A New Model to ATO: Agile Cybersecurity Engineering

Overcome Slow, Intermittent Authority to Operate Requisites with Continuous Monitoring

By James Bowlin, VP Cyber Security Engineering, NCI, Inc.
Five Principles to Help Security Keep Pace in an Agile and DevOps Culture

Federal agencies are increasingly adopting Agile frameworks and DevOps methodologies to accelerate software, system and application development, aiming to achieve the same benefits as commercial organizations: faster time to market, lower costs and more efficiencies. Moving away from traditional Waterfall models to more dynamic approaches centers on more rapid, iterative development. DevOps brings about an even higher level of continuous integration and delivery benefits. A recent MeriTalk survey, found 78 percent of federal respondents see DevOps as a means to help their agency innovate faster responses and services, which is a boon for keeping up with evolving user requirements and needs.

As agencies are embracing this evolution, they also are facing challenges with regard to incorporating security into Agile development and production cycles. These agencies often find their current security processes, workflows, and team structures are in conflict and cannot keep pace with the Agile and DevOps culture. The gains in cost, schedule and performance are overshadowed by the inability for cybersecurity professionals to adapt to fast-paced continuous improvement release cycles.

Achieve More Frequent ATOs with Continuous Monitoring

The idea of discovering defects and vulnerabilities more frequently should not be shied away from as a system owner or organization. Instead, the inherent nature of more iterative development and continuous monitoring bears more opportunity to uncover and mitigate vulnerabilities with higher frequency. The idea of continuous monitoring — a fundamental tenet of DevOps — makes it a more
fluid, ongoing timeline to achieve a higher level of security for systems and system of systems. The risk management framework (RMF) is very adaptable and allows for seamless integration into Agile methodologies. The National Institute of Standards and Technology (NIST) 800-53 controls and enhancements are constantly monitored and verified as well as major updates at least once every three years.

Moving to secure DevOps facilitated by Agile, automation and lean management techniques, helps agencies keep their systems up-to-date and accredited at the pace of innovation itself. And, this can be further enabled by virtualization of the host, network and storage systems, plus the advent of cloud and cloud-like utility infrastructures of low-cost massive capabilities.

NCI has identified five guiding principles for agencies to integrate and implement security into Agile and DevOps environments to achieve value securing code under high-tempo development.

**Principle #1 – Bake-in Security During Requirements Elicitation**

Our desire and ability to innovate is outpacing the speed with which security is being embedded into the hardware and software design process. Too often, security requirements are treated as an afterthought when gathering and documenting system development requirements. Quite often, even when security is a consideration, the security requirements are typically written in such broad brush strokes that do not specifically speak to the design or development functions set forth to be useful to the end users. Peppering in security requirements after development leaves the chance of the application and systems being at risk from origin — rendering them ineffectual.

The concept of baking in security at the IT development level is gaining traction. However, secure design and development is a long way from being the golden standard. In mission critical industries such as military operations, healthcare and financial transactional systems, the need to implement and assign a lead security engineer to help identify security requirements either in one-on-one interview sessions or as part of larger brainstorming group workshops is increasingly more imperative. To be more integrated and granular; this role should be the counterpart to the chief engineer or lead engineer. Creating a co-chair leadership structure feeds a commitment to both functionalities and includes security.

The benefit will be a more realistic and tightly integrated requirements package based upon RMF or NIST security requirements and objectives that can serve as the foundation for traceability. This model also will continue reassessing security control processes at every release to ensure accuracy and validity of the control measure.

**Principle #2 – Assign Security Engineers to Scrum Team Rosters**

It still is not uncommon to find IT engineers and cybersecurity teams segregated from one another, especially at the federal level. Cybersecurity always will be a top priority for leadership, but disparate groups will bring diverse views of the challenges that agencies face...
securing systems and program milestones. In an Agile environment, development Scrum teams aim to work more closely with, say, leadership or cross-functional departments. The security team shouldn’t be treated any differently. Pulling in security once a product has been developed or deployed may work in traditional Waterfall models, but falls flat in DevOps-minded culture.

Part of a Scrum team should include software and system security engineer leads and specialists, whose responsibility is to translate product security requirements into user stories (working alongside the Scrum master, technical lead and program manager). They will manage the user stories, perform tasks as required to deliver user story value, and propose and implement new technologies and processes to satisfy requirements.

By forcing closer collaboration between security and development, Scrum teams will gain more opportunity to receive and maximize feedback from various perspectives. This structure will help everyone make more informed planning goals — and further cultivate continuous improvement inclusive of dynamism, focus, and transparency to project planning and implementation.

**Principle #3 – Track Vulnerabilities as Defects**

A core benefit of DevOps is continuous monitoring. When looked at through the security lens, agencies can continually assess, monitor and detect security posture at the source code level — identifying threats and vulnerabilities before they become security incidents. Especially with cloud-based applications and platforms being released with exponential speed, source code analysis can be regularly monitored and analyzed to verify the security posture of the application code base. Vulnerabilities can be documented and validated, and then prioritized based on severity, likelihood and organizational impact. Tracking vulnerabilities as defects determines the residual severity and priority levels entered on the defect. With DevOps and Agile methodologies, defects are tracked very meticulously. The defects are a form of measurement and determine the quality of the product being delivered. The addition of security vulnerabilities into the defect tracking process provides the weight and visibility needed to take security to the forefront of the quality of system releases.

**Principle #4 – Focus on Systems Primed for Agile and DevOps**

Keep in mind that an Agile cybersecurity engineering approach is not for every agency program. Not all systems are created equally and, therefore, not all system environments should be moved to Agile or DevOps just for the sake of keeping up with the times. Federal agencies, in particular, should be judicious about this given the nature of many mission critical programs, such as tactical combat systems, and the vast amount of legacy code, like COBOL, still in operation today. Start by evaluating the system’s engineering capabilities, lifecycle management and advancement assessment. Ask whether or not the transition to Agile and DevOps would pay dividends or simply cause more headache, and potentially open up the system under evaluation to more vulnerabilities. Systems with a shelf life, for example, may not need to be moved over but still could benefit from applying a lighter touch of more continuous monitoring of security threats and vulnerabilities.
Principle #5 – Prioritize Fixes with Input from Security Leads

Security leads actively providing input to release prioritization may seem like a given, but because the tendency to wall off security still exists, this is not always the case. Lead security engineers should be deemed a “voting member” of the committee that decides what gets fixed and what gets pushed to the next release or patch cycle, based on their expertise and understanding of how it impacts the overall functionality and risk assessment. This principle is important and pretty simple and has minimal barriers to achieve. Why is it not commonplace then? The desire to meet program milestones and target dates remains a priority yet is often arbitrary. The emphasis should instead be on the outcome you want of high functionality with minimal risk. Security engineers can perform and translate security impact assessments that align with the short and long-term objectives.

Cybersecurity is an Endurance Sport

Agile and DevOps instill a culture of persistent and fluid changes and enhancements. Simply put, an application and system are never seen as perfect or finished — only evolving to address shifting user needs. Cybersecurity is the same. New technologies and devices create new vulnerabilities, hackers and threats. Securing mission critical federal systems becomes an exercise in endurance versus a dash to static finish lines.

A continuous monitoring program that folds in security processes, workflow and team management allows an agency to track the security state of an information system and maintain security authorization on an ongoing basis — and within weeks versus months. This type of progressive understanding of the fluid security state of systems is essential in dynamic operational environments.

Integrating a cybersecurity engineering approach into an Agile and DevOps environment provides the means and methods required to monitor a system and, more importantly, continuously positions the agency to be more forward-looking. It anticipates challenges and provides near-real-time feedback for decision support. It is a sustainable methodology that is lightweight and efficient, and provides effective means of operating within a complex project.

In a closing thought, automation and artificial intelligence (AI) are essential to the success and future of cybersecurity. Finding methods and tools for efficient and consistent testing and validation is crucial to the evolution of security validation. For example, my team at NCI is actively investigating and developing AI tools for cybersecurity evaluations and RMF assessments. These tools will be the next generation of efficiencies, ease of use and consistent results following RMF and NIST.
About the Author
James Bowlin leads NCI’s enterprise strategy for the cyber market. He has more than 15 years of experience in cybersecurity, software development, and system of systems integrations across high visibility federal healthcare programs, including serving as chief engineer for DHA’s Defense Medical Information Exchange (DMIX) Program Office and lead security engineer helping the VA integrate an Agile information assurance framework for the Veterans Benefits Management System (VBMS). Bowlin is a certified ethical hacker, certified information systems security professional (CISSP), security+ holder, and fully qualified Navy and Marine Corps validator.

NCI provides cybersecurity and advanced Agile analytics to help federal agencies securely unleash the power of their data to achieve better user experiences, enhanced predictive analytics, and more effective data sharing and interoperability. For more information, please visit NCI online at www.nciinc.com.