DevSecOps and the cyber imperative

Elevating, embedding, and evolving your risk response

TO ENHANCE THEIR APPROACHES TO CYBER AND OTHER RISKS, forward-thinking organisations are embedding security, privacy, policy, and controls into their DevOps culture, processes, and tools. As the DevSecOps trend gains momentum, more companies will likely make threat modeling, risk assessment, and security-task automation foundational components of product development initiatives, from ideation to iteration to launch to operations. DevSecOps fundamentally transforms cyber and risk management from being compliance-based activities—typically undertaken late in the development life cycle—into essential framing mindsets across the product journey. Moreover, DevSecOps codifies policies and best practices into tools and underlying platforms, enabling security to become a shared responsibility of the entire IT organisation.

DevOps tactics and tools are dramatically changing the way IT organisations innovate. And in the midst of this transformation, IT leaders are finding that longstanding approaches for integrating security into new products are not keeping pace with high-velocity, continuous delivery software development. Indeed, in the DevOps arena, traditional "bolt-on" security techniques and manual controls that are reliant on legacy practices are often perceived as impediments to speed, transparency, and overall security effectiveness.

In a growing trend, some companies have begun embedding security culture, practices, and tools into each phase of their DevOps pipelines, an approach known as *DevSecOps*. Deployed strategically, DevSecOps can help improve the security and compliance maturity levels of a company's DevOps pipeline, while boosting quality and productivity and shrinking time-to-market. How? Automation tools execute tasks uniformly and consistently, whereas humans using manual controls can and do make mistakes. At the same time, with DevSecOps, application changes flow freely through DevOps pipelines, giving developers more autonomy and authority without compromising security or elevating risk.

To be clear, DevSecOps is an evolution of DevOps culture and thinking. Rather than disrupting your current cyber agenda, it actually embeds many of the security processes, capabilities, and intelligence learned over the years into your underlying platforms and toolchains. Building on your experience of developing and operating applications, DevSecOps enables you to automate good cybersecurity practices into the toolchain so they are utilised consistently.

The *DevSecOps* trend is only beginning to gather steam. For its 2018 *DevOps Pulse Report*, Logz.io surveyed more than 1,000 IT professionals worldwide about the state of DevOps in their industries. Roughly 24 percent of respondents indicated their IT organisations were practicing some DevSecOps elements. The other 76 percent said their IT organisations either do not practice DevSecOps or are still in the process of implementation.[1]

Notably, 71 percent of respondents feel that their teams currently lack adequate working knowledge of DevSecOps practices.[2] During the next 18 to 24 months, expect that working knowledge to grow markedly as more CIOs and development leaders explore DevSecOps opportunities. Likewise, those with more advanced DevOps programs in place may begin implementing governance, maximising automation, and cross-training both DevOps and cybersecurity specialists with new processes and tools.

Building on your experience of developing and operating applications, DevSecOps enables you to automate good cybersecurity practices into the toolchain so they are utilised consistently.

DevOps' fundamental value is speed to market.[3] Organisations that do not incorporate security into every phase of their development and operations pipelines risk leaving much of its value on the table. Every product you stand up should be a known entity—tested, secure, and reliable. Internal and external users should not have to waste time grappling with cyber surprises, nor should you.

It's time to stop playing the patch management game with security.

In a DevSecOps state of mind

Even as IT organisations began embracing agile development practices over the last decade, many continued to approach security issues in the same incremental, siloed way they had with waterfall.[4] Building on agile's nimble, team-based approach to development, DevOps is now driving dramatic increases in end-to-end velocity. Yet with its heavy reliance on legacy processes and manual controls, security remains a challenge. In many DevOps pipelines, security is still treated as a bolt-on rather than a design feature. This can create pipeline bottlenecks, in part because few developers and system operators have cyber expertise and even fewer cyber specialists possess a deep understanding of development and operations. As a result, DevOps teams and cyber specialists continue to work separately within the pipeline, often slowing progress.

Increasingly, CIOs and DevOps leaders understand that unless these groups work as a unified team to bake security into products throughout the development and operations cycles, their companies may never realise DevOps' full promise.[5]

DevSecOps is not a security trend in and of itself but, rather, an aspect of the ongoing DevOps revolution that *Tech Trends* has chronicled in past issues.[6] It is also more of a mindset than a formal set of rules and tools. DevSecOps offers companies practicing DevOps a *different way of thinking* *about security*. Consider the following characteristics of DevSecOps, and how they differ from the way you are approaching security in your development pipeline today:

- **Open collaboration on shared objectives.** DevSecOps creates shared expectations and metrics for measuring success. It aligns security architects and focuses activities based on business priorities.
- **Security at the source.** DevSecOps features consumable, self-service security capabilities, establishes security guardrails, and makes it possible for teams to monitor results and provide targeted feedback. It can find cyber vulnerabilities early in the application development cycle, reducing the need for rework just before or after deployment.
- **Reinforce and elevate through automation.** By automating recurring tasks, DevSecOps makes it possible to orchestrate an integrated process flow, embed preventative operational controls, and create ongoing audit trails.
- **Risk-oriented operations and actionable insights.** Organisations incorporating DevSecOps into their development pipelines can utilise operational insights and threat intelligence to drive process flow, prioritisation, and remediation recommendations. They no longer have to rely solely upon code scans and can take a more risk-based approach to testing.
- **Holistic approach to security objectives.** Integrated frameworks help secure both the pipeline and application. This helps create a more comprehensive, end-to-end defense throughout the production environment.
- **Proactive monitoring and recursive feedback.** Automated, continuous testing helps identify problems before they become issues. Developers can also leverage logging and telemetry to drive learning and innovation.
- **Automated operations security.** Because visibility into some aspects of operations security can be limited, CIOs overseeing security audits have often found themselves in a position of having to *assume* (hope) that various security administrators have performed their jobs correctly. Security-as-code may offer a more effective approach. New techniques in containerisation and public cloud infrastructure automation now make it possible to audit security and compliance in operations reliably and consistently, with less effort.
- **Operations engineering.** When humans are part of the loop, the process of detecting an intrusion and taking action can eat up precious hours or even days. However, in secure infrastructure-as-code environments in containers or public cloud/containerised environments, engineered response capabilities can automatically and instantly redirect traffic, freeze nodes for later inspection, notify operators, and spin up fresh instances—all automatically.

Taken together, these DevSecOps elements can help improve the overall quality of security, boost productivity, and reduce compliance issues. Importantly, they can break the bottleneck that traditional security creates in high-velocity development environments, thus unleashing DevOps' full potential.

DevSecOps in four parts

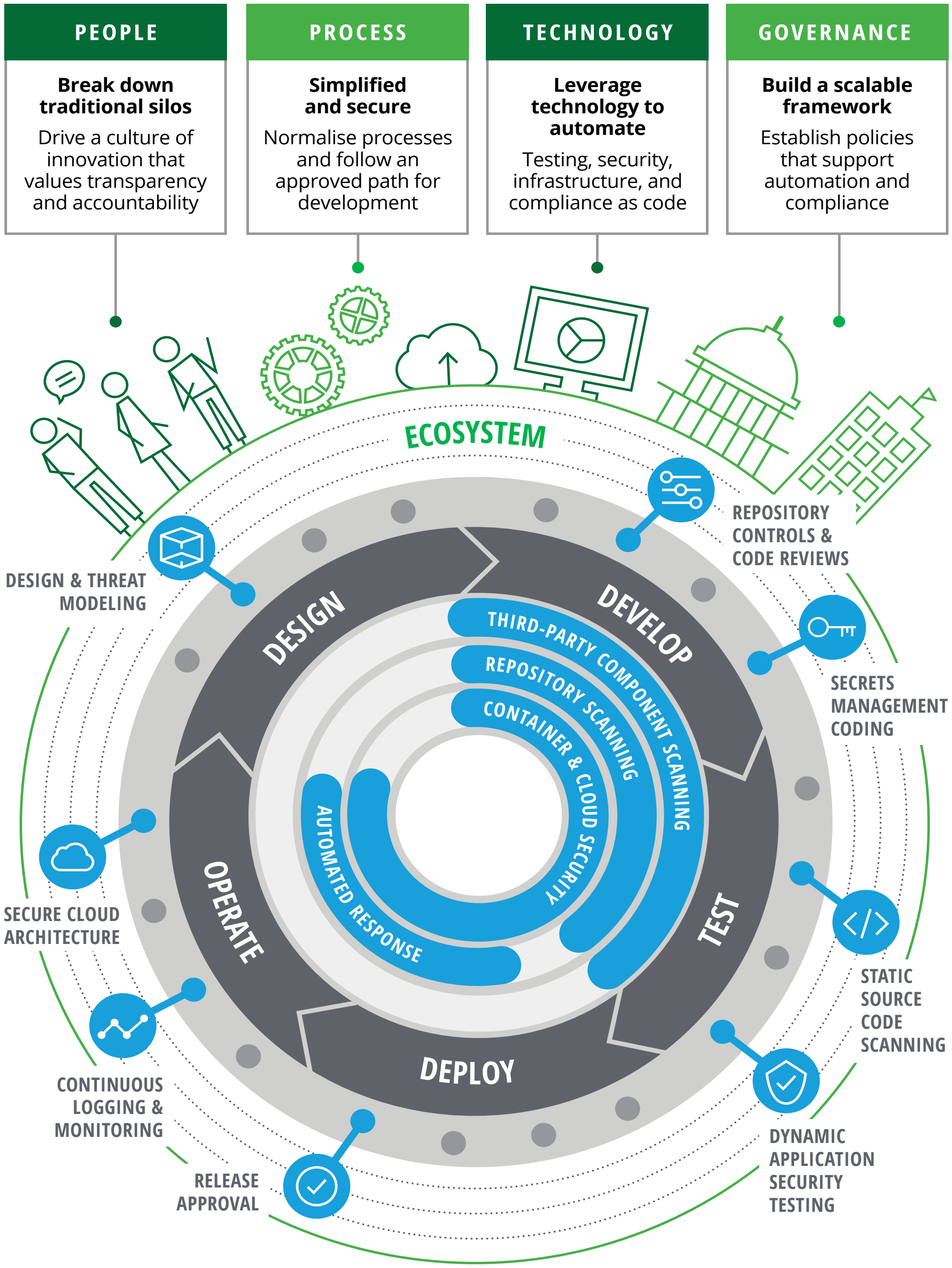
DevSecOps incorporates secure culture, practices, and tools to drive visibility, collaboration, and agility into each phase of the DevOps pipeline. Though companies can tailor their security approaches to support their own cyber agendas and product needs, DevSecOps initiatives typically rest on four foundational pillars:

- **People.** As you integrate security into your DevOps pipeline, remember that people are still your greatest efficiency (or inefficiency) asset. In the traditional waterfall model, the development, security, and operations teams are siloed. As you

FIGURE 1

What is DevSecOps?

It is a transformational shift that incorporates **secure** culture, practices, and tools into each phase of the DevOps process.



PEOPLE

Break down traditional silos

Drive a culture of innovation that values transparency and accountability

PROCESS

Simplified and secure

Normalise processes and follow an approved path for development

TECHNOLOGY

Leverage technology to automate

Testing, security, infrastructure, and compliance as code

GOVERNANCE

Build a scalable framework

Establish policies that support automation and compliance

ECOSYSTEM
DESIGN
DEVELOP
TEST
DEPLOY
OPERATE
THIRD-PARTY COMPONENT SCANNING
REPOSITORY SCANNING
CONTAINER & CLOUD SECURITY
AUTOMATED RESPONSE
REPOSITORY CONTROLS & CODE REVIEWS
SECRETS MANAGEMENT CODING
STATIC SOURCE CODE SCANNING
DYNAMIC APPLICATION SECURITY TESTING
RELEASE APPROVAL
CONTINUOUS LOGGING & MONITORING
SECURE CLOUD ARCHITECTURE
DESIGN & THREAT MODELING

Source: Deloitte analysis.

move into the DevOps world, teams may still operate that way for a while; breaking down those traditional barriers can be the first and most important catalyst to your DevSecOps journey. Try to identify and remedy those silos quickly, create shared goals within DevSecOps teams, and drive a culture of innovation that consists of openness, transparency, ownership, and accountability. While the human resource hierarchy may remain separated, the development culture should be product-based and therefore lead by *product teams*. Each responsible party (dev, sec, ops) owns a portion of the product success.

A positive by-product of DevSecOps is that cybersecurity specialists often develop a greater understanding of development pressures and therefore drive more backend automation of security functions.

It is also important to start small. Small teams gradually come together cohesively; if successful, more and more product teams may start self-adopting DevSecOps practices across the enterprise. As you scale DevSecOps, the product teams will likely become ever more self-sufficient, identify their own security challenges, and automatically course-correct for the benefit of secure product delivery. A positive by-product of DevSecOps is that cybersecurity specialists often develop a greater understanding of development pressures and therefore drive more backend automation of security functions. Likewise, development teams with a deeper understanding of cybersecurity approaches can proactively adopt secure coding practices. The net result in both instances is increased efficiency.

- **Process.** Keeping in mind that speed and quality are key to DevSecOps, try to simplify manual processes as much as possible without sacrificing cybersecurity needs. Since development and deployment are now accelerated much faster than before, security software development processes should become more factory-like. Otherwise, efforts to exponentially accelerate secure software deployments may be unsustainable.

  Consider creating normalised development processes that follow consistent approaches. This is where the security process concept of "shifting-left" becomes important.[7] For example, try incorporating design thinking to understand customers' security needs. Implement threat-modeling storyboards into software changes to build cyber resilience into the application even before the first line of code is written. And incorporate incremental static code scanning into the integrated development environment before the application is packaged. Yes, the shift-left mentality takes a bit of extra effort upfront, but it can help prevent many more breaches waiting to happen—and a lot of product rework. In a nutshell, consider your cybersecurity requirements right away and try to move them as early into the design stage as possible, aiming to eliminate manual security "gatekeeper" delays later on.
- **Technology.** The introduction of DevOps has created a plethora of cloud-based solutions that development teams are using to speed delivery. Fortunately, cybersecurity software is now beginning to keep pace. For example, assorted pipeline tools—testing-as-code, security-as-code, infrastructure-as-code, compliance-as-code, and others—can eliminate the need for some manual security activities, thus boosting velocity. When tools such as these are implemented with the right processes, development and security teams can become more unified, defect costs can plummet, and quality can become consistent

throughout the pipeline. Consider taking an incremental approach to technology deployment, testing these new security tools with specific product teams before releasing to the enterprise.

- **Governance.** The term *governance* is broad by design, but there are two ways to think about governance for cybersecurity in the world of DevSecOps:

  - **At the micro level (the world that revolves around the product teams).** Embedding cybersecurity into DevOps can boost efficiency in governance. How? DevSecOps, by design, requires a highly consistent process that uses a uniform set of tools and automated controls. This helps simplify the monitoring and testing of required controls. In fact, by designing DevSecOps processes to accommodate the needs of compliance and control teams, you may be able to gradually automate testing processes and free up developer resources. The process of pulling a list of tickets, selecting samples, and identifying all relevant audit trails from multiple systems might have taken *days* of a developer's time. Using compliance-as-code, it can be accomplished in minutes.
  - **At the macro level.** DevOps has transformed how IT organisations work. In some companies, IT operations—traditionally comprising a mix of senior management, management, and engineers—is moving to a flatter hierarchy made up of fewer management positions supported by architects and engineers. At the same time, penalties for running insufficiently governed IT environments have grown. This means that the overall governance of the *projected* IT landscape is more important than ever before. The success of your company brand increasingly depends on products developed using DevOps.

Like any other IT program, DevSecOps should directly tie to your broader IT strategy—which, in turn, should be driven by your business strategy. If a DevOps program supports your IT and business strategies, then embed the "Sec" at the same time. In short order, it may help you bolster your cyber maturity posture and save you from having to rework your DevOps program later when it's much harder to do.

LESSONS FROM THE FRONT LINES

NOTHING TO SNEEZE AT: NIAID PRIORITISES CULTURE CHANGE IN ITS DEVSECOPS TRANSFORMATION

LESSON ONE

The National Institute of Allergy and Infectious Diseases (NIAID) works to keep us all safe by conducting and supporting research to prevent infectious, immunologic, and allergic diseases. Within NIAID, the IT organisation is working to "future-proof" itself and provide timely and secure support to researchers and staff who conduct and manage key research projects. While the agency has used DevOps to ensure faster delivery of its software solutions, its need to protect sensitive health data has resulted in a vision for automated security everywhere and led to DevSecOps—the next logical step to DevOps.

"I think of DevSecOps as three legs of a stool: management practices, technological practices, and cultural practices," says Joe Croghan, chief of NIAID's software engineering branch.[8] "The cultural piece is the most challenging, though: You ask teams for transparency, to admit mistakes, and to change continuously; it can be a lot for people to put their arms around." Croghan believes the change is vital to ensure continued productivity in the face of rapid change, and that it has enabled his team to continue to quickly respond to requests with secure products.

Long software release cycles were causing bottlenecks in delivering technology solutions at NIAID, compounding the existing challenges of the rapidly changing security landscape. Implementing DevOps practices—continuous integration and continuous delivery, automated testing, and infrastructure-as-code—has helped shorten the lead time to deliver software and to patch critical defects. Infrastructure-as-code practices reduce vulnerabilities by making some aspects of security, such as application and server configurations, inspectable. And integrating security scanning tools like Fortify into the DevSecOps pipeline stops coding vulnerabilities from getting to production in the first place.

"The challenges we've always had with security are consistency, predictability, and putting security policies into a systematic framework," Croghan says. "By implementing a DevSecOps approach, we can run scans and put specific, consistent security protocols in place. When we're using these techniques, we can be very confident in what our servers look like, and if there's a problem, we can fix it consistently by changing the code."

In the coming year, Croghan hopes to address some of the cultural and management changes that are crucial to sustaining the team's DevSecOps momentum. Staff and customers now see the value of the new approach and have been pleased with the speed of new application deployment, with the software engineering team completing more than 250 automated deployments in a month. But Croghan aspires to change the culture and do much more. "I think a year from now we will continue to adopt new technologies," he says, "but we need to change the way we work. The culture of DevSecOps is to constantly measure, reevaluate, and change." These changes include aligning behaviours by educating his staff on delivering secure code within a DevSecOps framework and exceeding his customers' expectations of software delivery, security, and velocity.

THE FDA'S PIPELINE DREAMS

LESSON TWO

Security and safety lie at the heart of everything the US Food and Drug Administration does. Each day, the agency's 17,500 employees work diligently to ensure the safety and efficacy of the United States' food supply, pharmaceuticals, medical devices, cosmetics, and more. Amid recent calls for the agency to accelerate the process by which approvals occur, teams throughout the agency are working to strike the right balance between speed and safety.

Given the criticality of this mission, the FDA needs to support the security, privacy, and stability of its IT systems at speed. To this end, the Center for Biologics Evaluation and Research (CBER) has launched an ambitious DevSecOps initiative to reengineer its approach to security throughout the product development process. Although the project is still in the early stages, its goals are clear: 1) build in security upfront rather than treating it as an afterthought, 2) automate as much as possible, and 3) transform the agency's development culture into one that emphasises agility and speed.

According to senior IT project manager Christopher Kiem, DevSecOps represents a significant opportunity to get everyone working on the same page from the start of every project. "On day one, we want our operations talent to provide security insights and guidance to our developers in what we hope will become a loop of continuous conversation in which everyone is learning from each other," he says.

This project loop will include inputs from security automation tools as well. A static code analysis tool will scan the source code for security issues. Application scanners will review open-source library files for security issues. Upon detecting problems, all of these tools will open issues for developers and DevSecOps engineers to assess and resolve.

These and other DevSecOps tools will streamline the overall development process and accelerate the pipeline. "When project managers come up with new system requirements, critical development tools, processes, and automation will already be in place," Kiem says. "This will make it possible for project managers to make decisions quickly. Our goal is to eliminate the meetings, emails, and back-and-forth that slow people down."

Currently, the CBER is performing a modernisation analysis to identify the different pieces and parts—data standards, regulatory rules, submission types, stakeholders, among others—to include in a formal game plan. This plan will also identify the DevOps elements already in place that can be leveraged. In the coming months, CBER's IT leadership will present the plan to the Center's management to solicit their input and secure their sponsorship. "Once we have backing for the project, we will begin assessing our technology needs and developing plans for putting a pipeline in place," Kiem says. "We will also be working closely with our enterprise IT partners to design a DevSecOps architecture that fills any current infrastructure gaps and supports our priorities."

The priorities to which Kiem refers are not limited to software development and enhanced product security. Indeed, with DevSecOps he sees a tangential opportunity to reengineer core systems and, in doing so, keep the agency's costs in check. "I think that across the public and private sectors, there are opportunities to decrease expenditure on IT. As you reengineer development processes to enhance security and quality, you can use this opportunity to consolidate your technology footprint. When things are humming along in a well-developed pipeline and you are releasing the products your users want, you should no longer need your less secure legacy systems, the massive suites of tools, and the time required to support and enhance them. Imagine being able to get rid of all those things—and the costs related to them."[9]

MY TAKE

ADAM BANKS, CHIEF TECHNOLOGY AND INFORMATION OFFICER AND CHIEF DIGITAL OFFICER, MAERSK

Maersk, like many other industrial organisations, has become digitally dependent—for operational efficiency and as the driver into new products, offerings, and markets. Maersk has always been a forward-looking business, but we have a heightened focus today in part because of a global cyberattack in 2017 that infected our network across ports and offices across dozens of countries. As part of the recovery, we rebuilt our core IT capability, including reconstructing server and network infrastructure, moving more than 60,000 devices to a new common standard, deploying global operating system upgrades, restoring our entire application stack, and restarting the world's most automated terminal, all in a matter of weeks. We now have one of the most standardised environments of any company in the industry—a foundation that's letting us deliver change at the pace of digital business.

Given the ever-changing cyber landscape, we're building an even more secure and reliable infrastructure that can support the future growth of Maersk. We're focusing on automated toolchains, building relevant static and dynamic scanning processes into our continuous integration and deployment processes. We've adopted post-deployment monitoring through production, and we've been able to go from writing a line of code to deploying it in production with no human touch. That's posed some interesting challenges across the organisation: When do you do a product release? With such rapid churning and changing, at what point do you declare it a new version? Currently, we're spending a good deal of time exploring these concepts, making DevSecOps a core area of interest.

We've asked our CISOs to identify the gaps we have in our infrastructure as well as the compensating controls available to address those gaps. One of the main things we've done in the last two years is move the governance of risk from a central corporate function to a CISO function, so the CISO makes policy as well as enforces it. I want them to kick down the door where there's an area of the business that "doesn't have any risk," because that's just not possible. CISOs work with business owners to make deliberate decisions, and business owners can decide how to address the existing risks when they are contained within their functional geography. It's a consultative approach, but it's consultation with teeth.

To that end, our CISO may not be a standing member, but there's not a quarterly audit committee meeting where he's not on the agenda. At our supervisory board, every other update has some cyber topic associated with it. We show the board a funnel diagram representing the number of attacks on the external surface, the penetrations, all incidents, and then the major incidents—not to show them what we are doing but, rather, to demonstrate that our processes are working. We want them to understand that if the external surface attacks go from 200 to 800 a week, they should be asking us questions; if they see an increase in those that are penetrating, we want a dialogue to ensue about how we can handle the uptick. We want non-technologists as well as IT leaders to understand there is a minimum level of control and recoverability that should be in place if and when we fail to stop a future attack. With their support, we can control the amount of damage done and speed our recovery.

In this environment, I don't think it's an either/or approach when it comes to traditional waterfall development, DevOps, integrated toolchains, and agile delivery. We still organise our people under traditional plan-build-run structures, with functional homes organised around technology or IT life cycle capabilities. But we deploy our people in a DevSecOps model aligned to products, platforms, and initiatives. This enables all areas of the business to gain from improvements in any area, across all activities. Without a functional home for people to return to, you're constantly churning people and process, which means you're failing to improve them each time. For example, I don't want every one of my global teams solving for automated regression testing.

So we implemented a center of excellence that provides team members the tools, thinking, and models they need to complete their tasks. This model has allowed us to increase in maturity and capability, while deploying applications in a more modern, diverse fashion.

This model only works, however, if everyone around the leadership table understands the inherent value in the technology organisation. Maersk is a digital business, and we are incapable of operating if the technology does not work correctly, so our business leaders need to understand what's at stake. You need a degree of trust, openness, and desire in all the players around the table to play as a single team. I knew we'd achieved that at Maersk when I proposed a reduction to the technology budget and my peers argued against the notion, fearing we'd miss out on too much value. I think that's the target for which we're all aiming for: complete understanding of how security and DevSecOps can impact business outcomes.

My goal is to have anyone around the executive table capable of leading the technology function in a few years. This would reflect that operations and the underlying technology stack have stabilised, and that the business leaders are tech-savvy enough to step in and lead the charge. But the real test will be if they strive to take roles in which technology enablement responsibilities are as recognised and as important as leading sales or a line of business. We're well on our way.

MY TAKE

WES HUMMEL, VP OF SITE RELIABILITY ENGINEERING, PAYPAL

For PayPal, with more than 254 million active account holders and over 7.5 billion payment transactions in 2017, security and trust are central to everything we do and what our customers expect. As such, we treat security as a strategic business priority and a fundamental part of how we develop, release, and maintain our product code, integrating it by default into every layer, throughout the entire development life cycle.

For me, DevSecOps means not only empowering our developers with the tools necessary to develop high-quality, secure software, but also creating a culture that builds secure products by default. With our company, customer base, and transaction volume growing so fast, we need security at scale: As of 2017, PayPal had 4,500 developers, 50 million lines of code, 1 million builds per month, 2,600 apps, nine availability zones, 230 billion hits, and 42,000 batch executions per day! We give developers as much control as possible over their code and its outcomes to help them achieve this scale. When you offer developers flexibility and autonomy, it is important to build a talent base that lives and breathes your security mantra. We've worked to create a culture in which developers understand that successful products require an equal appreciation of development, security, and operations. That's been our journey: to meet security, availability, and quality needs while enabling high-velocity code releases.

We adopted an agile methodology at the start of our journey, and we're currently transitioning to DevSecOps. We try to find balance between development and operations by providing tools that make every step—from ideation to releasing code—frictionless for developers. We empower them with the freedom to use our recommended tool suite, the "opinionated path," which includes security penetration testing, auto-enabled security controls, threat modeling, automated scanning, and other features. But we also believe developers shouldn't be forced to use a specific tool suite, so we give them the autonomy to follow an un-opinionated path and bring their own stack. We provide the tools and processes they need to deploy code while meeting our security, availability, and quality standards. This way of working under a DevSecOps framework is resulting in better performance and productivity for our developers. We are also seeing a reduction in potential vulnerabilities and improvements in maintaining product security standards—a result of ingraining risk-based thinking and processes within the DevOps pipeline.

Our primary focus now is to stand up an autonomous fleet of development, operations, and security tools that we can maintain virtually hands-free. There's tremendous value in being able to run scans and end-to-end tests on a minute-by-minute basis, deploy patches through automation, identify potential vulnerabilities at regular intervals, and ensure that applications are meeting standards, including configuration changes or vendor interface updates in production. Automating these processes is key for scaling a 200,000-plus-node fleet with speed and consistency while holding all deployments and processes to the same security and quality standards.

Our security and compliance automation have been helpful in related areas, as well. Our focus on automation has made it easier to address the complexities of legal and compliance obligations and policies, given that we operate in more than 200 global markets. With the nature of our business, security and trust will remain a core capability and priority for PayPal. It doesn't matter what size company you are—if you build in security at your core, it will serve your business well.

ARE YOU READY?

Embedding security into DevOps pipelines may initially seem like a straightforward proposition. After all, if DevSecOps is just a way of thinking about security, then deploying it in your DevOps factory should be a light lift, right? For those few who have fully mastered DevOps, perhaps. For everyone else—and that is most organisations—developing DevSecOps practices will likely be another component in existing DevOps initiatives that are still in early stages. For example, in its *2018 Global Developer Report*, GitLab surveyed roughly 5,300 IT professionals about their DevOps experiences. Thirty-five percent of respondents said the DevOps culture at their companies was "somewhat established." Only 23 percent of those surveyed would go so far as to describe their development method as DevOps.[10]

As you explore DevSecOps opportunities, ask yourself the following questions not only about security but about how they may affect your current DevOps efforts.

▶ **Will I have to hire developers with security expertise?**

Not necessarily. First, work to turn the combined knowledge of the security expert and the developer into code. Next, upskilling existing talent may be the only viable staffing option as the *DevSecOps* trend progresses, but it allows you to retain important business knowledge gained over the years from each respective area. Besides, developers with security expertise (and vice versa) are in high demand right now and increasingly hard to recruit (and keep).[11]

▶ **Won't DevSecOps slow down my pipeline?**

Probably not. Granted, if you had no security controls prior to DevSecOps, there will be some efficiency trade-off, but DevSecOps provides two major efficiency benefits: 1) Incorporating security into a DevSecOps pipeline still results in a faster pipeline than the waterfall method, and 2) DevSecOps gets faster as time moves forward because vulnerabilities are mitigated over time and efficiency increases. Developers also gradually gain more freedom and autonomy to move product through the pipeline because of automated controls.

▶ **Can DevSecOps be compatible with my compliance requirement?**

Yes—if anything, it helps ease the burden of maintaining compliance. In an ideal DevSecOps state, security auditing, monitoring, and notification are fully automated and continuously monitored, enhancing compliance.

My DevOps process is still immature. How can I make sure that my DevSecOps governance is scalable?

Plan, storyboard, and start small. Sustainable and scalable DevSecOps governance models typically feature the following components:

- Clearly defined roles and responsibilities in all *cross-functional* teams
- DevSecOps-specific policies and procedures that enable organisations to keep up with the pace of application development in a DevOps environment
- Automated security tools throughout the pipeline that reduce vulnerabilities and the lower the frequency of human error
- Security monitoring and notification systems in DevSecOps that create automated audit trails throughout the software development life cycle—which, in turn, facilitate compliance reporting
- Continuous monitoring of security metrics, which helps DevOps teams constantly improve their security decision-making

BOTTOM LINE

The ever-growing need to get quality products out the door faster has elevated DevOps practices to the position they hold today in the arena of software development. In a natural extension of DevOps evolution, the DevSecOps trend offers CIOs and their development teams a new mix of tools, practices, and automation that, deployed in concert, can help secure development and operations.

Contact

AARTI BALAKRISHNAN
Director, Technology Consulting
Deloitte MCS Limited
+44 20 7303 7005
abalakrishnan@deloitte.co.uk

Authors

VIKRAM KUNCHALA

Principal, Deloitte LLP

KIERAN NORTON

Principal, Deloitte LLP

MICHELLE SHUTTLEWORTH

Managing Director, Deloitte Consulting LLP

DYLAN HACK

Senior Manager, Deloitte LLP

Endnotes

1. Logz.io, “The 2018 DevOps pulse,” 2018.
2. Ibid.
3. DevOps Research and Assessment, *Accelerate: 2018 State of DevOps*, August 29, 2018.
4. Mark White et al., *Right-speed IT*, Deloitte University Press, February 24, 2016.
5. Warwick Ashford, “Firms need to move from DevOps to DevSecOps, says expert,” *Computer Weekly*, March 20, 2018.
6. Ayan Chatterjee and Alejandro Danylyszyn, *Real-time DevOps*, Deloitte University Press, February 21, 2014.
7. Chris Riley, “The how and why of shift-left security,” Twistlock, May 31, 2017.
8. Interview with Joe Croghan, chief of NIAID’s software engineering branch, November 14, 2018.
9. Interview with Christopher Kiem, senior IT project manager, US Food and Drug Administration, November 19, 2018.
10. GitLab, “2018 global developer report,” March 7, 2018.
11. Ashford, “Firms need to move from DevOps to DevSecOps, says expert.”