Aligning Acquisition With Agile Practices

The acquisition process presents challenges that an Agile or DevOps culture could alleviate.

William Drew

Tue, 03/05/2019 - 14:16

Photo credit: shylendrahoode/iStock

As federal IT attempts to move toward a mature Agile and DevOps posture and culture, it is encountering challenges that are in many ways unique to the federal government. Many of these are not technology issues, but are associated with the acquisition process and contracting of IT services. The following are a few of these challenges as well as some potential solutions that would more closely align acquisition with Agile and DevOps practices and principles.

Maintaining Core Competency Within Multi-Disciplinary Teams

Organizations that have achieved a more mature DevOps culture understand that teams should be singularly focused on delivering a product or service to their end-user community. And the composition of these teams should be multi-disciplinary, drawing from development, testing, security, operations or whatever organization and skill that’s needed to take the product from ideation to production.

It’s expected that these tight-knit teams will stay together over multiple iterations and versions of the product. They will continually learn from their successes as well as their failures forming a core competency and deep knowledge of their product and users. And while individuals may come and go, the core of the team remains.

While federal IT leaders may recognize this winning formula, the conventional approaches to contracting, acquisition and the role of federal employees offer some special challenges.

Traditionally, government IT contracts are structured with contractors providing the development and delivery of products and services with federal employees largely in a managerial and oversight role. Once the contractor has transitioned the product to operations, the core knowledge and technical competency to effectively evolve the product further are essentially gone.

Federal acquisition needs to form these product-focused teams as well as build and maintain long-term hands-on technical competency in their federal employees.

Contracts should stipulate a set of key roles that must be maintained throughout the life of a product, and these must be filled largely by federal employees. This ensures that as contractors transition on and off the project, the critical core competency is maintained within the government. This not only ensures that the product can be maintained and evolve as needed, but also allows for greater insight and real visibility into what is being developed and delivered.

Define Leadership Roles and Expectations for Each Environment

In a standard delivery pipeline, there will be multiple environments such as development, testing, pre-production, staging and production. And at any point in time, there may be several active release candidates at different stages or environments in this pipeline. Along with active development, there may be a release candidate in testing, another in pre-prod and even a third in staging.

So, while its important to define the roles in these multi-dimensional teams, it’s also important to understand that the leadership expectations for these roles will change depending on the environment in which a person is working.

For example, in the development environment, developers and testers will assume the leadership role while security and operations will take a consultative and overview role. In higher environments such as pre-production or staging these roles are reversed with operations and security assuming a leadership role with development in a support role.

It’s not only important to form a multi-dimensional team with the necessary skill set and explicitly state the expectations in terms of tasks and deliverables, but also the expectations in terms of leadership in each environment must be specifically stated in order to optimize their contribution to quality and velocity.

Provide Development and Delivery Platforms

Tools selection always seems to be the first step when addressing development and delivery. Organizations, especially technical support and operations begin by designating a specific set of tools that each and every team are allowed to use to develop and deliver their product.

But instead of spending time and energy identifying this “golden set” of tools that will serve all projects, organizations should instead direct their energy toward providing the platform upon which teams can develop and deliver products.

This platform would be in the form of the development and delivery pipelines — the CI/CD of DevOps — engineered and implemented using well-defined requirements and design principles.

And given the “Cloud First” initiative, these pipelines should be provisioned and maintained via Infrastructure as Code (IaC), which can be controlled and maintained by government operations staff. This set of code would be the basis for provisioning a consistent, well-understood development and delivery pipeline across all projects in a predictable, repeatable manner.

This IaC also provides the necessary and expected extension points allowing the incorporation of a wide range of tools. And if the pipeline doesn’t support a required tool or tools, the IaC can be modified (with government oversight and approval) in order to allow for additional tools and capabilities for specific projects.

It’s still important to review and provide oversight on the tools stack that a product team is intending to use for development and delivery. But the real focus and energy instead should be given to engineering, specifying and providing a core implementation of the development and delivery pipelines.

Apply Appropriate Security Levels to Facilitate Development and Delivery

Although it’s important to establish these development and delivery pipelines, they’re of limited use if product teams cannot easily access and use them.

Too often the government applies a “one size fits all” approach to security including the lower environments of development and testing. This results in little or no real additional security for the organization and simply adds unnecessary restrictions and inhibits and slows development and delivery.

Easy access and availability to the components on these platforms and its capabilities are critical for product teams to meet the expectations of development and in particular delivery. Adopting a security posture with the appropriate level of security is critical in ensuring this access.

Contracting should detail and document the appropriate level of security along with the necessary restrictions on PII and other sensitive data for each environment and make each product team responsible for adhering to and supporting these rules.

Contracting Should Apply Agile Principles

Federal IT has fully embraced the concept of Agile development. A core principle of the Agile approach is that it's unrealistic to expect that all requirements are known upfront; requirements will emerge and evolve as we understand more and more of what the user expects and needs in their end product.

As new requirements emerge in this fluid environment, many teams find themselves lacking the necessary skills and/or knowledge. And while these skills and knowledge may be present in other teams within the project, contract boundaries prohibit true collaboration beyond simple information sharing.

Government contracting needs to match the same agile, fluid expectations of the development and delivery of products. A first step in this direction would be to provide funding for these unknown but expected changes and discoveries in requirements. Along with setting aside these funds, there should be a streamlined approach established for allocating funds once these tasks are identified and scoped.

Provide a Clear Definition of 'Done'

More often than not, the definition for what constitutes a capability or functionality being complete and accepted or "done" is left up to the project to define. This definition is generally driven by development teams where a capability is considered "done" as soon as a product owner has accepted a story or capability. This is usually well before the code has a chance to be deployed for review and inspection.

This short-sided definition of “done” has several implications. It not only overstates the delivery and development metrics in terms of story points completed, but also bypasses the critical feedback from the inspection of release candidates produced by the continuous integration (CI) pipeline. These fine-grained cross corrections enable product teams to practice true agile product development.

The definition of done is far too important to be left up to teams to define on their own. “Done” should be explicitly stated and spelled out in the acquisition process, communicated to the product teams and enforced by project management. Without a clear, consistent application of this definition, it’s impossible to determine and fine tune a team’s true delivery velocity.

Federal IT has some unique challenges when it comes to moving toward a mature DevOps culture. The traditional approaches to acquisition and contracting have proven to be one of the hurdles in that journey. But with some reasonable and practicable adjustments, acquisitions can move from being a barrier to a facilitator in that process.

View printer friendly version
agile
agile advocate
DevOps
cloud migration
Standard