TCG
POSITIVELY
DISTINCT®

7438 GEORGIA AVENUE NW, WASHINGTON DC 20012
202-986-5533 | WWW.TCG.COM | INFO@TCG.COM

NAVIGATING THE DEVSECOPS LANDSCAPE

Common Blockers to DevSecOps Releases and Solutions

Introduction

The Blockers to Rapid Software Releases

In the ever-evolving landscape of DevSecOps, understanding and resolving common blockers to efficient releases is paramount. These blockers can lead to significant business risks, including security breaches, loss of customer trust, and diminished productivity. Addressing these issues is crucial for maintaining a competitive edge and ensuring timely, secure software delivery. Organization stakeholders face various challenges such as slow and insecure release processes, significant delays in requirement identification, regulatory barriers, prioritization dilemmas, and remediation challenges emanating from security scans that impact different roles. By exploring these pain points from a role-based perspective, we aim to provide targeted solutions that enhance collaboration, streamline processes, and promote rapid, secure and reliable software releases.

Background and Importance of DevSecOps

DevSecOps builds on the foundational principles of DevOps, such as agility, collaboration, and automation, while embedding security into every stage of the software development lifecycle. Traditionally, development, operations and security teams work in silos, leading to slow releases, inefficiencies and unresolved vulnerabilities. DevSecOps breaks down these barriers, promoting close collaboration across all teams. This ensures that secure code is developed quickly, tested thoroughly, and deployed efficiently.

For federal agencies, DevSecOps offers the added benefits of enhanced security and compliance, which are crucial given the sensitive nature of government data. By incorporating automated security scans, continuous monitoring, and real-time compliance checks, DevSecOps ensures that software is not only delivered quickly but is also resilient against evolving threats.

Identifying and addressing Key Pain Points from a Role Based Perspective

Slow, Unreliable and Insecure Release Processes

CIOs are often challenged by delayed releases due to inadequate integration between development, operations and security teams. They also face problems when the release process is insecure, posing strategic risks to the organization. To cope with this, CIOs can push for the implementation of automated DevSecOps processes. This will speed up release times while ensuring a secure release process through automated security checks, leading to increased efficiency and reduced risk.

Product Owners find it difficult to manage stakeholders when the release process is too slow or unreliable. They struggle to satisfy customers and meet their expectations. By collaborating closely with the dev and operations teams to implement Agile processes they can create a feedback loop that allows them to better communicate delays. Also, automation will significantly reduce the time spent on approvals, security checks, and deployment, making the product release quicker and more reliable.

Development teams feel the pressure when the process of releasing user stories to the production environment is slow or halted due to security issues. This lengthy release process often involves a long list of steps that must be executed manually by a team member who is also responsible for other tasks, all while the team is under pressure to increase velocity. This could contribute to lower morale and productivity. Addressing this by adopting automation tools to streamline build, test, and release processes can reduce the chance of errors, improve efficiency and provide reliable outcomes, leading to a more effective release process.

Significant Lag in Identifying Requirements and Deployment

CIOs can face strategic issues due to the time lag in identifying key requirements and then deploying solutions. Such a delay can affect overall project timelines. To prevent delays, CIOs can promote the adoption of Agile methodologies, enabling short, iterative cycles that allow for rapid adjustments in response to new intelligence or changes in strategic requirements.

Product Owners, in their attempt to meet customer expectations, bear the brunt when there's a delay in identifying requirements or bugs and its implementation. They might also face reduced customer satisfaction. Implementing an Agile methodology to create shorter development cycles can enable faster movement from requirement identification to deployment and allow user feedback to be incorporated quickly, making products more responsive to market demands.

When **development teams** discover requirements, bugs, or security issues late in the cycle, this can result in significant rework, causing an imbalance in workload distribution and disrupting the iteration. Late requirements can be addressed by improving communication and feedback channels with the product owner to ensure timely input on evolving needs. On the other hand, late-discovered bugs or security issues can be mitigated through faster, automated deployments.

Identifying and addressing Key Pain Points from a Role Based Perspective (continued)

Regulatory Barriers

CIOs have to manage the balance between regulatory compliance, often seen as a barrier to innovation and timely delivery, and creating value for their stakeholders. By pushing for continuous compliance practices and prioritizing regulatory compliance in product roadmap development they can reduce delays and encourage continual improvement within regulatory bounds.

Product Owners can find it difficult to manage products given strict regulatory barriers which may prove inflexible to the product's roadmap. Regularly updating the team's understanding of these barriers and incorporating it into the planning process allows for proactive management and timely adjustments, reducing the impact of regulatory issues on release schedules.

For **Development teams**, compliance regulations can often mean more paperwork and less coding, which can reduce their productivity and morale. Where possible, automating compliance checks as part of the team's deployment pipeline can significantly reduce the time waiting for compliance approvals and increase efficiency.

Prioritizing Implementation

CIOs often have to balance long-term strategic goals with addressing immediate issues, which can lead to strategic initiatives being sidelined. CIOs can counter this by establishing clear strategic roadmaps and ensuring that iterative improvement processes, such as those supported by DevSecOps, are implemented consistently. This balance can ensure that immediate issues are addressed without derailing long-term strategic goals.

For **Product Owners**, it's a challenge to rank feature sets based on customer expectations and the team's ability to deliver. Developing a clear product roadmap and implementing it as part of backlog management can help handle prioritization. Regular backlog grooming sessions would also help in prioritizing tasks and managing unexpected needs effectively. Finally, short iterations with speedy deployments give them flexibility that makes hard prioritization decisions easier.

Development teams often juggle with a variety of issues, making prioritization a complex task. Implementing Agile methodologies will assist with managing tasks despite shifting priorities. Daily stand-ups and regular sprint planning can also provide a clear direction and keep the team

Identifying and addressing Key Pain Points from a Role Based Perspective (continued)

Security Scans and the Remediation Troubles

CIOs bear the responsibility for any breaches that take place, which makes security scans and remediation a critical task which can often hinder the overall release cycle. To address this issue, CIOs should advocate for the integration of security considerations throughout the development process, often referred to as "shifting left". By integrating automated security scans early and throughout the process, costly and time-consuming remediation can be mitigated.

Product Owners might face a decrease in customer satisfaction due to delayed releases caused by these security scans and their remediation processes. If they prioritize the incorporation of security practices early and at every step of the product development lifecycle they will prevent security issues and also proactively address them before they become a threat to the product's success and reputation.

For **Development teams,** scans that come late in a cycle can reveal a gamut of issues that need an immediate fix. This can cause undue stress, especially if the team has moved on to the next release. Instead, teams can integrate security testing such as static code analysis and software composition analysis into the CI/CD pipeline to ensure that security checks are performed automatically as part of the release process. Partnering with security from the onset and enabling automated security scanning can also help in identifying vulnerabilities early and reducing the need for extensive rework.

DevSecOps - A Strategic Approach

TCG's DevSecOps approach combines Agile principles, near neighbor technologies, and iterative implementation to promote collaboration, transparency, and right-sized DevSecOps solutions for customers. This approach aims to deliver high-quality software efficiently, with continuous feedback and improvement loops. Its three key tenets are:

3 Key Strategic Tenets

Agile Collaboration and Transparency

The DevSecOps approach works hand in hand with Agile methodologies to foster collaboration and transparency between development, operations, and other stakeholders. Agile practices such as Scrum or Kanban enable cross-functional teams to work closely together, promoting open communication and sharing of responsibilities. Regular stand-up meetings, sprint planning, and retrospectives ensure that everyone is on the same page and working towards common goals.

Identifying Near Neighbor Technologies

In the context of DevSecOps, "near neighbor technologies" refer to tools, technologies, or practices that can integrate with or easily replace those already in use by a customer to enhance the development and delivery process. During the initial stages of implementing DevSecOps, teams identify and integrate these technologies into their existing toolchains to streamline workflows. Examples of near neighbor technologies might include version control systems, continuous integration servers, automated testing frameworks, containerization platforms, and deployment automation tools.

Iterative Implementation

The DevSecOps approach follows an iterative implementation process to right-size the DevSecOps practices for the customer's specific needs. Instead of attempting a full-scale transformation all at once, the team takes incremental steps, identifying pain points and opportunities for improvement along the way. They start with a small-scale pilot project or a single application and gradually expand DevSecOps practices based on feedback and lessons learned.

By integrating Agile practices, identifying near neighbor technologies, and embracing an iterative approach, the DevSecOps implementation becomes collaborative, transparent, and tailored to the specific needs of the customer. This approach allows organizations to gradually adopt and optimize DevSecOps practices, ultimately leading to more efficient and reliable software delivery.

Conclusion

The successful adoption of DevSecOps for federal agencies hinges on addressing the key blockers that slow down software releases and expose systems to security vulnerabilities. By integrating automation, security, and agile methodologies, agencies can streamline their processes while maintaining robust security protocols. Solutions such as automated security scanning, real-time vulnerability remediation, and continuous compliance practices empower teams to overcome slow release cycles, regulatory challenges, and delayed CVE responses.

The strategic approach to DevSecOps emphasizes collaboration, transparency, and incremental improvements. By fostering close collaboration between development, operations, and security teams, and embracing technologies that integrate seamlessly into existing workflows, federal agencies can mitigate risks while improving efficiency. Iterative implementation allows for gradual scaling of DevSecOps practices, ensuring that solutions are tailored to the agency's unique needs. As these practices mature, agencies can expect faster, more secure, and reliable software deliveries, helping them achieve their operational goals while maintaining compliance with ever-evolving security standards.

To explore how this solution may fit your agency's needs, please contact:
Michael Drescher
Executive Director of Growth
michael.drescher@tcg.com
703-969-7010

For technical questions and more on how to implement DevSecOps, please contact:
Dr. Robert Buccigrossi
Chief Technology Officer
robert.buccigrossi@tcg.com
202-742-8473