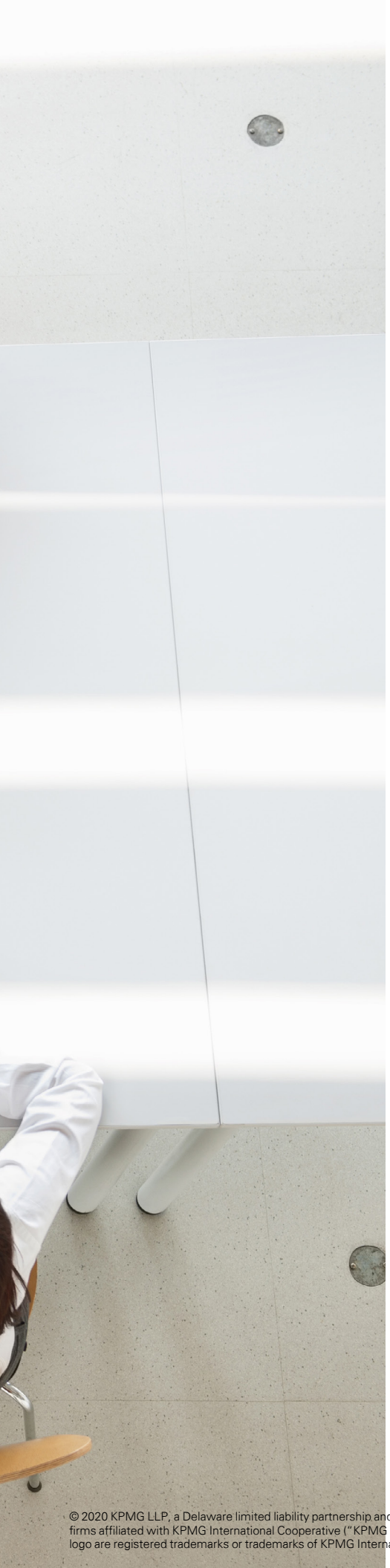
Role of internal audit in DevOps

January 2020

kpmg.com



Contents

Introduction 2

The shifting landscape of software development lifecycles 3

Agile Manifesto 5

What is DevOps? 6

Key risks and challenges in implementing DevOps 8

Role of internal auditors in DevOps 10

Where can KPMG help? 11

Introduction

Organizations are under constant pressure to keep pace with technical advances while optimizing the customer experience. As a result, a majority of organizations are adopting an Agile approach supported by DevOps practices to automate their software development process.

It has long been the precedent that software development mirrors the governments and organizations in which it's embedded. In such entities, laws often dictate a separation of duties and access restrictions to sensitive materials. While traditional techniques, such as the Waterfall model, reflect this unidirectional structure—assigning specific tasks to individual teams—they have a tendency to clash with more modern disciplines, such as DevOps, Agile and Lean. Insufficient documentation only aggravates the situation, confusing team members as to their responsibilities and compliance requirements.

When implemented amid this confusion, traditional audit plans find themselves quickly outdated, built around last year's performance and risk assessments. Modern practices in software development require that the functions of auditors evolve accordingly. By including internal auditors within DevOps, companies equip themselves with a means to guard against inefficiencies and opportunities to design control procedures that acknowledge the latest processes and tech.

Experts recommend that auditors take the initiative to educate by working collaboratively with engineering teams on risk mitigation, allowing for a healthier DevOps environment. When controls are designed and implemented correctly in DevOps, it enables the organization to address the end-to-end traceability of the change.

In this document, we detail the implementation of internal audits in DevOps, the key risks and challenges involved in this process, as well as, controls to mitigate these risks. Being a prominent voice in this space, KPMG commits itself to sharing industry leading practices and expert knowledge that are relevant to internal auditors and DevOps teams everywhere.

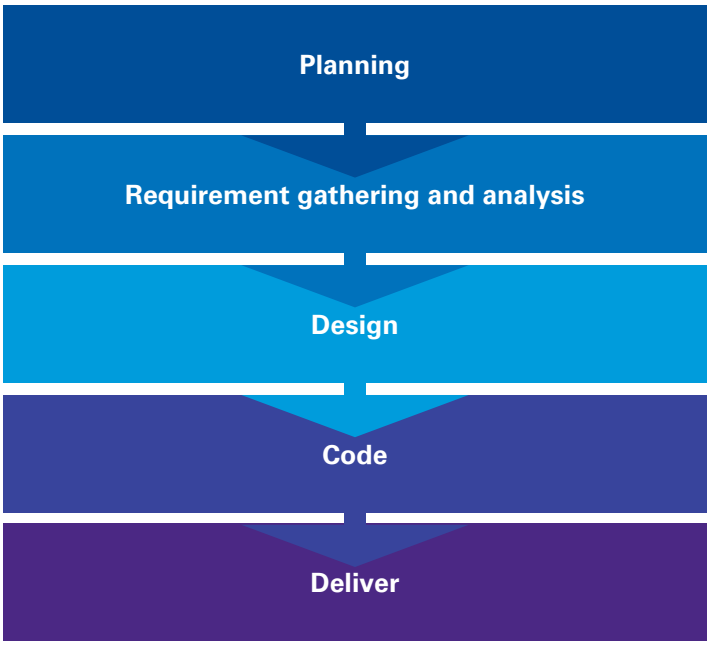
The shifting landscape of software development lifecycles

Conventional software development methodologies have always dealt with unidirectional sequences of phases.

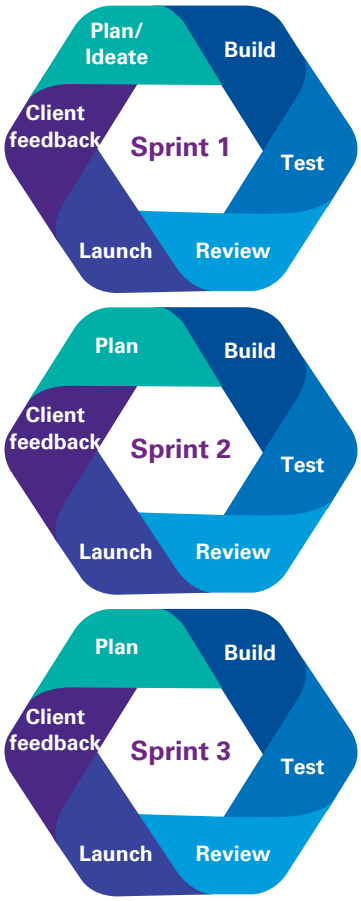
In each phase, a specific activity must be completed before the subsequent phase begins. Since the advent of Agile methodologies in 2001, software development has taken on a more interactive, dynamic, and team-focused approach. Shifting from a methodology of phases to one driven by sprints or iterations, Agile approach moves work forward incrementally without impeding the progress of the work through the development flow.

Given the numerous advantages of a sprint-based technique, companies are now adopting en masse to the Agile approach. Agile proves itself adept at preventing substantial delays in software development and ensuring customer satisfaction, as well as higher-quality work. With the Agile methodology, the enterprises are able to respond to the customer needs, and at the same time, increase the speed of delivery. Agile SDLC and DevOps practices not only accelerates development and release, but also when architected correctly, incorporates a faster feedback loop across the delivery chain.

Waterfall model



Agile methodology



In the Annual State of Agile 2019 Survey conducted by CollabNet VersionOne, 97 percent of their respondents reported that their organization was utilizing the Agile methodology in software development. While Agile adoption is on the rise, a majority of those same respondents claimed that not all teams at their company had fully adopted the Agile approach yet. Instead, many companies now follow a hybrid approach—a mix of Agile and Waterfall processes—over the course of a software development lifecycle.

Agile Manifesto

In 2001, a group of developers calling themselves "the Alliance" sought to overhaul software development as a whole and "restore credibility to the word 'methodology.'" Opposed to the Waterfall model and the concept of "documentation-driven, heavyweight software development processes," they formulated a manifesto, distilled into four fundamental values and 12 supporting principles, which assist and guide in software development.

Central to the Agile approach is its commitment to generate software incrementally while accelerating the release of new versions of software in shorter cycles of development.

Proponents of the methodology invoke the four values outlined in the Agile Manifesto—promoting software development processes that emphasize quality, through the creation of products that meet consumers' needs and expectations. The set of 12 principles pairs seamlessly with these values, creating and fostering work environments where the customer is the focal point. In such an aligned organization, business objectives are more easily met, and developers have the ability to quickly respond to users' needs and other market forces.

The Agile Manifesto

The Agile approach applies the four values for software development.

— Individuals and interactions over processes and tools

— Working software over comprehensive documentation

— Customer collaboration over contract negotiation

— Responding to change over following a plan

The Agile principles

— Customer satisfaction

— Accommodation of changing requirements

— Periodic delivery of working software

— Collaboration between the stakeholders and developers throughout the software development lifecycle

— Team motivation and support

— Face-to-face interaction

— Primary measurement of progress is the working software

— Consistent development pace

— Attention to detail

— Simplicity

— Attend continuously to good design

— Retrospect and adjust regularly based on lessons learned

Source: Agile Alliance website; Agile 101; The Agile Manifesto

What is DevOps?

Based on Lean and Agile principles, DevOps (an abbreviation for Software Development and Operations) emphasizes the collaboration of development and operation teams, as per project requirements, to efficiently automate the software development and delivery process. Using various practices and tools, DevOps encourages a culture where small, interdisciplinary teams take collective ownership of their projects, delivering software that meets users' needs. By combining Agile methodology with DevOps practices, teams deliver iteratively in small batches, focused on value delivery of their products and automating much of the repeatable tasks in the software development lifecycle. Through automation and connected toolchain, DevOps enables a flow of continuous integration, continuous deployments, continuous delivery, and automated application testing. Key stakeholders of the DevOps teams include:

— Product Management

— Development

— Quality Assurance (QA)

— Internal Audit

— IT Operations

— Information Security

Automating the software development lifecycle through DevOps

By reducing manual and repetitive work, and streamlining workflows, DevOps vastly improves the speed and quality of the software development cycle. An effective DevOps utilizes tools, people, and processes available to an organization—automating formerly manual tasks like testing, deployment, and delivery of products. Automation tools and processes in DevOps include:

— Source Control library structures

— Automated builds

— Automated deployment to all environments, including production

— Automated developer testing (i.e., unit testing)

— Security testing – static application security testing (SAST), dynamic application security testing (DAST), software composition analysis (SCA), fuzzing, and penetration testing

— Continuous integration

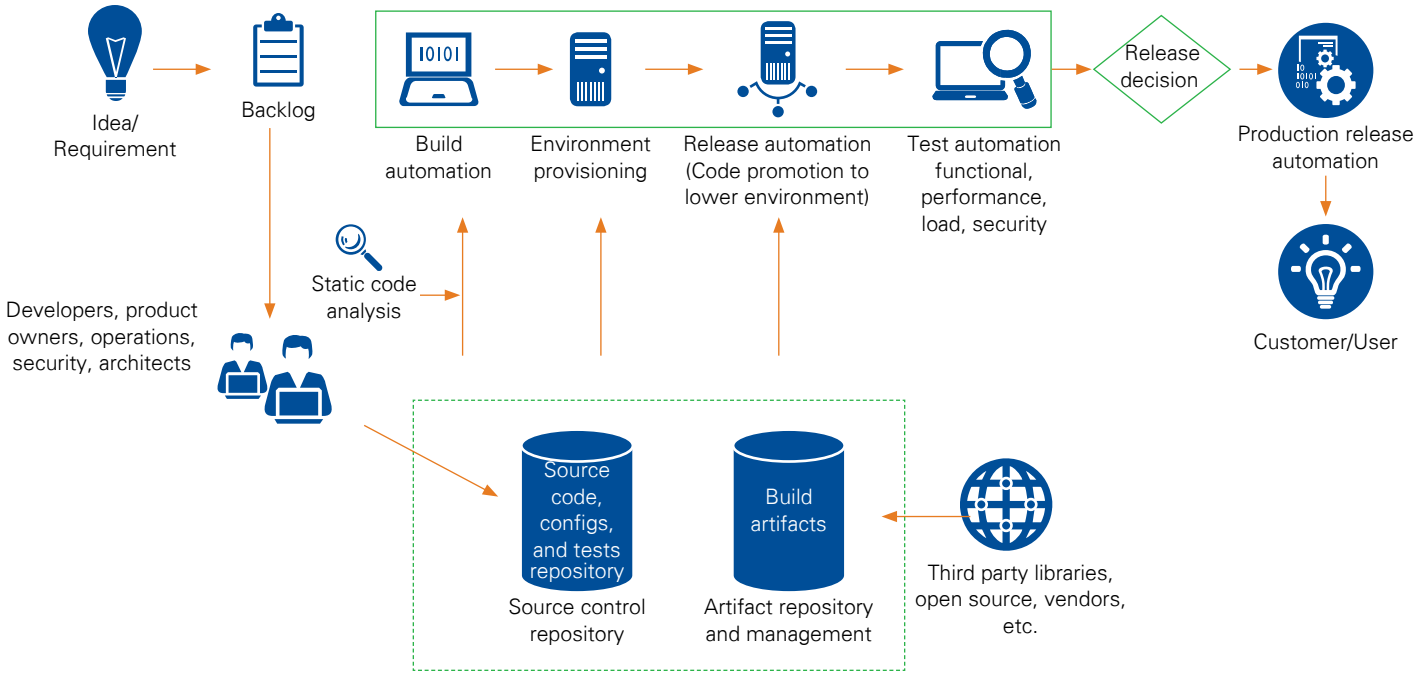
— Production application self-monitoring automation

— Flexible infrastructure environment

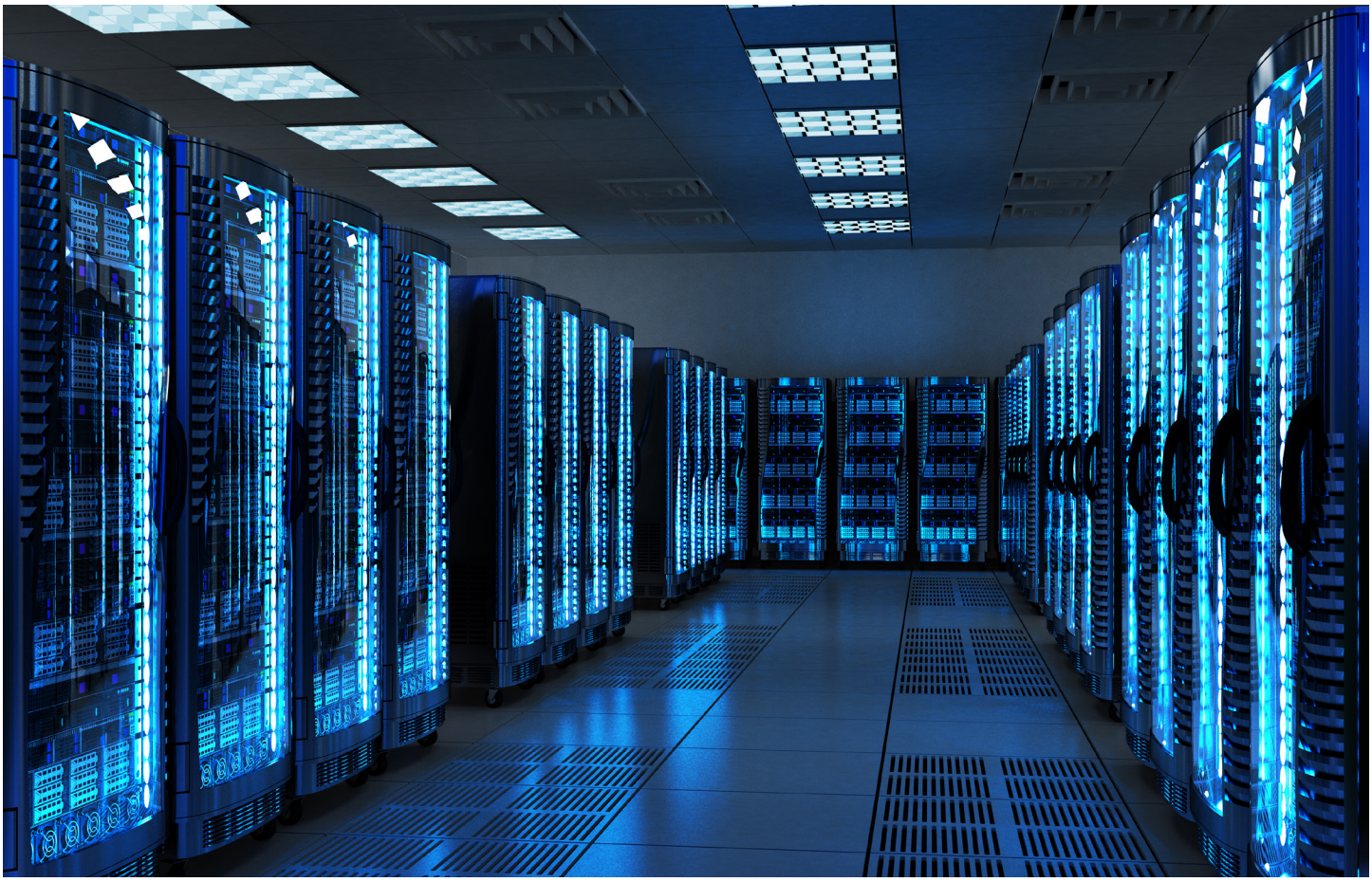
— Test data management automation

— Agile teams using Scrum or Kanban

*Note: In some leading organizations, the operations team is shifting further left and incorporated into the development team (such as Google SRE), while in other instances the operations team is eliminated based on the product focus and automation. In such cases, the tasks carried out by the operations team are shifted to infrastructure.



A set of controls is orchestrated into the delivery pipeline to enable traceability while making data-driven decisions to progress the change faster and replace manual handoffs.



Key risks and challenges in implementing DevOps

A faster and more flexible business model, like DevOps, must be paired with adaptive risk strategies to fully succeed. Although many companies now adopt DevOps, most of their risk frameworks still remain linear in nature—more suited to conventional software development. Inevitability, this incompatibility leads to a number of bottlenecks.

Key risks	Key controls
Adoption and organizational change	
Adopting DevOps without adequate preparation Organizations that adopt DevOps with a limited staff of DevOps-trained professionals increase the risks of failing the regulatory compliance, decreased morale, and poorer quality of work produced.	**Documentation of policies and procedures** Maintaining documentation of all policies and procedures ensures that security and risk requirements remain fulfilled in DevOps. Policies can be incorporated into all the tools used by DevOps teams, such as Jira or Azure DevOps, and routinely reviewed for updates.
Change management while scaling DevOps Large, cross-functional teams and complex solutions can cause additional work. Opting employees into DevOps early on in adoption—allowing them to plan for the transition—affects its successful deployment down the line, as it's gradually scaled to the entire organization.	**Peer reviewing the changes** As a part of Continuous Integration, facilitating and performing the peer review process and documenting all discussions as a part of feedback, ensures a greater consistency between segregation of duty for the changes made in development and those in production.
Tools and processes	
Insufficient documentation Processes and Principles can be applied asymmetrically, driven by individual experience and/or knowledge.	**Critical versus noncritical changes** Critical and non-critical changes may require separate treatment such as peer review. Automated controls driven by data from connected toolchain and code check in criteria may help in determining the eligibility.
Presuming Agile sufficiently prepares for DevOps Agile methodologies alone will not be enough to implement DevOps effectively, without the organization of tools, processes, and people.	**Appropriate code management** The Agile methodology enables developers to use a wide range of tools, establishing a single-source repository, which helps reduce the risk of introducing malicious code into the development and production environments.
Focus on speed over collaboration Any gap in communication, due to fixation on rapid product delivery, has the potential to overlook client feedback or ferment into conflicts between development and operation teams.	**Blue Green deployments** Blue Green deployment is a process that mitigates the risk of downtime before software development, by requiring DevOps to run two parallel production environments—one in testing and one in production.

KPMG

Key risks	Key controls
Security and compliance	
Security risk in scaling up As development infrastructure scales itself, the intersection of multiple security groups, multiple server and cross-functional teams raises considerable security risks. Secure DevOps, or DevSecOps (an abbreviation for Development Security Operations), is an IT security approach based on these principles. Spanning the entire software development lifecycle, Secure DevOps identifies and mitigates attacks, while monitoring and protecting applications. Although an emerging concept in DevOps organizations, its implementation proves invaluable. **Compliance risk** Processes pertaining to change management must be controlled in the software production environment, or organizations will fall prey to compliance violations. **Segregation of duties** DevOps breaks down organizational silos, sharing access and responsibilities between developers and operations. This may increase the risks of mistakes, frauds, and insider attacks.	**Automation of controls** Automated vulnerability scans can monitor for potential to compromise the security of production platforms and identify any issues that may disrupt the software release process. Automated testing can serve as a quality gate for code, determining if it contains too many vulnerabilities. **Preauthorization of users** Preauthorized logins provides developers the access to functions entitled to them, while curtailing the number of users with said authorization. Auditors may then more easily track and monitor changes that occur in production. **Reconciliation of system transactions** This is a detective control, which can be used to reinforce monitoring of developer production data edits. An automated reconciliation is performed between the database transactions versus transactions captured by developer production data edit reports to verify the completeness of the reports being reviewed. **Segregation of duties and access logging and monitoring** The logging of access and changes in production code by developers allow for this compliance. Additional layers of access restrictions should be implemented so that developers do not have access to critical and conflicting business functions.

Key risks	Key controls
Governance	
Coping with continuous high-speed changes As teams adopt modern DevOps practices, it increases the team's productivity resulting in faster releases of solution delivery to the end customer. Due to this higher delivery rate, it becomes difficult for security and compliance to unblock the delivery changes.	**Deployment of small code batches** The volume of risk increases with the number of functional code lines per deployment. Such liabilities can be eliminated within smaller batches of code, bringing shorter cycles, quicker feedback, increased efficiency, and lower overhead costs. **Embedding security into the developer experience** Trainings for developers, such as the Open Web Application Security Project (OWASP) Top Ten, advise on secure coding techniques and common vulnerabilities. Security Champion Programs may also be used to enhance security awareness of development teams, with minimal investment. Given the growing frequency and complexity of cyberattacks, it may be prudent to include security engineers to respond to the highest-risk applications.

Role of internal auditors in DevOps

During the design and planning stages, auditors should be focused on assessing quality and partnering with organizational stakeholders. By expanding beyond the traditional scope of test-and-run project end stages (particularly when a high level of DevOps automation is in place), auditors can provide more value and overall satisfaction to organizations. Once among operation and development teams, they can recommend controls from the beginning, gaining the trust and camaraderie of their peers.

Internal auditors play a pivotal role in ensuring compliance, maintaining security, and mitigating risks. The ability to collaborate with developers, operations, and security allows augmentation of existing teams with the skills, behaviors, and competencies needed. At the same time, internal auditors can identify gaps and controls in a process and design it accordingly. This new way of cross-functional collaboration is a "shift-left" practice in DevOps. It helps the teams to improve quality and security by way of continuous testing in SDLC process. The increase in automation increases the data and artifacts produced consistently in a repeatable fashion throughout the gates and control points in the pipeline.

The traditional approach to change management and logical access control objectives invariably leads to slowdowns in delivery. Audit focuses should no longer be confined to production servers and systems but extended to the continuous integration (CI) and continuous deployment (CD) tools and technologies that support key processes in DevOps. Teams with the right balance of controls and continuous auditing/monitoring can still take advantage of the automation capabilities that DevOps offers.

It's imperative that organizations develop an audit strategy that keeps pace with DevOps. Potential areas for internal audits include:

Pre-postimplementation assessments and readiness for audit

Evaluate the DevOps strategy – Conduct an assessment of the overall governance and strategy around DevOps. A review can be completed over the product team's adoption of DevOps practices, production management, DevOps capabilities, and training/development.

Assist Management in secure DevOps – IT internal audit can assist management in identifying opportunities to increase security through an evaluation of controls, behaviors, or capabilities across the SDLC for risk reduction. This can be achieved through "lighttouch" security testing across the organization's most critical applications, which could then translate to more detailed security requirements to provide to developers at project initiation.

Deep dive on change management – Within an Agile environment, developers may have access to deploy code to the production environment in order to enable the continuous development process. A deep-dive audit into change management – related access controls and segregation of duties, could identify gaps brought about by developer access to production.

Conduct a Dev Ops tools review – IT internal audit can perform a review over the integrity of the CI/CD tools (where these are used to strengthen the control environment).

Where can KPMG help?

Several leading organizations have turned to IT internal audits to determine if they possess the appropriate technical skills, experience, and flexibility to meet the demands of a complex IT environment. KPMG keeps pace with the constant evolution of business conditions, as well as compliance, legislation and regulations that overshadow organizations. Our potential IT Internal Audit services for DevOps include the following:

— Identify the controls in a CI/CD process and aid teams in documenting the risk and controls matrix

— Ensure appropriate technical and governance controls are built to support high-speed delivery

— Embed compliance as part of the "continuous everything" process

— Train your cross-functional team on maturing the controls.

KPMG differentiates from its competitors in that it offers enhanced IT governance and risk management, courtesy of qualified IT audit resources with global experience.

— Real-time assessment of significant systems/ applications with a clear Agile and DevOps practices

— Gap analysis that includes security/Segregation of Duties (SoD) and outsourcing assessment

— Robust audit findings based on external experience

We leverage an in-depth understanding of how an integrated audit functions, DevOps and cloud tech, being a Microsoft Gold DevOps Partner. Benefits of our partnership include:

— Microsoft Go-To-Market services for clients of Microsoft Partner

— Access to DevOps Partner Sales and Marketing Portal to search useful resources

— Provider of Microsoft Developer Tools Deployment Planning Services.

When utilized in a DevOps environment, our services yield immediate benefits throughout an organization, aiding senior business management, senior IT management, boards of directors, and audit committees. Our record for identifying potential improvement areas/weaknesses has saved our clients considerable sums of time and money in software development and lowered the risks of IT security with systematic, preventive, and detective controls.

Contact us

Nicole Lauer
Principal
IT Internal Audit Leader U.S. and Americas
T: 410-949-8949
E: nlauer@kpmg.com

Lavin Chainani
Director, Advisory
T: 410-949-8834
E: lchainani@kpmg.com

Shahnavaz Alware
Director, Advisory
T: 858-366-3440
E: salware@kpmg.com

kpmg.com/socialmedia

Produced by Create Graphics/Document number: CRT056309A