Progress® | ipswitch®

A PROGRESS PROFESSIONAL GUIDE

The 3 Key Considerations for DevOps Migration

Overview: What is DevOps?

DevOps is the practice of operations and development engineers participating together in the entire service lifecycle, from design through the development process to production support. DevOps is also characterized by operations staff making use many of the same techniques as developers for their systems work. Those techniques can range from using source control to testing to participating in an Agile development process. This document will examine how IT can migrate to DevOps and the three key considerations required to make that migration successful.

Migrating IT to DevOps

DevOps defines a set of processes, tools, practices, and interactions that foster collaboration between developers and operations to achieve high-quality business outcomes. It's a methodology that's become increasingly popular and has been rapidly adopted throughout the business world. If your IT organization hasn't yet begun to adopt DevOps principles into your infrastructure provisioning and application deployment processes, you may already be behind.

Traditional IT vs. DevOps

IT organizations have many processes. Traditional IT operations organizations use processes to twine together interactions between various operations teams. These teams are typically either purposely organized as, or have evolved into, silos – a team of specialists responsible for a specific function of the operations. One team may have responsibility for compute, another for storage, another for networking. Hand-offs between teams may take hours, even days. A single server may take days to weeks to fully provision. This delay frustrates the business owner who is waiting for the system.

DevOps requires a fundamental shift from the traditional IT operations processes. It's about taking the standard roles – "Dev" and "Ops" and melting them together and ensuring that they, along with the business owners, work together as a team. The team is then responsible for the outcome and the speed of the outcome. This can be achieved either organizationally (all the players on the same team) or with shared goals, outcomes, and incentives to deliver a product quickly and with high quality. Automation plays a key role in delivering the services quickly, but it's not the only key.

You may think that your infrastructure deployment process is already automated, and may be also fooled into thinking your organization is "doing the DevOps already". If that's the case, ask yourself how your teams work together, or if they work together. Does everyone on a team (or project) have a common goal? That common goal should be working together to enable a rapid delivery of the software or system, which is enabled by having as many of the processes slowing down the traditional delivery methods being as automated as possible. As one step completes, the next kicks off.

Now have a second look at your processes and think if your server provisioning or software deployment process is completely automated from start to finish. Are there steps in between that require approval? Emails to be sent? Paperwork to be filled out? Files to be copied from one environment to another? It is still acceptable to have approval steps required; in fact, these types of "gates", or "checkpoints" are often required for auditing and compliance. These gates should be present in the process to ensure that previous steps have been completed successfully, but not to block or hold up the deployment. The processes that define these checkpoints or interim steps should also be automated.

Tools for Automation

Tools are an absolute necessity for the automation of the processes, and there are many different tools that can be used for each step of the process. Source control, build systems, testing software, and deployment products are all components that add structure or enable automation, and each type of tool has a few different vendors and options. Give them all a quick evaluation.

Don't get hung up in trying to find a tool that does everything. Acknowledge that a few tools will be needed, and that your talented software engineers may need to add a few ad-hoc items to ensure that the pipeline is completely automated.

For instance, **MOVEit Automation** can automate the transferring of files securely and reliably. If you are moving sensitive data between teams or external sources then you should consider an automated solution.

High Performing IT

DevOps advocates preach concepts such as "fail fast, recover faster". IT organizations are sometimes fearful of DevOps because of the "fail fast" portion of this concept. Failure is deemed unacceptable, and can cause downtime that can cripple your organization. This fear is unfounded, however. In the 2016 State of DevOps Report, statistics show that in a "high-performing" IT company with frequent deployments, the failure rate is 3 times lower and 24 times faster to recover, compared to "low-performing" IT companies who deploy less frequently.

DevOps is All about People

Lastly, success of DevOps relies very heavily on teamwork between the business, developers, testing, and IT operations personnel. Organizing the team in such a way where they are truly a team – having a common set of goals and responsibilities, sometimes even the same management or reporting structure – can help to drive the consistency and focus needed to be successful. Success in a DevOps world relies on teamwork, and it can be a huge cultural challenge for teams that aren't used to working this way.

A large organization, rooted in traditional processes, may view DevOps as a radical shift in the way business is done – and it is! However, the entire lifecycle for an application, and the infrastructure to which it is deployed, doesn't have to be transformed on the first day. Start with understanding what your organization's needs are. What could you provide them that would make their processes quicker? Start with a small, non-mission-critical item, and work with the team to make that process faster. Once the build process and feedback loop are successful, move to deploying that piece with confidence.

IT organizations should begin by building the foundation for rapid deployment – automation, a solid testing framework, a team bonded by common goals, and a decent toolset. All the building blocks must continue to remain a focus to be successful with DevOps and deliver changes to the customers and end-users with the required quality and speed. In order to ensure this, there are three key considerations that must be kept in mind when migrating.

Consideration One: Security

For the past several years, DevOps has been the go-to method for fast software development lifecycles. But just like testing is an integral part of app development, security is an integral part as well. That's why the term DevSecOps was coined.

In today's landscape, apps and services, such as containers and microservices, must be secured since sensitive data is or will be sent through them.

Any seasoned developer or IT person knows that security has always been a part of DevOps mindset, but DevOps has never really been advertised as a methodology that aligns with information security. That is the purpose of DevSecOps, since it incorporates security into the DevOps way of thinking, without making security an afterthought.

What is DevSecOps?

When we think of DevOps, we think of continuous delivery. DevOps is an agile development model and methodology with the sole purpose of streamlining how IT and dev teams interact. However, when we think of DevOps, we should also consider how dev and IT operations teams interact with software testing, or in the case of DevSecOps, security teams.

DevSecOps is the infusion of security into the Agile framework. Rather than having security be a stop gap between software releases, the idea is to have security be a shared responsibility for all members of the software development lifecycle. This includes increased communication between developers, testers, security teams, and operations.

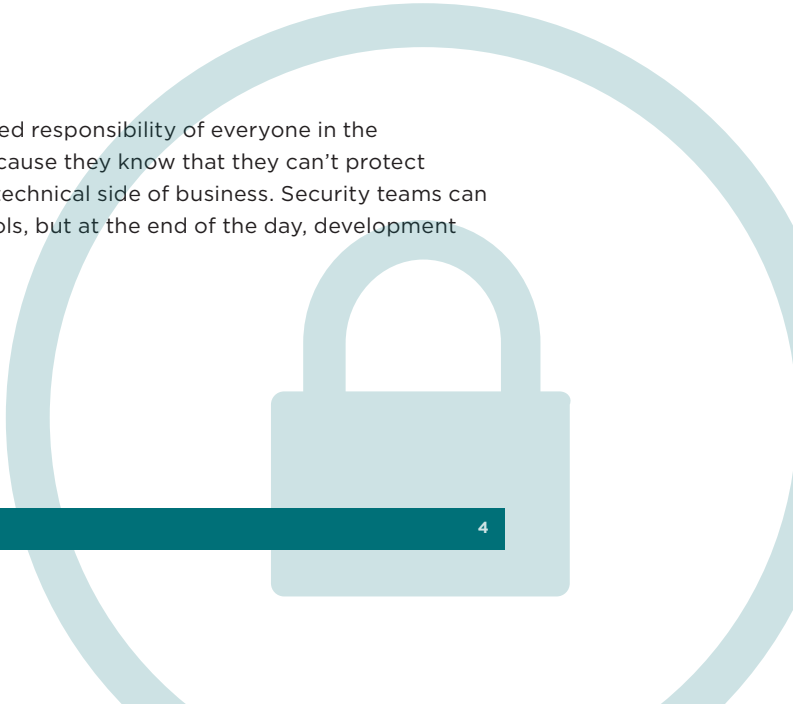
When teams originally grasped the idea of DevOps, security wasn't the first thing that came to mind. That's changing.

"A couple of things have happened in the last couple of years," said Joseph DePlato who is the CTO + Co-Founder of data security firm Bluestone Analytics. DePlato is a professional hacker who has served as a Senior Cyber Security Consultant for companies, such as BP, American Express, Home Depot, and Palantir. "Number one, a couple of massive breaches that have occurred...Equifax comes to mind. As well as a couple of compliance frameworks, policy frameworks that came out, such as the EU's GDPR."

Organizations were on the defensive and security was now recognized to be just as important as other aspects of the software development process. That's why the DevSecOps was coined. There needed to be a way to harness the efficiency of an agile framework with information security in mind.

Security is a Shared Responsibility

IT may be responsible for the network, but security is the shared responsibility of everyone in the organization. IT teams train employees this very sentiment because they know that they can't protect everyone from phishing attacks. The same goes for the more technical side of business. Security teams can speak the gospel on security practices and implement protocols, but at the end of the day, development teams need to consider security in all the code they write.



"Ideally, it's a shared responsibility. So I believe one of the main goals of any security team should be to automate as much as they can, so that it can be integrated into the new development lifecycle," says DePlato. One example of an automated process of security procedures might be 3rd party scanners that can be used to analyze new builds and new code each day. "We recently helped an org integrate a 3rd party scanner into their cycle. So what happens now is, once per day, if there is a new build or new code changes, the code goes through both a static and a dynamic analysis. The output of which the tickets are generated and sent back into their system and added to their current sprint," DePlato said.

The help of automation is key, especially when software development needs to be continuous and quick as is predicated in the Agile and DevOps process. Security is just another cog in the wheel that is agile development. But change doesn't have to be painful for IT, security, or dev teams. With the right mindset and with help from automation, more secure software and at the end of the day a more secure world can be achieved.

Consideration Two: Automation

DevOps is, by definition, all about the joining of development and IT operations, with the goal of delivering applications and services quickly, with continuous updates. To achieve those goals, DevOps teams must use lean methods, aimed at reducing back and forth between development, operations, and customers, ensuring fast feedback through the DevOps pipeline, and ultimately speeding up deployment. Automation is, of course, a key component of those methodologies.

DevOps Starts with Automation

It's simple: If you want your organization to be lean and agile, then you'd better automate. The longer it takes your developers or IT teams to accomplish routine manual tasks, the less flexible your organization is.

Automation is no longer a luxury in the modern enterprise, it's a necessity, and it should be in place even if a DevOps transformation isn't. As the networked world expands, and the borders of corporate networks mesh with the borders of the cloud, and the internet writ large, the number of physical and virtual entities that require management also expand exponentially. To manage all of those resources and tasks manually would be cost prohibitive and virtually impossible, but with automation tools, you are able to amplify the performance on individuals to meet that demand. And for workers, this can be a godsend, as it takes over the dull, repetitive tasks that make up much of their workload, and lets them focus on the business of DevOps.

Continuous Delivery Needs ITs Support

DevOps may mean different things to different companies, or different people, but at its core, DevOps means a company culture that supports and prioritizes continuous delivery.

Continuous delivery is an approach to application development and delivery that emphasizes the ability to deliver application changes and updates at a moment's notice. Necessarily, in a continuous delivery environment, applications must always exist in a deployable state. To achieve this state, development and operations must be agile and efficient—which means automation is essential. But that doesn't just mean automation on the development side.

Efficient IT processes are essential to DevOps, and so, IT automation is critical as well. If IT processes are slow and inefficient, it slows the entire delivery pipeline. Development teams need resources readily available at all times to avoid scheduling conflicts and meet deadlines. IT's primary goal in a DevOps environment is to make sure dev teams don't run into any network, application, and hardware bottlenecks, and to eliminate unnecessary back and forth.

Smart Automation Means Starting Small

Of course, automation is nothing new and most IT teams have automated plenty of tasks before. It's estimated that the vast majority of environments have some level of automation in place, whether it's a simple script or an automated tool like managed file transfer.

However, just because someone has written a few scripts doesn't mean they can spearhead a successful automation initiative. Building useful IT automation that saves time, cuts down on errors, and doesn't require a lot of maintenance is a lot harder than one may think. When implementing an automation initiative it's important not to bite off more than you can chew. Look for a good, low-impact place to start. There's no advantage to upending your entire environment with a company-wide automation mandate—start small, show success, and build from there.

If you implement end-to-end automation on one workflow at a time, you can methodically remove the weakest links from your processes without being disruptive.

Consideration Three: Network Monitoring

Working in a DevOps environment, agility is everything. That next release needs to get into production quickly, so even a minor network issue can hamper speed and efficiency.

Consider the different roles within a DevOps environment. You have your developers and testers deeply embedded in their dev and test environments, rushing to get the next iteration of software out the door. You have your automation experts whose job is to implement automated processes and services that maximize development efficiency on-premise and in the cloud. Then you have your security pros working in unison with developers and testers, ensuring that security is not an afterthought in each release.

All of these roles have equal importance in a streamlined DevOps organization and all of these roles require maximum uptime of hardware, software, and all aspects of the network in general. Everyone cares about IT in a DevOps environment, which is why there are a variety of metrics that every DevOps role will need visibility into the network environment. However, it is generally IT's role to make sure that everyone has access to their designated environments and services they are accessing and that the servers are running. That can be an issue.

There are dozens of metrics that DevOps team members will need to be able to access. If you're a release manager, deployment times will be very important to know to make sure releases get into production in a timely manner. If you're a QA tester, you'll be very interested in seeing how many of your tests are passing and the percentage of false negatives and false positives. These are just a few metrics that different team members look at. As for IT though, IT has the monumental task of monitoring uptime for all applications, services, and hardware such as switches, routers, and servers. Generally, IT is on the hook based on their service level agreements (SLAs). These SLAs will define the amount of uptime vs. downtime of all the above is needed to ensure that a streamlined business can run smoothly.

These are the top three network monitoring functionalities every IT team should have in a DevOps environment.

1. Application Monitoring

Do you know how many applications your company uses on a daily basis? It's likely in the triple digits. Even if your colleagues use only a handful of apps, those apps are mission critical in getting the job done. But as with all technology from hardware to software, those apps depend on the services they use to run on the backend. Software is imperfect, so it is up to IT to make sure those apps run smoothly. That's the reason application performance monitoring (APM) is so important. Your SLAs probably include applications, and if they don't they probably should. Also, consider that you need to monitor applications that your customers interact with. Think about an update server issuing an update in product to a customer. What if the update server goes down or the customer doesn't get a critical security update right away when it is deployed? You're going to hear about it sooner or later.

Maybe there is a bug in production that is causing memory leaks on a customer's computer or maybe there is a database access problem with a mobile app. These are problems that need to be alleviated before they start impacting business operations and especially before they affect a customer using your products and services.

Application monitoring helps you with root cause analysis and pinpoints performance problems so that you can meet your SLAs and ensure a positive user experience with your applications and services.

2. Configuration Management

Configuration management usually affects the internal business operations rather than the customer-facing side, but in rare instances, it can do both. The impact of configuration changes to the network can be significant. They can degrade network performance. They can result in failure to comply with regulatory standards like SOX, PCI, HIPAA, and FISMA. And they can compromise network security.

You will want to make sure that if a configuration changes you can revert back to a previous state if that change causes problems that run amok on your network. This is why IT will want to be able to archive configurations easily within a network monitoring tool.

3. Cloud Monitoring

If you work in a DevOps environment, you probably use a plethora of cloud services from AWS, Azure, or some other cloud provider. Many developers and testers deploy their environments in the cloud for maximum efficiency. It's a lot easier than doing that manually and much faster. And this is why you need cloud monitoring.

Cloud providers aren't in the business of telling you (or least not making it easy to tell) how many resources your business consumes on a monthly basis. And IT usually has to foot the bill when there needs to be an upgrade to bandwidth or cloud resources. This is why monitoring your cloud environment is essential. Not only will it make sure your cloud services are up and running, but you will also be able to track resource usage and prepare to adjust budgets or limit resource consumption in the near future.

These are just a few of the main functionalities IT will need in a network monitoring tool if you are trying to avoid a collision course with your colleagues and customers. The next step is to take a look at the many solutions on the market and find one that best meets your needs (we obviously have **our favorite choice**). If you want some more help choosing, there are a number of buyer's guides available online - **IT Monitoring Buyer's Guide**, **Cloud Monitoring Buyer's Guide**, and **Bandwidth Monitoring Buyer's Guide**.

Conclusion

Every organization is different and every organization has differing requirements, infrastructure and goals. That said, the advantages of migrating to DevOps are simply too great to ignore entirely. Your organization may just be dipping a toe into DevOps, you may have some DevOps capabilities now or you may even have completely adopted it. Regardless, it's never too late to consider these three key issues that are going to impact your DevOps and IT teams. Take a moment to step back and ask yourself if there's room for improvement in any of these three key areas: Security, Automation, and Network Monitoring. There probably is and these should be seen as opportunities rather than liabilities. Remember – the whole point of DevOps is to make your organization faster, smoother, more effective and more successful. Enhancing DevOps will help everyone.

About Progress

Progress (NASDAQ: PRGS) offers the leading platform for developing and deploying strategic business applications. We enable customers and partners to deliver modern, high-impact digital experiences with a fraction of the effort, time and cost. Progress offers powerful tools for easily building adaptive user experiences across any type of device or touchpoint, award-winning machine learning that enables cognitive capabilities to be a part of any application, the flexibility of a serverless cloud to deploy modern apps, business rules, web content management, plus leading data connectivity technology. Over 1,700 independent software vendors, 100,000 enterprise customers, and 2 million developers rely on Progress to power their applications.

Learn about Progress at www.progress.com or +1-800-477-6473.

RITM0050496