. This makes her very open to collaboration and Agile development. Using DevOps, Olivia can take ownership of products designed with Developer Dan because she's part of the design process at each step. She can regularly consult with Dan to make sure the code meets organizational goals so she doesn't have to send it back later because of errors. With DevOps, Operations Olivia can take more ownership in the design process and ensure seamless and continuous development of products, so that the organization delivers exactly what the user needs.



HR Harry

Believe it or not, DevOps even helps Harry do his job better as Human Resources Manager. Harry seeks to help develop an organizational culture and environment in which teams from various departments of an agency work together and are aware of one another's work. He wants them to feel empowered and transparent in the work they're doing.

Through weekly scrum meetings of all agency department heads, Harry can stay in the loop about the integration and output of each team. He is able to stay abreast of what development and operations teams are doing, how they're meeting user needs, and in turn how he can improve in his job. Now he knows what to look for in candidates to join the DevOps team. With DevOps, HR Harry understands the work of the overall agency, making him more effective at his own job.



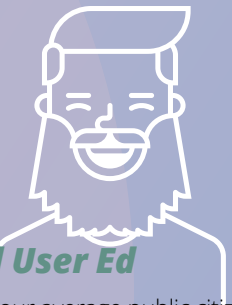
Agency Leader Louise

Louise is all about improving organizational culture to address citizen needs. She makes sure products meet the standards of your organization's users. Agile development is her preferred method because this means she can trust teams to work together within the organization so that everyone is on the same page with product development. DevOps makes Louise happy because she can ensure that projects are being updated according to users' needs in real time. She can count on her DevOps team to work together to quickly address any mistakes and smoothly and rapidly deliver products and services. DevOps helps Agency Leader Louise see reduced turnaround time, efficiency gains, cost savings, and improved organizational culture.



Security Sally

Sally sits on your agency's security team, ensuring that your products and software meet regulations and cybersecurity standards. If there's anyone whose standards you should meet when designing code and products, it's Sally's. She supports collaborative work because it builds trust within the team and she can ensure that products are being designed safely with security as a priority. When security is more integrated with DevOps, Sally can then identify potential risks with new products. Having security metrics in the design early on is incredibly important, especially considering the information that government has to protect. With DevOps, Security Sally can assess the risks of new products ahead of time and make sure that products and services are delivered according to all security regulations and standards.



End User Ed

Ed's your average public citizen. He has some questions that your agency can address, but he prefers to look up the answers online, so he uses your website frequently. Because your agency uses iterative approaches, Agile development and DevOps, citizens like Ed can see frequently updated information in real time and can give feedback if something's not working. With DevOps, End User Ed can feel like he's a priority for the agency because he knows his customer feedback will be quickly taken into consideration and used to improve services. Additionally, he can rely on your agency for good service delivery.

ACCELERATE I.T. WITH DEVOPS

Give your organization:

- A collaborative, open culture
- Faster, automated application delivery
- A dynamic, programmable platform

Red Hat's open source technologies
can help you succeed with DevOps.

LEARN MORE



redhat.

Using Culture & Technologies for DevOps

An interview with Jim Tyrrell, Principal Solutions Architect at Red Hat

If your organization is used to managing projects in a linear, siloed process, the idea of implementing the collaborative, iterative DevOps approach can seem daunting. While there are significant benefits to the methodology, it does require significant changes across an organization. How do you manage those changes to achieve DevOps success?

In a recent interview with GovLoop, Jim Tyrrell of Red Hat, an open source solutions provider, talked about how to tackle DevOps without overwhelming your teams or disrupting your organization's operations. He said the best approach is a low-risk, step-by-step process of encouraging cultural shifts and giving teams the tools to make those changes.

Starting Your Cultural Transformation

Many people assume that, because DevOps is a process focused on improving technology, the main focus should be on the technology itself. However, Tyrrell emphasized that deploying DevOps requires an investment in cultural change first and foremost.

"You're not going to buy DevOps in a box," he said. Nevertheless, you can seek partners to assist you in tackling the human dynamics of DevOps. For instance, Red Hat's professional services can provide the training and mentoring necessary to facilitate that cultural change.

Tyrrell outlined the approach Red Hat encourages at organizations pursuing DevOps. The first thing they recommend is to build a small team who is willing to test and champion the process. "You want to find the leaders in your organization – the people that can really help drive the DevOps change across the organization," he said.

Then, pick the right project. Tyrrell suggested starting with a straightforward initiative that gives your team room to test their new process. "You don't want your 100 percent mission-critical application to be your first foray into this," he said. "It should be that app that's used one day a month when people are doing some sort of reporting or some other small subset of functionality in your organization."

Picking a low-impact project minimizes risk in your first attempt, allowing your team to really dive into DevOps practices without the concern that a mishap will disrupt important operations. Moreover, Tyrrell said that selecting a project with narrow parameters allows the team to test unique process variables more easily. "From a good science perspective, if you're changing all these variables at once, how do you even know which variables are right? You have to limit the number of variables you're changing in an experiment to ensure success," he said.

Once your team has learned how to incrementally make the process work for them and the organization, they can begin applying DevOps to more dynamic, impactful projects.

Applying Tools to Transitions

As your team becomes more involved with DevOps, Tyrrell advised investing in tools to support your endeavors. "You want things like platform-as-a-service where your IT infrastructure is malleable and quickly able to be scaled up, scaled down, and provisioned for your developers," he said.

A platform like Red Hat's OpenShift allows developers to quickly develop, host, and scale applications in a cloud environment.

With OpenShift you have a choice of offerings, including online, on premise, and open source project options, which you can interchange as needed for different projects and requirements.

"It's really just building blocks that can be instantaneously allocated to different operational problems or developer programs, so that coders can start playing with an idea," Tyrrell explained. "It automatically builds a scaffolding for you to start the project as quickly and simply as possible."

For organizations that have taken an incremental approach to DevOps transformation, this frictionless environment is key. It removes barriers for teams to start transitioning to a DevOps workflow, and it allows them to start small and then build capacity as they increase the scope of their new processes.

And by combining those applications with middleware like Red Hat's JBoss, which seamlessly integrates software from myriad open source communities, teams can automate production and testing to even further accelerate the DevOps process. "You've completely removed all the friction of deployment out of your pipeline of software development, such that any changes or ideas you have to your applications as you're running them, can be quickly and efficiently implemented and put into production," said Tyrrell.

By matching technologies and services to your organization's unique approach to DevOps adoption, you can transform both the culture and operations of your teams without overwhelming either. "And that is DevOps Nirvana," Tyrrell concluded.

GET STARTED

with DevOps

Now that you have everything you need, how do you get started? Especially if your teams work in strict silos, without tools to foster collaboration, you might be overwhelmed at the prospect of completely revamping your development and deployment operations

The good news is there is no hard-and-fast rule for what makes DevOps work. Lapiduz from 18F explained: “With DevOps, you can say that you’re practicing it if your team is just talking — if you have a piece of the development team that is just talking to ops, and your ops team is just talking to your developers. But there’s also another end of the spectrum where you actually have no operations team. Everyone works under the same roof and it is all DevOps.”

DevOps requires iterative collaboration, but how you achieve that objective is entirely up to you and what makes sense for your organization. But whatever setup you choose, there are a few tactics our experts agreed make deploying and ingraining DevOps easier. Follow these steps to get your organization started:

STARTED

FOLLOW THESE STEPS TO GET YOUR ORGANIZATION STARTED:

2 BUILD YOUR FIRST TEAM.

Eventually, you'll want your entire organization to be a part of your DevOps strategy, but start small to minimize risk and really learn from your first attempt. Create a dedicated team of developers and engineers who are willing to execute your DevOps vision by testing new processes and tools, advocating for the process, and eventually teaching it to others.

4 COMMUNICATE TO EVERYONE.

Rather than mandating that silos be torn down and processes redrawn, all of our experts agreed that it's paramount to communicate why and how the methodology will benefit participants before it is deployed.

Then, as you begin to integrate teams and processes, be sure to keep communication going. Of course, you'll want to publicize your progress and share your successes — that builds enthusiasm for the process and recognizes the hard work of your change-makers. Eichorn said an easy way to do this is to offer demos of your new products to other teams. "We find it takes a lot of the mystery out of developing software, and it also gains quite a bit of momentum. There's visibility; people can really understand and be a part of something, so it really helps with the cultural aspects of DevOps," he said.

But this communication will be equally, if not more, important when you miss the mark with DevOps. Rather than brushing failures under the rug, communicate where processes go awry and what lessons learned you can apply to your next attempt.

1 DEFINE DEVOPS.

"Bringing people to a common standard and ensuring that you have a structured approach to how you lay out the application development lifecycle reduces risk and increases the likelihood of success for implementation projects," Cruz said.

Because DevOps processes vary in investment and sophistication, your first task is to define what DevOps means for you and your organization. This definition will allow you to determine which changes will be required to deploy the process and set expectations for how it will affect your project's outcomes. For example, if you are low on resources, you might want to implement daily meetings between teams to start information sharing. Alternatively, if you have the leeway to implement collaboration tools and a joint development environment, you may consider a larger revamp of processes.

3 MATCH TOOLS TO YOUR VISION.

While we've repeatedly said that technology is not the central component of DevOps, it is a consideration. You don't want to deploy an innovative process only to have it fail because you don't have the tools to support it. With your team's input, determine what tools you'll need to enable collaborative development across projects.

5 REMAIN FLEXIBLE.

Finally, even as you craft your implementation plan, don't get locked in.

"Don't be overly rigid," Eichorn advised. "If something isn't working for you, or you choose not to do all of the methods and techniques, that's fine. You'll still be successful. You don't want to replace one set of things you did with waterfall, perhaps, with some other aspects that are so rigid that they get in the way of actually doing the work." Remember, there is no one-size-fits-all for DevOps. Accordingly, be prepared to adopt more or fewer DevOps tactics as you experiment with the process in your organization.

U.S. Citizenship Immigration S implement DevOps

At Tunnessen's old job, there was a problem. Like at any other agency, U.S. Citizenship and Immigration Services officials would decide to create a service or application and set a timeline for production. The development team would create an application, but instead of deploying it afterward, they'd shelve it. As Tunnessen put it, "There was a latency in getting things deployed."

He watched this happen repeatedly, seeing good applications and systems metaphorically gather dust on the operations side of things. It drove him crazy, so he decided to do something about it. Recognizing a disconnect between teams, he called operations and told them that he was driving to their separate location to meet with them and solve this problem. And that's exactly what happened. Here's how:

Cooperation and collaboration are key to DevOps, and to start the process, Tunnessen instituted a Kanban process — a method for managing work that emphasizes just-in-time delivery and a list of tasks that team members pull from — with a Excel spreadsheet and SharePoint.

This process involved team leaders and program managers who were in charge of releasing applications, and Tunnessen put them in charge of entering all systems into the list and prioritizing them. From there, the operations team was responsible for monitoring the list each day and then finishing the tasks in order of priority. Tunnessen also instituted a notification system, so the entire team knew when operations deployed a product or if they required more information or assistance before deployment was possible.

To further ease collaboration, he moved the entire process to a commercial cloud, so he was able to oversee everything without having to be in the data centers. He also created new teams that included program managers, developers and testers, plus quality assurance, security and operations workers. Operations teams worked directly with development teams so they could communicate whenever there was a problem.

The results of these new matrix teams, or as Tunnessen put it, "everybody working together in a bucket," were incredible. Instead of separate teams working in

isolation, there was now a continuous delivery pipeline. Much of the development process became automated, and when a system was ready, the project owner could easily deploy it. Under the previous system, projects could require 18 months to three years and massive amounts of paperwork to develop. Now, it is possible for the team to receive a request for a four-page form in the morning and have it deployed and in production by the end of the day.

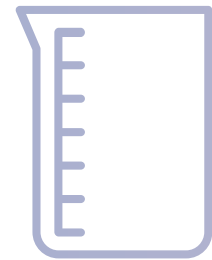
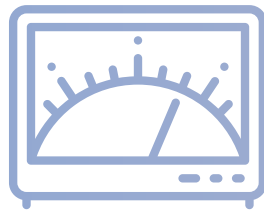
But "tools and technology will only do what people really allow them to do," said Tunnessen. He emphasized that process, culture, and organization are essential to developing strong DevOps. Development, testing, operations, and security teams all have to work together on projects and feel a responsibility for those tasks. Tools, such as cloud and continuous integration solutions, are important, but it takes a cultural and organizational change for teams that have traditionally focused on different aspects of the process to come together and collaborate at all stages. But when they do, great things can happen.

p & ervices



MEASURE

your success



We mentioned earlier that measurement is a core tenet of DevOps, but how exactly do you do that? With iterative approaches like this one, many people mistakenly assume that you can't measure the success of the project. In fact, DevOps and the agile processes it enhances are particularly outcomes-focused. After all, the whole point of implementing DevOps is to create a better product, quicker.

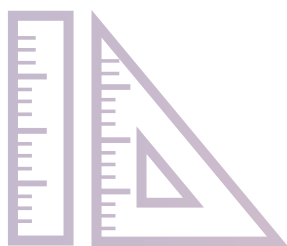
We mentioned earlier that measurement is a core tenet of DevOps, but how exactly do you do that? With iterative approaches like this one, many people mistakenly assume that you can't measure the success of the project. In fact, DevOps and the Agile processes it enhances are particularly outcomes-focused. After all, the whole point of implementing DevOps is to create a better product quicker.

One of our own operations staff, Josh Hill, shared his rubric for success. When he completes a project using DevOps, he asks three simple questions:

Does it work?

Whether it's a service, a platform, or piece of software, your product was created to perform a certain function or set of functions. The first and most basic measure of success is whether it is capable of performing those tasks upon deployment. Keep in mind that your project's objectives may not be totally technical, even if you are developing a piece of software.

E



Does it work for the user?

This second metric considers whom you're making a product for and how they will use it. A piece of software may technically work, but what good is that if no one can really use it to achieve a goal? Therefore, a holistic rubric will account for usability and functionality.

For instance, Tunnessen said he considers the value a new product will bring to an agency — whether that's hours of labor saved or enabling workers to serve a new area — as a metric of success on all his DevOps projects. Alternatively, Eichorn said he uses employee, constituent, and partner agency satisfaction as core determinants.



Does it work for many users?

This last metric considers sustainability, which is the product's longer-term usability and functionality. You'll want your product to continue achieving its objectives and maintain performance even as you increase your user base. That ensures that you'll get the most out of your solution today and in the future.

"The net of all of it is that you deliver what you say you're going to do and then meet the expectations of the agencies and the constituents. That's the ultimate measure of success," Eichorn said.

In many ways, this rubric for success isn't that different from any other process' criteria. Even with a linear waterfall approach, your ultimate aim is to create a quality application or service. You're just taking a different — arguably better — route to that end goal.

With DevOps, it's unlikely that you'll reach



these three criteria on your first, second or even third iteration. "That first project is probably going to fail, or at least it's not going to work as well as you thought it would," Tunnessen said. Therefore, it's a good idea to set incremental benchmarks toward these three objectives.

When he was at the California Department of Health Care Services, Cruz used dashboards "to really manage and measure the ability for those tools to provide the necessary outcomes that we wanted."

"I always say what gets measured gets managed," Cruz said. "It's extremely important that you have data to drive decisions. And I think having dashboards in place gives you the visibility where you're able to assess levels of maturity as you move forward with this approach."

Use the sample worksheet on the next page to get started on measuring the success of your DevOps projects.

DEVOPS

project scoring tools

	1st Iteration	2nd Iteration	3rd Iteration	Final Product
Does it work?	Metric What basic functionality do you want to achieve with your first prototype?	Metric What additional functionality do you want to achieve with your second iteration?	Metric What additional functionality do you want to achieve with your third iteration?	Metric What ultimate functionality do you want to achieve with your final product?
	Score	Score	Score	Score
	Next Steps What do you need to do to improve functionality in your next iteration?	Next Steps What do you need to do to improve functionality in your next iteration?	Next Steps What do you need to do to improve functionality in your next iteration?	Lessons Learned What lessons from this project can you apply to your next one?

Does it work
for the user?

1st Iteration		2nd Iteration		3rd Iteration		Final Product	
Metric What value do you want to provide to product users with your first prototype?		Metric What value do you want to provide to product users with your second prototype?		Metric What value do you want to provide to product users with your third prototype?		Metric What ultimate value do you want to provide to product users with your final product?	
Score	Next Steps What do you need to do to improve usability in your next iteration?	Score	Next Steps What do you need to do to improve usability in your next iteration?	Score	Next Steps What do you need to do to improve usability in your next iteration?	Score	Lessons Learned What lessons from this project can you apply to your next one?

Does it work
for many users?

1st Iteration		2nd Iteration		3rd Iteration		Final Product	
Metric How many users do you want to support with your first prototype?		Metric How many users do you want to support with your second prototype?		Metric How many users do you want to support with your third prototype?		Metric How many users do you ultimately want to support with your final product?	
Score	Next Steps What do you need to do to improve sustainability in your next iteration?	Score	Next Steps What do you need to do to improve sustainability in your next iteration?	Score	Next Steps What do you need to do to improve sustainability in your next iteration?	Score	Lessons Learned What lessons from this project can you apply to your next one?

ACCESS

more resources

After reading this playbook, we hope you feel prepared to bring DevOps to your organization and start making better services more quickly. As you start transforming your culture, implementing automation, measuring your progress, and sharing your success, reference these five additional resources to truly get the most from your DevOps endeavors.

AGILE, DEVOPS, & MORE: HOW TO SUCCEED AT GOVERNMENT PROJECT MANAGEMENT

This GovLoop research guide explores various project management techniques, with added insights from experts across government.

Use it to dive even deeper into project management theories and execution techniques.

DEVOPS DICTIONARY

This digital wiki is a central repository of facts, figures, and discussions regarding DevOps.

Use it as a reference for definitions and taxonomies as you deploy DevOps.

DEVOPS FUNDAMENTALS

This GovLoop online training explains DevOps, its use in the public sector, and when it could be right for your organization.

Use it to educate and engage remote users in the DevOps methodology.

DEVOPS GUIDE

This government service design manual outlines how the United Kingdom uses DevOps.

Use it to show your agency how another public sector organization approaches DevOps.

WHAT IS DEVOPS?

This simple, online explainer video lays out the basics of DevOps.

Use it to communicate DevOps to others in your organization.

About & Acknowledgments

GovLoop's mission is to "connect government to improve government." We aim to inspire public-sector professionals by serving as the knowledge network for government. GovLoop connects more than 250,000 members, fostering cross-government collaboration, solving common problems and advancing government careers. GovLoop is headquartered in Washington, D.C., with a team of dedicated professionals who share a commitment to connect and improve government.

For more information about this report, please reach out to info@govloop.com.

www.govloop.com | [@GovLoop](https://twitter.com/GovLoop)

Thank you to CA Technologies, NetApp, and Red Hat for their support of this valuable resource for public-sector professionals.

Authors:

Hannah Moss, Senior Editor & Project Manager
Francesca El-Attrash, Editorial Fellow
Joshua Hill, Software Engineer

Designers:

Jeff Ribeira, Creative Manager
Tommy Bowen, Graphic Designer
Kaitlyn Baker, Junior Designer
Martin Nera, Design Fellow



1152 15th St NW, Suite 800
Washington, DC 20005
Phone: (202) 407-7421 | Fax: (202) 407-7501
www.govloop.com
[@GovLoop](https://twitter.com/GovLoop)