Five keys to an effective DevSecOps framework

Cross-functional collaboration and automated controls integration are vital across the board

kpmg.com

DevSecOps

Where are we now?

Reducing risk across the software development lifecycle (SDLC) today is not well understood across many software organizations. We believe a collaborative, multidisciplinary DevSecOps framework should be standard operating procedure for facilitating rapid, compliant, and safe service delivery.

But what's required end-to-end across an organization to do it right while simultaneously managing conflicting priorities?

Ask a CIO, CRO, or CISO what it means to carry out DevSecOps effectively and the typical answer is "We have a plan for that." They likely do, but in many cases each has different priorities and perspectives. Unfortunately, there's no single sheet of music from which all three are reading. As a result, too often there is no harmony, no agreement as to how these constituencies can most effectively work together.

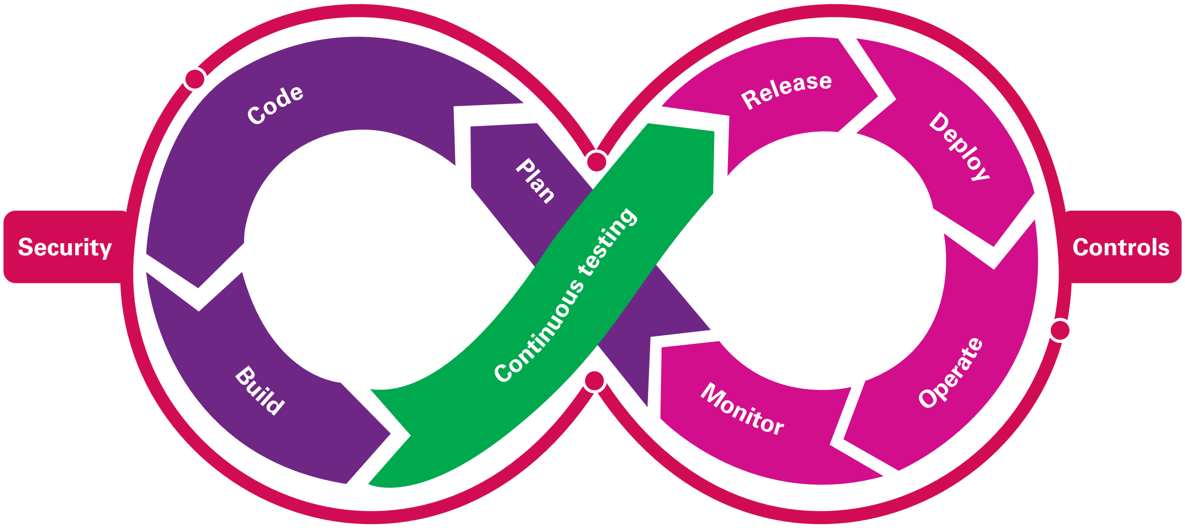
A well-designed DevOps framework is predicated on increasing delivery speed and customer value through an automated SDLC. Many companies are adopting this approach in an effort to establish themselves as leaders in a steadily digitizing economy. This initiative is a critical priority for many organizations and requires broad leadership support.

With vulnerability concerns growing, companies need to embed security into the SDLC holistically so development teams can work quickly and safely at scale. The challenge for highly regulated and non-regulated organizations alike is working safely and securely without interrupting innovation and creativity. Companies for which this is not a priority are likely to experience ongoing release delays, application instability, and high levels of cyber risk.

Companies need to align all relevant areas—IT, risk, information security, technology, data, and privacy—to formulate a cohesive, end-to-end vision. Today, data breaches, software failures/outages, and cyber-attacks are often revealed on social media before most legacy monitoring platforms pick them up, making availability and reliability more important than ever. The true financial and reputational cost of recovering from a negative event is elusive, yet massive.

We believe companies should take a holistic view of DevSecOps, prioritizing speed and agility while simultaneously implementing a comprehensive governance framework characterized by a suite of relevant controls, security scanning, and automated testing. Through this approach, organizations align development, security, and risk/operations to create an optimized software delivery architecture that identifies key roles, processes, and technologies. Embedding security and governance tooling directly into the development and delivery pipelines—shifting left—negates the possibility of developers circumventing these controls.

Transitioning from DevOps to DevSecOps



DevSecOps enhances the traditional DevOps model through the cross functional collaboration of the CIO, CISO, and CRO, ensuring that security and controls are integrated through the SDLC to mitigate risks and deliver software faster.

Imperative #1

Remove barriers from the development team’s path

Developers want to work within a fully automated pipeline, where they can write elegant, game-changing code. Their primary objectives are to increase the speed and agility with which they write and deliver software in order to drive value both externally to customers and internally across the organization. They have long viewed as speed bumps the legacy manual processes that companies have in place to govern and control their environments.

What’s the implication?

Simply put, many developers think they don’t need governance or change-management initiatives. However, as any CRO will attest, they're not the ones that get the email or phone call when something fails operationally or there’s an outage. They just want to be able to write code, unhindered. This is nothing new, but in an environment marked by ever-increasing changes in the realms of people, processes, and technology this stance is in conflict with all the considerations that support an efficient, effective SDLC.

As demand for new features and functionality grows, development teams must accelerate their work. To that end, developers have implemented highly automated CI/CD pipelines so they can develop, build, test, and deliver software quickly. In many instances, shortcuts are taken to circumvent governance in the interest of speed. Developers may desire autonomy, but they've got to realize that achieving that state cannot come at the expense of security and compliance. And, of course, sidestepping governance—inadvertently or not—exposes the organization to risks that might otherwise be avoided.

What can companies do about it?

Companies shouldn’t have to choose between speed and security. As mentioned above, companies can achieve a fully automated DevSecOps pipeline by shifting left a number of security, governance, and change-control mechanisms.

Leadership can make it easy for development teams to do the right thing automatically. By embedding the relevant controls as architectural enablers directly into the CI/CD pipeline from the start, they can ensure governance and compliance across workstreams. This approach will enable developers to operate at full speed without exposing the company to increased risk and incurring regulatory penalties. Happy developers contribute to happy auditors, which means happy customers.

Imperative #2

Give information security, governance, and compliance seats at the table from the outset

Managing security in a cloud-native, highly automated DevOps environment has become one of the great challenges among companies that develop and distribute software. Security teams have the painstaking job of ensuring that companies avoid negative headlines. They are the unsung heroes who work to protect sensitive customer data and limit the company's exposure to hackers and other bad actors.

What's the implication?

Every day the CI/CD pipeline grows more complicated as development teams gain more visibility and control. In many cases, Information Security professionals at many companies are working with developers looking to make hundreds, sometimes thousands, of production changes every day. In years past, there was often only a handful of changes every several months.

When companies are racing to release new software and application upgrades, security controls are often bypassed, leaving companies vulnerable to breaches or substantial outages. More and more cyber incidents are happening across numerous industries—think SolarWinds, Colonial Pipeline, and JBS, to name just several—creating significant disruptions for these organizations and their customers.

What can companies do about it?

We believe building enhanced security and governance controls directly into the processes developers use to rapidly develop and release software is both a worthy and feasible goal. These controls can be strategically embedded into existing development pipelines, without interrupting or slowing down the CI/CD process. By automating security scanning, controls, and testing to the same degree developers have automated their environments, you're essentially giving security teams native control of the pipeline.

Giving security this level of control will help ensure development teams are able to innovate rapidly and deliver new features and capabilities without sacrificing safety. CISOs will sleep better knowing that development speed and agility is not exposing the company to undue risk.

Imperative #3

Empower operations to better support what developers build

In its ideal state, DevOps should be supported by a single, cross-functional team working toward a common objective. However, with developers under increasing pressure to deliver code faster and faster, team priorities are too often misaligned. When problems arise company leadership leans on Operations—i.e., Risk—to assess and repair the damage, with an eye toward maximizing uptime and maintaining reliability through a suite of IT Service Management (ITSM) controls.

What's the implication?

In the past, legacy non-automated ITSM strategies were effective because the developers' change rate was so low. Typically, it was very common to hear that a company would only release new software every six to nine months. In the current environment, companies are releasing exponentially more application changes on a daily basis. Today, development teams typically operate in a very rapid, iterative fashion. Even well-seasoned operations teams are having trouble provisioning capacity and governance at a similar pace.

This is creating significant challenges for legacy ITSM controls. Often, the common practice is to provide development teams with "pre-approved changes" or other solutions that seek to bypass controls; the hope is that these changes won't disrupt anything that is already in production. Unfortunately, most developers are unaware of the pipeline's end-to-end vulnerabilities because they are focused on writing code for a specific area or purpose. Problems arise when portions of code that ran fine on the developer's computer become unstable in production.

What can companies do about it?

We believe companies can keep up with the development team's increasing push for speedy releases by automating operational functions, such as complying with the relevant ITSM controls. The key to success is to implement automated ITSM controls for change and release management, gather data and draw insight, and then make strategic, policy-driven decisions to automate governance. In this regard, a fully automated CI/CD pipeline includes not only automated security controls, but automated Information Technology Infrastructure Library (ITIL) controls.

Maintaining ITSM controls within a fully automated site reliability engineering (SRE) model will enable Risk to keep pace with developers as they work to ensure maximum reliability and uptime—two primary customer priorities. The result is a highly efficient CI/CD pipeline along which code is built, testing is conducted, and new or updated applications are safely deployed.

Imperative #4

Visualize value across the pipeline by focusing on value streams

Executives in an evolving digital environment must manage with a customer-centric point of view, which enables them to understand how well they're creating value. Of course, it's also important to identify where issues are arising along the delivery supply chain. Clearly, the whole purpose of DevSecOps is managing effectively across all customer value streams.

What's the implication?

In the new, ever-changing digital world, customers expect to see value quickly, transparently, and seamlessly. Companies that are unable to provide a product or service as fast, or faster, than the competition are likely to lose customers. In theory, this is far simpler in a digital environment than in traditional business models, yet difficult to execute.

We believe companies can improve the way they manage software delivery by adopting the principles, tooling, and procedures of Value Stream Management (VSM), a focus that extends across all segments of the digital operating model. VSM seeks to ensure transparency, quality, and continuous improvement in the SDLC.

What can companies do about it?

By leveraging tools and platforms that provide deep insights and analytics across all delivery pipelines, companies can make decisions based on the real-time data they extract from existing applications, as well as from customer feedback.

Value streams track the flow and efficacy of new code to customers. VSM helps facilitate the identification of areas or tasks that will deliver value in the form of faster releases, more efficient operations, and overall security. Prioritizing VSM enables companies to maximize the value they provide while improving customer satisfaction.

Imperative #5

Make finance a prominent partner with a dynamic funding model

CFOs play a major role in the organizational quest for a DevSecOps framework. The way programs, projects, and products are funded needs to change as dramatically as the way development, security, and operations processes are evolving. How various aspects of the SDLC are capitalized, operationalized, and reassessed for ongoing viability must be flexible.

What's the implication?

The nature of software development is at once iterative and fluid. CFOs need to be able to alter funding dynamically if developers pivot or the scope of a product or project changes based on customer feedback or market conditions.

The finance department must be able to modify the way they fund projects or products in real time. Without this flexibility they will be left with a chaotic scenario as they attempt to determine where dollars are going and why and quantify ROI.

What can companies do about it?

We encourage companies to design and implement a dynamic investment strategy that empowers CFOs to apply their investment strategies as quickly, and with similar agility, as developers write and release code. Transitioning from traditional budgeting to a dynamic investment philosophy enables companies to transform their funding governance, update capitalization policies, adopt leaner business cases, and evolve their financial analytics capabilities.

With this financially flexible point of view, if a project falters or fails to deliver value over a period of time, the nature of the dynamic funding mechanism allows for the finance team to quickly alter the investment and transition to a different solution. It's about funding value rather than funding projects, and solidifying a culture of experimentation, as well as building scale and sharing success across the value chain.

Dynamic funding inspires IT organizations to think like a venture capitalist—focusing on continuous learning, financial viability, and potential outcomes. In this way, mature companies can reflect how technology start-ups have been operating for years to fund their innovation efforts.

The KPMG perspective

In increasing numbers, companies are experiencing outages, failures, and high-profile cyber-attacks primarily because their DevOps delivery chains suffer from a lack of collaboration and security. Our solution for controlled software releases through automated engineering and governance is based on cross-network collaboration to ensure modern service delivery models effectively apply security policies and controls in an effort to enable clients to maintain speed-to-market without compromise. To achieve that goal we believe IT, security and risk leaders must align their priorities.

Clearly, it’s a mindset shift for all stakeholders, but we believe companies that embrace this approach to software development will be better able to innovate, invest, drive value, and measure bottom-line impact.

A fully integrated DevSecOps structure is intended to increase value from development and delivery perspectives quickly, while mitigating the ever-increasing vulnerabilities and cyber risks that exist in the marketplace today. Our DevOps professionals are ready to help you embed security and governance as key components of your SDLC framework, while maintaining development speed and agility.

About the authors

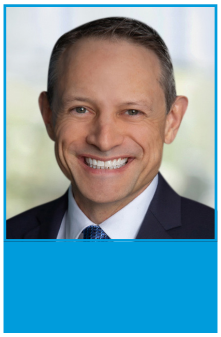
Lavin Chainani

Director

Technology Risk Management

T: 410-949-8834
E: lchainani@kpmg.com

Lavin has 15 years of experience in Technology Risk Management. His professional experience includes IT risk and assurance services where he specializes in IT Internal Audits, System Implementation Reviews and building IT Compliance teams with focus on technology controls and compliance. Prior to joining KPMG, Lavin worked as a network administrator.



Caleb Queern

Director

Cyber Security

T: 512-320-5104
E: cqueern@kpmg.com

Caleb has more than 15 years of technology and information security experience. Prior to joining KPMG, Caleb served as the Chief Scientist for a cyber security services firm based in Northern Virginia. He focuses on reducing risk and improving performance so organizations can meet business goals and grow. Caleb has led engagements with companies in the technology and pharmaceutical sectors, as well as higher education. His specialties include DevSecOps and application security, security operations, and operational excellence—all with an eye toward ensuring the connection between Information Security, IT Engineering, and Risk.

James Williams

Director

Modern Delivery

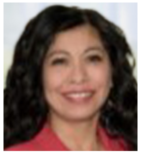
T: 214-840-4822
E: jameswilliams@kpmg.com

As the KPMG Modern Delivery solution leader, James integrates Agile principles, DevOps approaches, product management methodologies, and automated value chain functions to create collaborative cultures that drive outcomes. With more than 20 years of global management consulting and business experience, he applies his advanced knowledge of application development, cloud, and infrastructure automation capabilities to define and execute strategies that break down traditional silos and address critical operational challenges while enabling the successful delivery of new and improved client services.

Contact us

Marcus Murph
Principal
CIO Advisory
T: +1 214-840-2671
E: marcusmurph@kpmg.com

Kyle Kappel
Principal
Cyber Security Services
T: 949-431-7359
E: kylekappel@kpmg.com



Cyndi Izzo
Principal
Technology Risk Management
T: 617-988-5613
E: cizzo@kpmg.com

Eric Ledyard
Managing Director
CIO Advisory
T: 904-354-5671
E: eledyard@kpmg.com

Deepak Mathur
Managing Director
Cyber Security Services
T: 408-367-7676
E: deepakmathur@kpmg.com

Some or all of the services described herein may not be permissible for KPMG audit clients and their affiliates or related entities.

kpmg.com/socialmedia

The information contained herein is of a general nature and is not intended to address the circumstances of any particular individual or entity. Although we endeavor to provide accurate and timely information, there can be no guarantee that such information is accurate as of the date it is received or that it will continue to be accurate in the future. No one should act upon such information without appropriate professional advice after a thorough examination of the particular situation.