The Federal Government is continuously under pressure to deliver technology solutions and capabilities faster and at a lower cost. To meet this need, agencies are optimizing their systems development lifecycles (SDLC) by adopting more modern practices that promise to improve and expedite delivery (e.g., Department of Veterans Affairs’ [VA] new Veteran-focused Integration Process [VIP]). In some cases, agencies are establishing internal communities of excellence to address the challenge (e.g., General Services Administration’s [GSA] 18F team).

While the landscape of technologies, methods, and processes to expedite the delivery of new capabilities is evolving, any velocity gains achieved through contemporary Agile or DevOps methods are quickly erased in the event of a high-profile data spill or exposure of the nation’s assets resulting from a vulnerable system, such as the 2015 Office of Personnel Management data breach. In 2012 alone, DoD public websites had 4 billion visits, according to Christopher Lynch, who heads the Pentagon’s U.S. Digital Service. He states that 25 percent of the visits were nefarious in some way—in other words, there were over 1 billion attempts to undermine DoD system security in a single year.

So, while the Federal Government is eager to adopt many of these forward-leaning DevOps and Agile methods and technologies, first the commercial industry must develop and adopt them. The commercial industry has a different set of security governance controls and regulations than those required for federal agencies. Therefore, the challenge becomes how to adopt modern development practices and transform a federal agency’s delivery model without sacrificing information assurance and system security controls.

The challenge must be approached from many angles. Specifically:

+ **Mission Understanding:** When developing any system, security requirements and controls can’t be segmented from technical requirements. Compliance models, such as the Risk Management Framework (RMF), ICD 503, NIST 800-53, DISA Security Technical Implementation Guides (STIG), and the Federal Risk and Authorization Management Program (FedRAMP) are necessary controls established to protect the nation’s assets. There must be a deep understanding of how these security requirements complement capability requirements for the system under development and how to incorporate them intelligently as part of the SDLC.

+ **Technical Acumen and Innovation:** Many of the commercially popular methods to improve velocity are limited to building and delivering code as expeditiously as possible. But without addressing the hardening of the architecture and federal compliance controls iteratively as a part of the process, the same waterfall-centric inhibitors exist when attempting to attain an Authority to Operate (ATO) in the Federal Government. Expertise in how security is incorporated, tested, and monitored as part of DevOps methods (continuous deployment, infrastructure as code, containerization, continuous diagnostic monitoring) is critical to increase velocity with confidence.

+ **“Secure First” Culture:** It is often said that the adoption of DevOps and Agile methods is a “team sport,” and the same holds true when addressing the integration of security controls. The waterfall model of deploying code to the Authorization and Accreditation (A&A) team and waiting for an ATO has to shift to meet an agency’s goals for expedited delivery of secure, high-quality code. Key
to this shift is the challenge of incorporating representation from the branch’s or agency’s information assurance or A&A body into the story elicitation and prioritization process and reinforcing a shared understanding of the requirements that must be satisfied to achieve an ATO. This evolution requires a change management initiative harnessed by Agile Coaching, training, and strategic communications. A deliberate organizational change approach, led by experienced professionals, is required to transform an agency’s delivery model. This is the difference between “doing Agile” and “being Agile.”

Similar to how modern software systems are developed, it is best to take an incremental approach toward this transformation goal by inspecting, adapting, and changing through this journey. This paper provides proven approaches to guide the transformation journey to secure agility.

SUCCESSFUL AGILE TEAMS UNDERSTAND HOW SYSTEM SECURITY IS INTEGRATED WITH MISSION

As organizations adopt Agile methods to meet strategic mission goals with increased velocity (e.g., the needs of the warfighter or veteran, national emergencies, or improving the lives of citizens), understanding those missions, and how security plays into the larger picture, is imperative to any successful systems delivery effort. As an example, the VA has a strategic objective to ensure healthcare claims are processed in a timely, secure manner and in turn, veterans receive the care they need and deserve. Therefore, it is important in any systems delivery program supporting the VA for all relevant team members, including the User Experience (UX) Engineer, Developer, Tester, Scrum Master, Product Owner, and Project Manager, to understand how the system they are developing supports this mission and how information assurance and privacy play into that equation.

As the best Agile teams are cross-functional, team members take collective responsibility for the total solution by meeting mission requirements and satisfying the required security regulations. This approach includes cross-training across multiple dimensions: role, technology, architecture, and solutions. Only by having a perspective on the VA’s Enterprise Architecture, broader needs for interoperability across systems and platforms, and how protected health information (PHI) regulations and Health Insurance Portability and Accountability Act (HIPAA) compliance will be assured can a delivery team effectively implement Agile-based delivery cycles for the VA and deploy a secure system the intended user base will adopt. This is the difference between commoditized “software development” to deliver code and effective “systems delivery,” which results in a continuously secure solution.

AN IN-DEPTH KNOWLEDGE OF FEDERAL SECURITY POLICIES AND REGULATIONS IS PARAMOUNT

To effectively deliver secure software through Agile methods, teams must understand the current and future federal security standards and agency-specific controls up front to inform planning, and they must have experience delivering mission-critical systems within these controls. For example, if a delivery team commences sprint activities toward a Minimally Viable Product (MVP) without an appreciation for how the RMF or NIST 800-53 standards would ultimately be applied to attain an ATO, the team runs the risk of expensive refactoring or hardening sprints to meet compliance late in the lifecycle, eliminating any gains realized through agility.

For any system delivered for the Federal Government, a successful Agile development team will be continuously knowledgeable of the latest guidance on vulnerabilities from the U.S. Computer Emergency Readiness Team (US-CERT) and incorporate that guidance into iterative story elicitation and development activities. All DoD IT assets (operating systems, databases, open source software, wireless devices, etc.) must meet varying levels of STIG compliance before they can operate on DoD networks, and these models are continuously updated. The Agile development team must incorporate these controls when eliciting technical stories. If cloud service providers are a part of the agency’s long-term hosting strategy, incorporating input from FedRAMP Third Party Assessment Organizations (3PAO) as a part of technical story development becomes important. Finally, and most importantly, delivery teams must understand the sensitivity of the data created, stored, and leveraged across the organization and address its protection as part of
A delivery team with a deep understanding of an organization’s mission will bring to the story development process a knowledge of data privacy needs, as well as the motivations and capabilities of those who would attempt to attack the system under development.

ADOPTION OF AGILITY PROVIDES A MODEL OF “CONTINUOUS AUTHORIZATION” ACROSS THE STACK

Contemporary Agile and DevOps tools and methods are bringing a step-function improvement to the effectiveness and efficiency of systems delivery activities across industry. According to the 2015 State of DevOps Report conducted by Puppet Labs, organizations implementing DevOps processes and tooling are realizing a 60-percent reduction in failed deployments and a mean time to recover over 150 times faster. Similarly, at Booz Allen, implementation of lean Agile development methods has resulted in an average defect removal rate that is nearly 10 times better than industry standards. Organizations employing these tools and processes are deploying code more frequently, with higher quality, and in more stable environments. As automation and measurement are two of the guiding principles of Agile-based development, organizations that leverage continuous integration and continuous deployment (CI/CD) pipelines to automate security testing and code scanning are able to use build and deployment success metrics in real time to remediate vulnerabilities at the time of development, when they are exponentially less expensive to correct. However, simply conducting code scanning as part of CI/CD activities is insufficient.

The Agile team must be able to continuously address security through the entire stack (e.g., application level, operating system, network, and database) to meet most agency-specific controls, as well as NIST 800-53, ICD 503, DoD 8500, and/or DISA STIG compliance. Leveraging proven procedural and architectural security patterns (e.g., Open Web Application Security Project [OWASP] Proactive Controls, Build Security in Maturity Model, Microsoft Secure Development Lifecycle) facilitates the creation of secure software in an Agile environment by layering proper protocols, implementations, and automation throughout the process of construction. These considerations also extend to how the team’s development environment is established. For example, security-minded Agile teams implement secure configuration management processes and infrastructure as code tooling (e.g., Chef Compliance) to prevent the introduction of vulnerable or noncompliant code and reduce the risk of damage posed by a malicious actor. If the development team is leveraging open source utilities or frameworks, it is important to conduct static analysis on that code, using tools such as Black Duck, to understand its posture against current known vulnerabilities. Employing modern microservice-based architectures, combined with container-based deployment utilities (e.g., Docker), is also becoming increasingly popular because they not only advance the extensibility and stability of the system under development but also reduce the surface area for potential attack. However, even when leveraging a containerization-based approach, it is critical to establish a repeatable workflow that continuously hardens the images as part of the development process and protects the containers as part of deployment.

Also, it is important to recognize that what is secure today may not be tomorrow. Agencies must assume that there will be an eventual attack or compromise and develop continuous diagnostic monitoring and investigative capabilities for root-cause analysis as a part of their DevOps process. Using architectural security patterns, coupled with strategies such as data segregation, tooling to verify security configuration of cloud implementations, intrusion detection and prevention systems (IDS/IPS), trusted containerization approaches, and automated penetration scans, begins to create the full complement of addressing security throughout the entire stack.

SUCCESSFUL AGILE TEAMS INCORPORATE SECURITY AS PART OF PROJECT “CEREMONIES”

The Agile Manifesto values “working software over comprehensive documentation.” Therefore, as agencies migrate to leaner development models, one of the primary challenges is to meet the needs satisfied by traditional security documentation while optimizing development lifecycle
activities. To address this challenge, more organizations are now leveraging lean wiki-based platforms or modern tooling to iteratively generate and collaborate on security-based documentation as the system under development iteratively evolves. Also, by incorporating security-related activities as a part of Agile ceremonies, agencies are able to meet the required security governance controls while achieving the efficiencies of lean, Agile-based development. Examples include incorporating NIST 800/RMF/ICD 503 compliance stories (i.e., technical stories) as part of backlog grooming activities; generating threat model designs as part of “Sprint 0” or architectural sprint activities; leveraging automated OWASP-based unit tests in AppScan, Fortify, or CAST as a part of CI and build activities; radiating security-focused defects to provide visibility into the security state of new functionality; and leveraging log monitoring and alerting tools as part of DevOps processes. All are effective approaches to provide a lean development and deployment model with real-time monitoring for security posture.

ADOPTION OF A “SECURITY-FIRST” AGILE MINDSET REQUIRES PROJECT COACHING AND TRAINING

Security in the modern Agile world starts with a deliberate shift in mindset about security’s role before, during, and after the software development process; it should be ever present across the lifecycle in a systematic way. In fact, security needs to be elevated to a first-class citizen. This approach feeds into the Agile principle of developing a multidisciplinary team and breaks down the barrier of those responsible for implementing security and those responsible for the business requirements around security. In this model, security becomes a fully embraced practice and concern that exists for the entire life of the project. All talent, from developers to security professionals, should be “security intelligent”: trained in software development practices that are secure throughout. This knowledge includes awareness of threats and attack vectors not only in the layer of application being developed but also in the surrounding layers (e.g., teaming partners, third-party vendors).

Developing an environment where the Agile team is security intelligent is often facilitated through the use of Agile Coaches. The Agile Coach is not a security expert, but rather a credentialed (e.g., ICAgile, Scaled Agile Framework) systems delivery practitioner who works closely with teams and organizations to improve how the work gets done while leveraging Agile principles. The Coach discovers how the team is operating, helps craft a vision of where the program is heading, and serves as a guide to help the team reach that destination. For example, an Agile Coach may lead a team through its release and sprint planning to prioritize and schedule the delivery of capability. In a security-minded Agile approach, a threat modeling activity is incorporated as a part of that process to understand the potential security risks and drive the creation of technical stories to address those risks. By helping the team adopt lean principles, practices, and philosophies, as well as the use of tools within an organizational context, the project team is guided on how to optimize the necessary security practices.

ADOPTION OF SECURE AGILE PRACTICES MUST EXTEND TO INFORMATION ASSURANCE EXPERTS

In addition to the deployment of Agile Coaches, other organizational change management activities to integrate secure agility typically include organizational training on Agile and security standards starting at the executive level; establishment of a visible sponsor who is responsible for the institutionalization of the new processes; collection and internal communication of Agile success stories and metrics to evangelize the adoption of lean development processes; and reinforcement and governance of the use of automated security testing utilities, secure coding standards, and lean processes. This mindset shift represents a cultural change in how traditional security practices are seen and implemented; it is a transformational change that removes the costly “it’s them, not us” problem, which slows and burdens so many traditional enterprise software projects. As with any enterprise-wide change, an incremental and iterative approach is a proven best practice. Leveraging pilot programs or a maturity model-based approach to attain “quick wins” and iteratively advance the organization’s posture is more effective than a “big bang” strategy. By implementing strategic change management principles and desegregating security from daily Agile development practices, the agency or organization can lean its delivery model without sacrificing security controls.
CASE STUDY:
SECURE AGILITY AT JIDA

The Joint Improvised-Threat Defeat Agency (JIDA) is DoD’s Agile response mechanism for providing timely near-term solutions to the improvised threats endangering U.S. military personnel around the world. JIDA leverages unique institutional resources and a risk-managed environment to accelerate and streamline battlefield adaptation and facilitate rapid response to warfighters’ needs. Its success is defined by iterative solutions delivered at the speed of war.

JIDA selected Booz Allen as its partner based on our understanding of JIDA’s mission and ability to rapidly innovate. Through this partnership, Booz Allen created an adaptive technology platform and analytic tool suite that could react as quickly as the organization to answer emerging questions from an evolving data landscape and counter an increasingly resourceful, improvised threat. By leveraging advanced Agile and DevOps methods, the Booz Allen team developed a state-of-the-art data analytic platform that went from concept to full operational capability (FOC), with full security accreditation, in less than a year.

In stark contrast to the conventional systems development approach, the initial design activities focused on the core capability goals, accreditation type, boundary, and environmental constraints (user access, enterprise services, infrastructure, etc.), as opposed to the technical solution design. The critical difference was that we iteratively evolved the full operational Concept of Operations before designing a system, thereby ensuring that post-delivery, the system could be rapidly fielded with minimal certification effort. The Booz Allen team identified all security controls at the beginning of the project, with compliance and validation tracked and managed through deployment. After deployment, we collected customer feedback through a series of customer engagements to monitor risks to continued use and ultimate adoption by the user community.

Booz Allen designed a Secure Agile SDLC that increases speed drastically but, more importantly, protects JIDA’s information security by ensuring compliance with all DoD security requirements for software delivery on mission-critical systems. By focusing on continuous monitoring and validation of processes versus enforcing multiple approval gates, Booz Allen’s implementation of DevOps methods enables JIDA’s mission IT features to go from backlog to production delivery in near-real time. The design is composed of a fully automated code delivery pipeline equipped with software configuration management, unit and end-to-end testing, static code analysis, security control compliance scans, vulnerability scans, and penetration testing while offering full transparency and continuous monitoring to achieve true Continuous Authorization of the Catapult and ANTS systems. The result has been a 300-percent increase in delivery time and an average of 50 releases per year.

The Booz Allen team increased JIDA’s enterprise data holdings from 71 feeds in the legacy system (October 2011) to over 800 feeds (March 2016), increasing warfighter access to over 1,000 percent more data. JIDA was able to produce 11 major analytic tools, providing intelligence analysts with a true multifaceted fusion platform that vastly accelerates their workflows, increases the identification of high-value individuals, and ultimately saves lives. Beyond operational impact, the Booz Allen approach resulted in more secure and reliable capabilities. With an aggregate application quality score of 99.8 percent and “Excellent” third-party audit results for the cybersecurity posture of the enclave, the solution has emerged as a model of secure and reliable operation across the DoD and intelligence communities. The solution maintained a 100 percent score for JIDA Cybersecurity A&A metrics on ATO management and risk assessment quality.
SECURE AGILE CHECKLIST

Getting Started with Secure Agility

While all systems delivery programs must include information assurance and system security considerations as part of the solution, adoption of lean Agile development methods isn’t anathema to meeting federal regulations for security controls. Agile-based systems delivery is a team sport, and system security representation is part of that team. Assuming the delivery team possesses an understanding of the current and future controls as part of the agency’s mission, has the technical depth to leverage the latest Agile/DevOps tooling, and brings a consultative approach to the delivery of the system, any agency can adopt lean development methods without sacrificing system security. The checklist below contains a set of practices and process to help your organization assess the implementation and adoption of secure agility.

<table>
<thead>
<tr>
<th>Mission Understanding</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Is security talent embedded within teams and is each team member, from developer to security professional, “security intelligent”?</td>
<td>☐</td>
</tr>
<tr>
<td>Are software security fundamentals implemented, such as user authentication and access controls, protection against known attack vectors, and defense-in-depth?</td>
<td>☐</td>
</tr>
<tr>
<td>Does the development team have an understanding of current and impending regulatory security requirements (e.g., Risk Management Framework)? Have these requirements been addressed as technical stories and applied to sprints?</td>
<td>☐</td>
</tr>
<tr>
<td>Does the development team have an understanding of agency-specific SDLC governance models (e.g., VA’s Veteran Integration Process, DoD 5000) and how modern methods and tools can be leveraged to meet these requirements with agility?</td>
<td>☐</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Technical Acumen</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Are automated security scans included as part of continuous integration for each code commit? Do they provide a transparent, real-time view of the security posture?</td>
<td>☐</td>
</tr>
<tr>
<td>Does your security strategy address the entire technology stack for vulnerabilities, including secure containers, network, firewalls, and operating system?</td>
<td>☐</td>
</tr>
<tr>
<td>Have automated security test scripts been developed and executed to verify security features, such as authorization, authentication, field-level validation, and PII/PHI compliance?</td>
<td>☐</td>
</tr>
<tr>
<td>Does the configuration of security components, such as the perimeter firewall and intrusion detection/prevention system, follow a similar model in terms of provisioning and configuration as application servers?</td>
<td>☐</td>
</tr>
<tr>
<td>As part of the DevOps process, is advanced network monitoring in place to actively find vulnerabilities or active attacks?</td>
<td>☐</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Change Management</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Is the process of defining, implementing, and monitoring security an iterative cycle throughout the development and maintenance lifecycle of the software? Is the team providing constant feedback, reevaluation, maturation, and evolution of secure software?</td>
<td>☐</td>
</tr>
<tr>
<td>Is the project or organization employing agile coaching to drive organizational or project-level change management?</td>
<td>☐</td>
</tr>
<tr>
<td>Have appropriate organizational resources been allocated to sponsor, measure, and reinforce the implementation of security standards as a part of agile development activities?</td>
<td>☐</td>
</tr>
<tr>
<td>Is the delivery team addressing security concerns as part of traditional agile ceremonies and practices (e.g., standups, information radiators, story elicitation)?</td>
<td>☐</td>
</tr>
</tbody>
</table>
About Booz Allen

Booz Allen Hamilton (NYSE: BAH) has been at the forefront of strategy and technology for more than 100 years. Today, the firm provides management and technology consulting and engineering services to leading Fortune 500 corporations, governments, and not-for-profits across the globe. Booz Allen partners with public and private sector clients to solve their most difficult challenges through a combination of consulting, analytics, mission operations, technology, systems delivery, cybersecurity, engineering, and innovation expertise.

With international headquarters in McLean, Virginia, the firm employs about 22,600 people globally, and had revenue of $5.41 billion for the 12 months ended March 31, 2016. To learn more, visit www.boozallen.com.

For More Information

GARY LABOVICH
Executive Vice President
labovich_gary@bah.com
703-377-1177

DAN TUCKER
Principal
tucker_dan@bah.com
703-377-8124

BOB WILLIAMS
Chief Scientist
williams_robert_l@bah.com
843-696-9067

© 2016 Booz Allen Hamilton Inc.
C.05.027.16

www.boozallen.com/systemsdelivery