SYM-AM-15-095

PROCEEDINGS
OF THE
TWELFTH ANNUAL ACQUISITION RESEARCH SYMPOSIUM

THURSDAY SESSIONS
VOLUME II

Addressing the Barriers to Agile Development in the Department of Defense: Program Structure, Requirements, and Contracting

Su Chang, MITRE Corporation
Pete Modigliani, MITRE Corporation

Published April 30, 2015

Disclaimer: The views represented in this report are those of the author and do not reflect the official policy position of the Navy, the Department of Defense, or the federal government.

ACQUISITION RESEARCH PROGRAM
GRADUATE SCHOOL OF BUSINESS & PUBLIC POLICY
NAVAL POSTGRADUATE SCHOOL

The research presented in this report was supported by the Acquisition Research Program of the Graduate School of Business & Public Policy at the Naval Postgraduate School.

To request defense acquisition research, to become a research sponsor, or to print additional copies of reports, please contact any of the staff listed on the Acquisition Research Program website (www.acquisitionresearch.net).

ACQUISITION RESEARCH PROGRAM
GRADUATE SCHOOL OF BUSINESS & PUBLIC POLICY
NAVAL POSTGRADUATE SCHOOL

Addressing the Barriers to Agile Development in the Department of Defense: Program Structure, Requirements, and Contracting[1]

Su Chang—is a principal economic and business analyst at The MITRE Corporation, specializing in IT acquisition. As MITRE's Contracting Technical Team co-leader, she fosters collaboration and professional development of MITRE's contracting and acquisition analysts. Prior to joining MITRE, Chang was a senior contract negotiator with the Missile Defense Agency, and is a graduate of the Department of Interior, Government-wide Acquisition Intern Program. She holds a BS in economics from the University of Utah, and an MA in U.S. foreign policy from American University. She is DAWIA Certified Level III in contracting.

Pete Modigliani—is the acquisition innovation area lead at The MITRE Corporation. He supports DoD acquisition and CIO executives' strategic initiatives in Agile, cyber, IT, and services acquisition. He manages a research portfolio to foster innovative acquisition solutions. Previously, as an assistant vice president with Alion, he supported the Air Force Acquisition Executive on C4ISR systems. As an Air Force program manager, he developed strategies for billion-dollar acquisitions. Pete holds a BS in industrial engineering from the Rochester Institute of Technology and an MBA from Boston College. He is DAWIA Certified Level III in program management. [pmodigliani@mitre.org]

Abstract

Program managers and executives in the Department of Defense (DoD) have struggled for years to tailor the acquisition framework to promote delivery of information technology (IT) capabilities in small, frequent releases—the approach that characterizes Agile development. DoD acquisition professionals increasingly recognize the potential of Agile methods, but do not know *how* to apply Agile within the unique and complex defense acquisition environment. Several aspects of the defense acquisition process have proven especially challenging in the implementation of Agile practices. For example, the lack of knowledge about how to tailor the DoD Instruction (DoDI) 5000.02 process for an Agile development can deter a program from considering the use of Agile techniques in the first place. Many DoD IT acquisition programs are unfamiliar with the IT Box requirements concept, and thus cannot take advantage of its flexibilities to enable Agile development. In addition, long contracting timelines and the tendency to lock down Agile requirements in a contract have become barriers to implementing the speed and flexibility necessary for successful Agile adoption. This paper offers specific acquisition solutions and strategies to address these identified "high barriers" to Agile development in DoD.

Introduction

Agile software development practices integrate planning, design, development, and testing into an iterative lifecycle to deliver software at frequent intervals. Structuring programs and processes around small, frequent Agile releases enables responsiveness to changes in operations, technologies, and budgets. These frequent iterations effectively

[1] Approved for public release; distribution unlimited. Case Number 15-0905. The author's affiliation with The MITRE Corporation is provided for identification purposes only, and is not intended to convey or imply MITRE's concurrence with, or support for, the positions, opinions or viewpoints expressed by the author.

measure progress, reduce technical and programmatic risk, and respond to feedback and changes more quickly than traditional waterfall methods (Modigliani & Chang, 2014).

While the commercial sector has broadly adopted Agile development to rapidly and dynamically deliver software capability, Agile has just begun to take root in DoD acquisitions (Lapham et al., 2010; Northern et al., 2010). A dozen or more DoD IT acquisition programs have incorporated Agile concepts and practices. These early adopters, like any new venture, have experienced mixed results. Furthermore, despite the early adoption of Agile across several DoD IT acquisition programs, the DoD has issued no formal guidance and training on DoD Agile practices. Many acquisition professionals see the value and promise of Agile, yet struggle to incorporate it effectively in the Defense Acquisition Framework. Given that Agile in many ways differs so radically from the DoD's traditional development practices, programs interested in using Agile encounter several challenges and barriers within the DoD acquisition system (Broadus, 2013).

MITRE performed initial research to examine the leading Agile methodologies and commercial practices and explore how DoD acquisition structure and processes could be tailored to adopt Agile. The resulting *Defense Agile Acquisition Guide*[2] provides acquisition professionals guidance and instruction for Agile adoption. Following the release of the guide in February 2014, MITRE conducted further research to capture best practices and lessons learned from early adopters across the DoD and other federal agencies. The research, based on years of experience and collaboration across Agile and IT acquisition communities, refined and extended strategies for tailoring each functional area of acquisition. This paper focuses specifically on three of the most difficult barriers to successful DoD Agile adoption: program structure, requirements, and contracting. The DoD can address these barriers by utilizing a proactively tailored Agile acquisition model, implementing an IT Box requirements process, and utilizing the flexible contracting approaches described in this paper.

The first half of this paper provides an overview of the Agile development process and identifies some of the primary challenges in adapting commercial Agile practices for DoD implementations. Next, the paper examines prerequisites for effective adoption of Agile practices in the DoD. The remaining sections describe each of the three "high barrier" problem areas and offer specific recommendations that the DoD can use today to overcome these challenges.

Background

Agile Development Overview

Agile software development emerged in 2001 after 17 industry leaders created the Agile Manifesto to design and share better ways to develop software (Agile Manifesto, n.d.). Agile prioritizes early and continuous deliveries of working software; adapts easily to changing requirements; depends on small, empowered teams; and promotes active user involvement during development.

Agile development can be distilled into four core elements:

[2] A copy of the guidebook can be obtained at http://www.mitre.org/publications/technical-papers/defense-agile-acquisition-guide-tailoring-dod-it-acquisition-program

- Focusing on small, frequent capability releases
- Valuing working software over comprehensive documentation
- Responding rapidly to changes in operations, technology, and budgets
- Actively involving users throughout the development process to ensure high operational value

The foundation of Agile is a culture of small, dynamic, empowered teams actively collaborating with stakeholders throughout product development. Agile development requires team members to follow disciplined processes that require training, guidance, and openness to change (GAO, 2012). While Agile does impose some rigor, the method does not consist of simply following a set of prescribed processes, but instead allows dynamic, tailored, and rapidly evolving approaches that suit each organization's IT environment.

Various Agile methods (e.g., Scrum, Extreme Programming (XP), Kanban, Test Driven Development) have emerged, each with unique processes, terms, and techniques (Modigliani & Chang, 2014). These methods focus on the development team and associated stakeholders. Agile Acquisition extends Agile development practices beyond the contractor development team to the government acquirer, users, and other stakeholders. It requires that both agencies and contractors change many acquisition roles, processes, and culture, thereby fostering a close government–industry partnership (Balter, 2011; Lapham et al., 2010). This, in turn, demands investments in time, training, and continuous improvement to pay long-term dividends. While both practical experience and research findings strongly indicate the value of Agile acquisition for many IT development programs, this approach may not be appropriate in all cases (Lapham et al., 2010). Programs should consider Agile Acquisition when

- Users can decompose requirements into small tasks for iterative development.
- The operational environment can support small, frequent capability deliveries.
- Users can engage throughout development to capture concepts of operations (CONOPS) and provide feedback on demonstrated capabilities.
- Program can use existing infrastructure and focus development on the application layer.
- Industry partners are available with relevant domain expertise in Agile practices.
- Milestone Decision Authority supports Agile development practices and tailored processes.

Agile and the DoD Acquisition Environment

Despite the success that Agile development has achieved in the private sector, commercial implementation of Agile does not directly translate to Agile adoption in the federal sector. The barriers to program structure, requirements, and contracting often stem from these key differences. First, the government must adhere to a set of rigorous policies, statutes, and regulations that do not apply to the same degree to the commercial sector (Lapham et al., 2011). Following the rules that govern federal acquisition often involves a bureaucratic, laborious, and slow process that greatly influences how effectively the DoD can implement Agile. Second, the commercial sector has a different stakeholder management process than the government. Private firms are accountable to an internal and layered management structure that usually goes no higher than a corporate board of directors; the few possible external stakeholders (e.g., labor unions) rarely cause frequent and major disruptions. The government bureaucracy has layers upon layers of stakeholders with a high degree of influence that can create frequent and significant disruptions. Everything from a change in the political administration to budget sequestration can exert significant external influence on a DoD program. Lastly, the bureaucratic layers of government make it difficult to empower Agile teams to the same extent as in the private sector. The commercial sector has considerable latitude to make adjustments throughout the course of the development because companies closely link accountability, authority, and responsibilities to push decision-making to the lowest levels. The government's tiered management chain of command makes it difficult for the Agile team to make decisions quickly and unilaterally.

The above comparisons demonstrate the need for the DoD to tailor Agile processes to its unique set of policies and laws. Herein lies the fundamental issue with Agile adoption in the DoD. The practices, processes, and culture that have made Agile development successful in the commercial sector often run counter to the current practices, processes, and culture in the long-established defense acquisition enterprise (Broadus, 2013). In many ways, the acquisition environment needed to execute Agile development is the opposite of the acquisition environment in place today.

- The small, frequent capability releases that characterize the Agile development approach directly contrast with the traditional DoD acquisition model designed for a single big-bang waterfall approach (Broadus, 2013). Currently, every step in the acquisition system must be extensively documented and approved prior to execution. For example, according to DoDI 5000.02, a DoD IT acquisition program must meet 34 statutory and regulatory documentation requirements prior to entering Milestone A (Defense Acquisition University, 2015), whereas Agile emphasizes working software over comprehensive documentation (Lapham, 2012).
- Agile also enables rapid response to changes in operations, technology, and budgets. By contrast, the DoD requires budgets, requirements, and acquisitions to be planned up front, sometimes several years in advance of execution, and changing requirements, budgets, and strategies during the execution process is disruptive, time-consuming, and costly (Modigliani & Chang, 2014).
- Lastly, Agile values active involvement of users throughout the development process to ensure high operational value, and continuously re-prioritizes the ongoing requirements process on the basis of feedback from the user community on deployed capabilities. Today's DoD requirements process is static, rigid, and limits active user involvement and feedback during the development process (Lapham et al., 2010).

Given these key differences, the DoD has been ill prepared to adopt Agile development practices and in fact Agile implementations so far have not always succeeded. Some early DoD adopters attempted what they thought or promoted as "Agile," yet they did not incorporate some of the foundational Agile elements into their structures or strategies. This resulted partly from the lack of definition and standardized processes for Agile in the federal sector. In some cases, programs implemented a few Agile principles, such as breaking large requirements into smaller increments, but did not integrate users during the development process to provide insight or feedback. Other programs structured capability releases in a time-boxed manner,[3] yet did not understand what to do when releases could not be completed in time.

Adopting only a handful of Agile practices without a broader Agile strategy often fails to achieve desired results (GAO, 2012). For example, one DoD early adopter initially attempted to implement Agile practices by breaking large requirements into several four-week sprint cycles. However, the program lacked high-level agreement on what to develop in each cycle, and did not have a robust requirements identification and planning process in place. Furthermore, the program lacked an organized user community and active user-participation throughout the development process—a fundamental Agile tenet. As a result, the Agile processes quickly degenerated and the program only delivered 10% of its objective capability after two years of failed Agile development attempts. The program finally retreated to a waterfall-based process. It simply could not execute the Agile strategy without the proper environment, foundation, and processes in place. On the other hand, the DoD has recorded some significant successes with Agile, such as the Global Combat Support System–Joint (GCSS-J) program, which has routinely developed, tested, and fielded new functionality and enhanced capabilities in six-month deployments (Defense Information Systems Agency, 2015).

Prerequisites for Agile Adoption

The Agile model represents such a radical change in the way the DoD conducts business that the DoD must actively rethink how programs are managed and structured to support Agile (Modigliani & Chang, 2014). This requires restructuring the current acquisition environment (i.e., policies, processes, and culture) to enable success.

As a starting point, the DoD should adopt a common understanding of Agile and identify the underlying set of values that describe the purpose and meaning of DoD Agile practices. The authors propose the following guiding principles for DoD Agile adoption:

1. **Focus on small, frequent capability releases to users**—Smaller releases are easier to plan, present lower risks, and are more responsive to changes. Projects should focus on delivering working software as the primary objective.
2. **Embrace change**—Projects must allow for changes to scope and requirements based on operational priorities, user feedback, early developments, budgets, technologies, etc. This requires flexible contracts, strong collaboration, and rigorous processes. Projects should plan early and then adapt based on current conditions.
3. **Establish a partnership between the requirements, acquisition, and contractor communities**—Projects should foster active collaboration on operations, technologies, costs, designs, and solutions. This requires committed users who contribute to development, tradeoff discussions, and regular demonstrations of capabilities.

[3] A time-box is a fixed time period allocated to each planned activity. For example, within Agile, a sprint is often time-boxed to a 4–6 week time period or a release is time-boxed for a 4–6 month time frame.

4. **Rely on small, empowered, high-performing teams to achieve great results**—Organizing around each release with streamlined processes and decisions enables faster deliveries that are more successful.
5. **Leverage a portfolio structure**—Individual programs and releases can deliver capabilities faster by using portfolio or enterprise strategies, processes, architectures, resources, and contracts.

These tenets align with the recommended set of principles in the Government Accountability Office (GAO) report on Ef*fective Practices and Federal Challenges in Applying Agile Methods.* They center on the Agile Manifesto themes of small, frequent capability releases, a dynamic requirements process that allows for the continuous prioritization of requirements, active involvement from the user community throughout the development process, and commitment to delivering working software based on a time-boxed schedule. Some efforts may succeed in implementing only a subset of these themes and delivering effective software solutions; however, one could argue that this would not constitute a pure Agile development.

The DoD would benefit from defining and standardizing Agile-based practices to ensure a Department-wide consistent and common understanding of what constitutes an Agile-based DoD program or project (Lapham et al., 2010). Today, many efforts are inaccurately labeled as Agile, leading to misunderstanding and misrepresentation of Agile principles. After defining the principles, the DoD needs to provide detailed guidance to the acquisition community that describes how to execute the Agile acquisition processes within DoD acquisition regulations and laws (Broadus, 2013). This level of detailed process-level guidance falls outside the scope of this paper, but the *Defense Agile Acquisition Guide* offers further details on the guidance needed for the DoD to make Agile adoption effective and widespread. This paper centers on the aspects of the DoD acquisition process that have proven most problematic when implementing Agile development concepts. The following sections focus on three of the most difficult barriers for DoD Agile adoptions: program structures, requirements, and contracting.

Structuring an Agile Program

Structuring a program for Agile development differs significantly from structuring an IT program around a traditional development methodology. Traditional waterfall programs usually have discrete acquisition phases driven by milestone events to deliver a large capability. Agile is more dynamic and requires the program to be structured to support multiple, small capability releases (Lapham, 2012).

Structuring an Agile program in this way represents a fundamental first step in developing a strategy for program-level Agile adoption. This activity requires the program to make significant adaptations to the traditional DoDI 5000.02 program structures and acquisition processes to support Agile development timelines and objectives (Modigliani & Chang, 2014). Although the DoDI 5000.02 acquisition policy places heavy emphasis on tailoring acquisition models to meet program needs, programs often do not know how to do so effectively and receive approval from process owners (Modigliani & Chang, 2014). It takes years of experience to truly understand the nuances involved in tailoring an acquisition program. Given the radical differences between Agile and a traditional development model, programs often view this activity as too complicated and therefore fail even to consider the Agile development process for a program. Programs must be designed in such a way that they not only meet all the DoDI 5000.02 statutory and regulatory requirements, but are also executable and marketable to senior acquisition executives who may be unfamiliar with the details of the Agile process. The following sections describe a recommended approach to structuring an Agile DoD program, starting with the process to structure an Agile release and building on this concept to develop a fully tailored DoD Agile acquisition program.

Agile Releases and Potential Program Structures

When developing an Agile program structure, a program should first decide how to structure its releases. The release represents the core element of the program structure, guiding how frequently the program delivers capabilities to the warfighters. The length of each release depends upon operational, acquisition, and technical factors. As a general guideline, most releases should take less than 18 months, with a goal of 6–12 month timelines. Program offices should tailor acquisition processes to support these release timelines. In some cases, this requires redesigning key acquisition processes around a 6–12 month release rather than a 5–10 year increment.

Each release comprises multiple sprints and a final segment for release testing and certification. Each sprint, in turn, includes design, development, integration, and test, and culminates in demonstration of capabilities to users and other stakeholders. Developers may be required to deliver interim code to the government at the end of each sprint or multiple sprints. The government can integrate the interim code into its software environment for testing and operational assessments. Figure 1 shows a potential 6-month release structure with five monthly sprints.

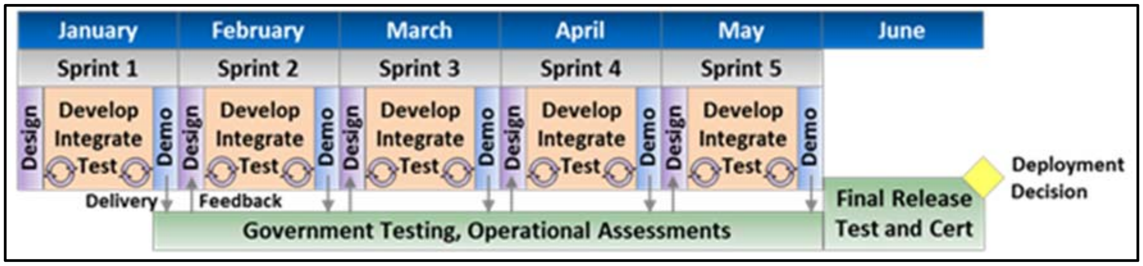


Figure 1. Potential Six-Month Release Structure

Figure 2 shows a potential 12-month release structure with seven 6-week sprints. Programs must adjust the length of the sprints and releases as conditions warrant. The key is to establish consistent, time-boxed releases, ideally small and frequent to allow for iterative development that responds easily to changes.

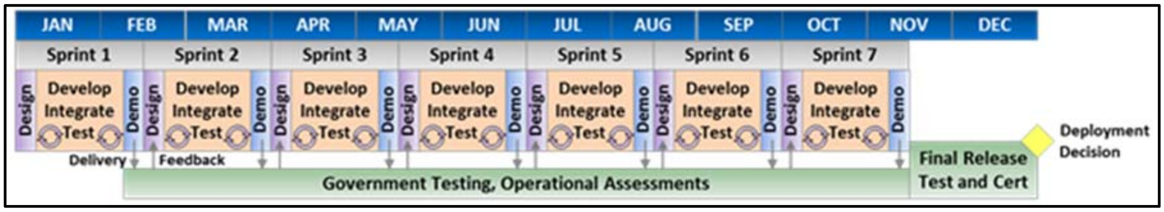


Figure 2. Potential 12-Month Release Structure

After determining a release strategy, each effort should tailor its programmatic and acquisition processes to effectively enable Agile development practices. Figure 3 illustrates one potential structure at a top level. In this approach, requirements, technology, and architecture development are continual processes rather than sequential steps in early acquisition phases. Each release involves a series of sprints to iteratively develop and test a capability, ultimately leading to capability deliveries to the warfighter every six months upon approval. Instead of bounding development via a series of increments with Milestones B and C at each end, development thus becomes a continual process. Semi-annual reviews with senior leadership and other key stakeholders ensure transparency into the program's progress, plans, and issues. Programs provide additional insight to executives via monthly or quarterly reports and other status meetings.

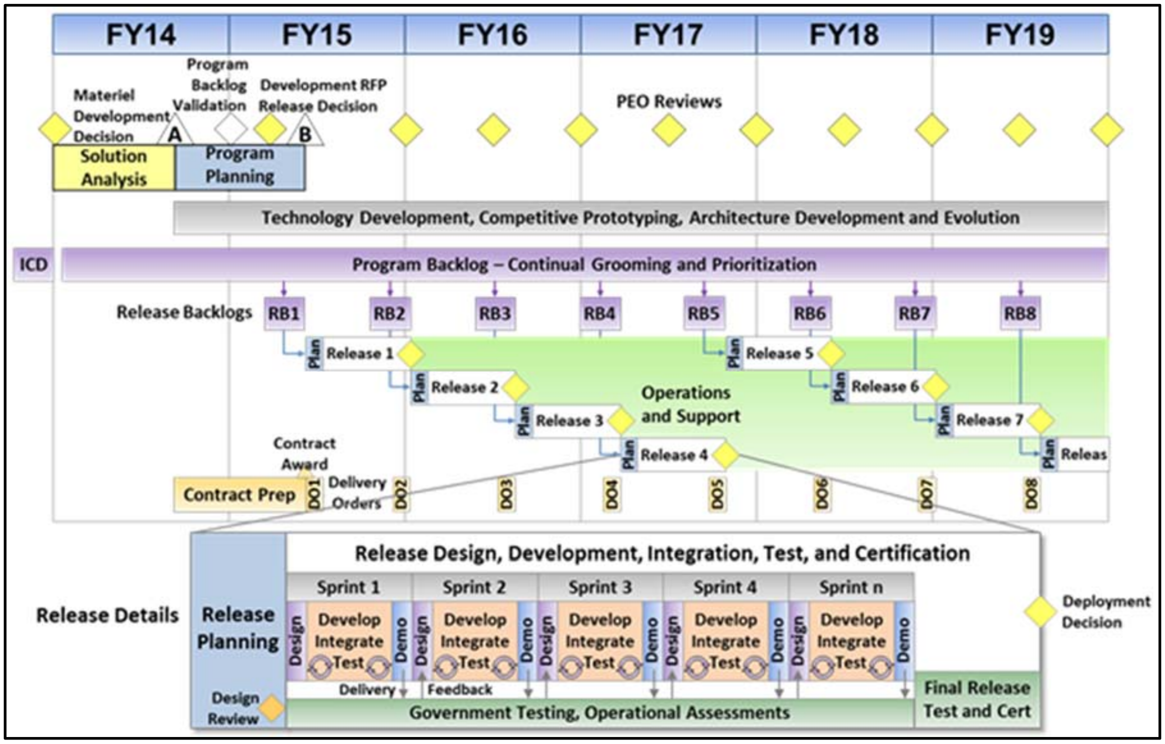


Figure 3. Potential Agile Structure

A core theme throughout DoDI 5000.02 is the tailoring of program structures and acquisition processes to meet the needs of the individual program. The policy includes several acquisition models to consider, such as Model 2 for defense-unique software, Model 3 for incrementally fielded software, and hybrid Model B for software dominant programs. Figure 4 shows a proactively tailored acquisition model based on the three software models in DoDI 5000.02. The *Defense Agile Acquisition Guide* contains more detail about the structure and accompanying acquisition processes for DoD Agile adoption.

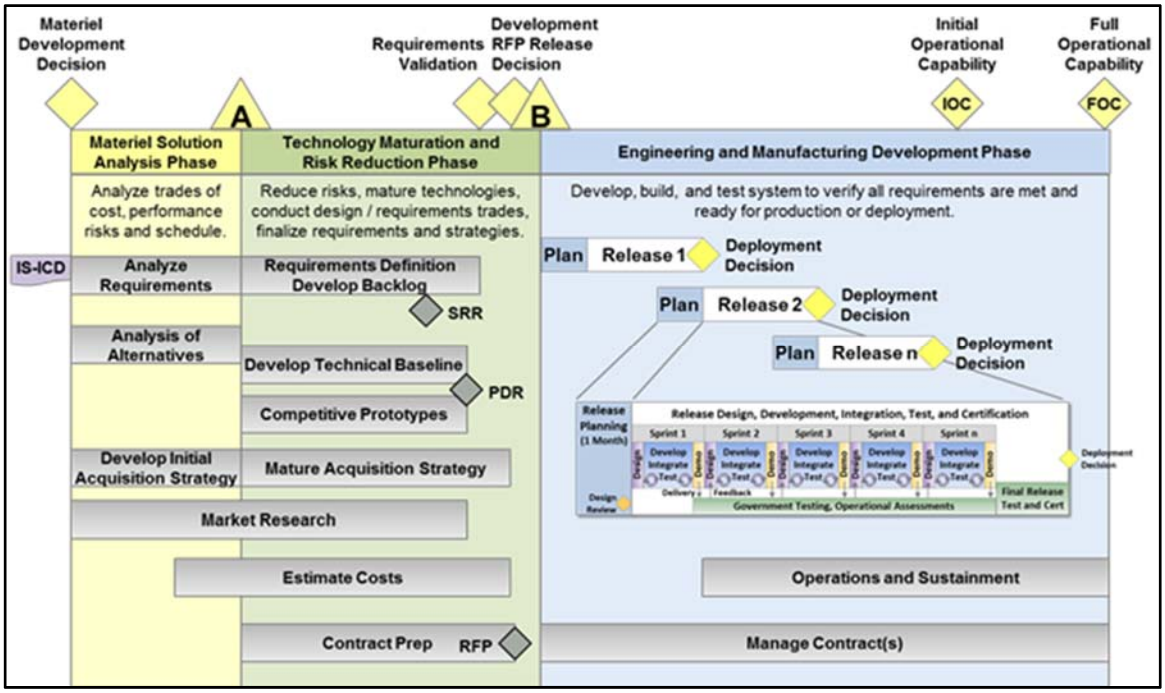

Materiel Development Decision
Requirements Validation
Development RFP Release Decision
Initial Operational Capability
Full Operational Capability
A
B
IOC
FOC
Materiel Solution Analysis Phase
Technology Maturation and Risk Reduction Phase
Engineering and Manufacturing Development Phase
Analyze trades of cost, performance risks and schedule.
Reduce risks, mature technologies, conduct design / requirements trades, finalize requirements and strategies.
Develop, build, and test system to verify all requirements are met and ready for production or deployment.
IS-ICD
Analyze Requirements
Requirements Definition Develop Backlog
SRR
Analysis of Alternatives
Develop Technical Baseline
PDR
Competitive Prototypes
Develop Initial Acquisition Strategy
Mature Acquisition Strategy
Market Research
Estimate Costs
Contract Prep
RFP
Plan
Release 1
Deployment Decision
Plan
Release 2
Deployment Decision
Plan
Release n
Deployment Decision
Release Planning (1 Month)
Release Design, Development, Integration, Test, and Certification
Sprint 1
Sprint 2
Sprint 3
Sprint 4
Sprint n
Design
Develop Integrate Test
Demo
Delivery
Feedback
Design Review
Government Testing, Operational Assessments
Final Release Test and Cert
Deployment Decision
Operations and Sustainment
Manage Contract(s)

Figure 4. Tailored Agile Model

Agile Requirements Process

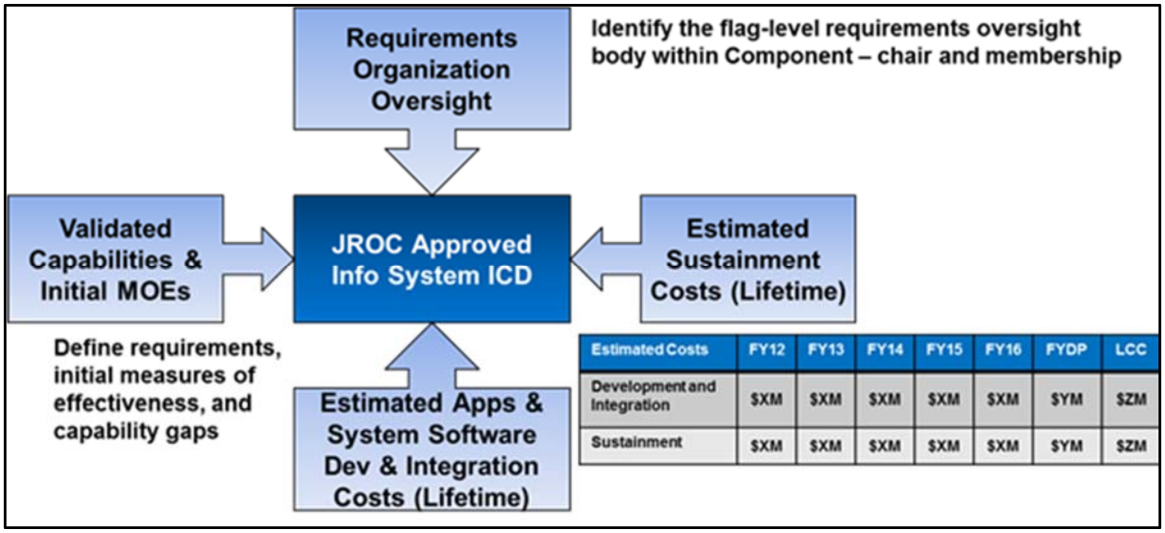
The Agile requirements process values flexibility and the ability to reprioritize requirements as a continuous activity based on user inputs and lessons learned during the development process. In contrast to current acquisition practices, the Agile methodology does not force programs to establish their full scope, requirements, and design at the start, but assumes that these will change over time.

At present, the Joint Capabilities Integration Development System (JCIDS) process guides the DoD requirements process. The traditional JCIDS process, based on lengthy and labor-intensive efforts to capture and define requirements, prevents agility (Lapham et al., 2011). The DoD has recognized that this process was particularly inappropriate to IT development because of the rapid pace of change in IT compared with the JCIDS requirements definition timeline. As a result, the DoD updated the JCIDS by approving an "IT Box" to better accommodate the dynamic nature of IT and the shortened timelines required to rapidly field IT-enabled operational capabilities (Office of the Secretary of Defense, 2010). The IT Box describes the operational performance and life-cycle affordability bounds of the program. The boundaries imposed by the "Box" expedite program initiation and streamline oversight by reducing return trips to the JROC for change approval.

However, even with the introduction of the IT Box model to provide more flexibility in the requirements process, many programs still struggle with how to apply this model to their IT development programs. As programs strive to structure their programs around Agile-based concepts as described in the previous section, they find it further confounding to figure out how to apply the IT Box model to satisfy the JCIDS requirements process. The following section contains specific recommendations on how to apply the IT Box concept to a DoD Agile development program.

Applying the IT Box Model for Agile Development

In the JCIDS IT Box model, an acquisition program develops an "IS-Initial Capabilities Document (ICD)" for JROC approval, while the traditional Capability Development Documents (CDDs) and Capability Production Documents (CPDs)[4] are no longer required. Figure 5 illustrates the four sides of the IT Box identified in the IS-ICD.


Requirements Organization Oversight
Identify the flag-level requirements oversight body within Component – chair and membership
Validated Capabilities & Initial MOEs
JROC Approved Info System ICD
Estimated Sustainment Costs (Lifetime)
Define requirements, initial measures of effectiveness, and capability gaps
Estimated Apps & System Software Dev & Integration Costs (Lifetime)

Estimated Costs	FY12	FY13	FY14	FY15	FY16	FYDP	LCC
Development and Integration	$XM	$XM	$XM	$XM	$XM	$YM	$ZM
Sustainment	$XM	$XM	$XM	$XM	$XM	$YM	$ZM

Figure 5. IT Box
(Chairman of the Joint Chiefs of Staff [CJCS], 2012)

As long as programs operate within these four sides of the IT Box, they need not return to the JROC for approval or oversight. In lieu of CDDs and CPDs, programs can develop Requirements Definition Packages (RDPs) to capture a subset of the IS ICD scope and/or Capability Drop (CD) documents for smaller items such as applications (see Figure 6).[5] Most important, the requirements documents are designed for a smaller scope of work and approval at a lower level. This flexibility and streamlining of IT requirements enables Agile development within a DoD program. Programs should take advantage of this and avoid developing a CDD or CPD. Managers can formulate the requirements process for the overarching acquisition using the JCIDS IT Box process to build in flexibility from a high-level operational standpoint. Once an Agile approach has been designed into the program, programs must ensure they establish a flexible process for managing requirements from a functional capability standpoint (Modigliani & Chang, 2014).

[4] CDDs and CPDs are traditional JCIDS requirements documents that describe the program and program increment requirements.

[5] Services and requirements oversight organizations have the flexibility to identify alternative names for these documents, along with their scope, content, and approval processes.

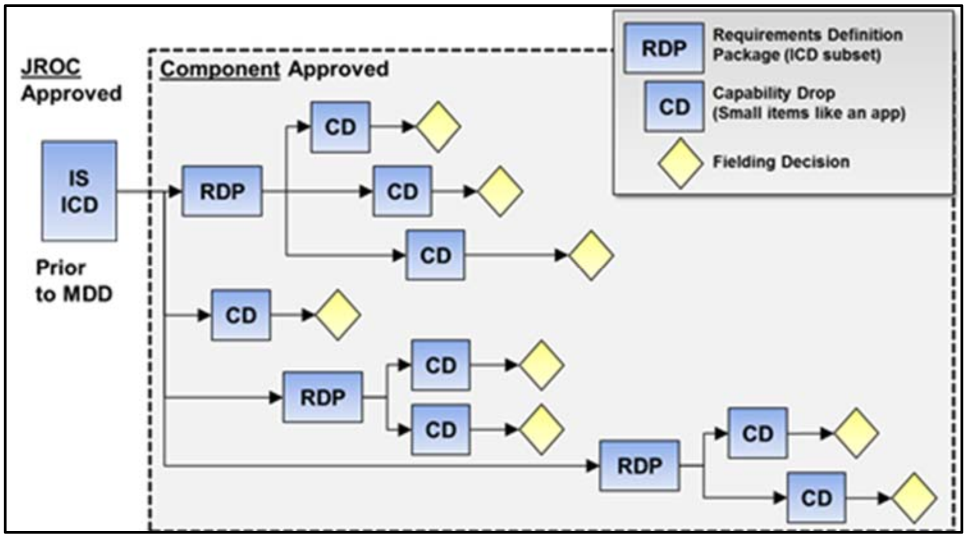

JROC Approved
IS ICD
Prior to MDD
Component Approved
RDP
CD
RDP Requirements Definition Package (ICD subset)
CD Capability Drop (Small items like an app)
Fielding Decision

Figure 6. Example of Requirements Documentation
(CJCS, 2012)

With the IT Box construct in place and the appropriate documentation requirements fulfilled, programs can manage the technical requirements in an Agile environment via program, release, and sprint backlogs.[6] Backlogs could take the form of databases, Excel spreadsheets, or Agile development software tools. The product owner, the person responsible for requirements, actively manages (grooms) program and release backlogs, working with the user community and other stakeholders to identify the greatest level of detail for the highest priority requirements.

Figure 7 shows the relationships among the program, release, and sprint backlogs. The program backlog contains all desired functionality and requirements. A release backlog typically comprises the highest priority requirements from a program backlog that a team can complete within the established timeframe. A sprint then addresses the highest priority requirements from the release backlog. Once the development team commits to the scope of work for a sprint, that scope is locked. Sprint demonstrations conducted by the contractor at the end of a sprint may identify new features or defects that the team would add to the release or program backlogs.

[6] A program backlog is the primary source of all requirements/desired functionality for the program. A release backlog is a subset of the program backlog listing features intended for the release. A sprint backlog is a subset of the release backlog listing the user stories to implement in the sprint.

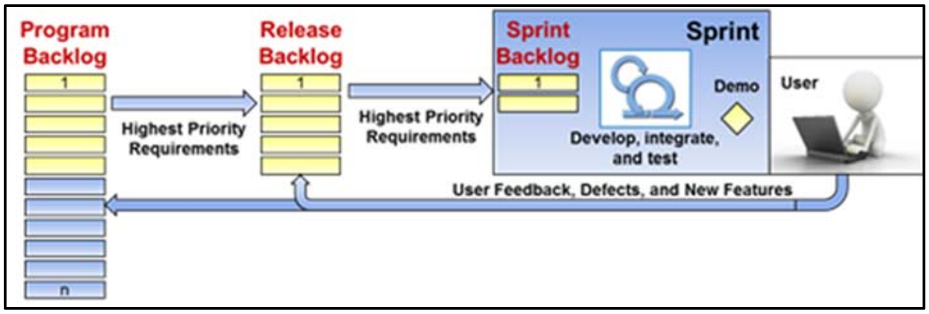

Program Backlog
1
n
Highest Priority Requirements
Release Backlog
1
Highest Priority Requirements
Sprint Backlog
1
Sprint
Demo
Develop, integrate, and test
User
User Feedback, Defects, and New Features

Figure 7. Program, Release, and Sprint Backlogs

The product owner, actively collaborating with users and stakeholders, is responsible for grooming the backlog to ensure the content and priorities remain current as teams receive feedback and learn more from developments and external factors. Users and development teams may add requirements to the program or release backlog or shift requirements between them. The release and development teams advise the product owner on the development impacts of these decisions, while users advise the release team about the operational priorities and impacts. To address a specific user story, the program must understand dependencies on existing or planned capabilities. Some programs may turn to a Change Control Board to make some of the larger backlog grooming decisions. The use of this model, combined with the IT Box structure, can help set a DoD Agile acquisition program on the right path for implementation (Modigliani & Chang, 2014).

Contracting for Agile Development

This section summarizes the difficulties of executing Agile development in the current government contracting environment and suggests available options.

Challenges

Contracting for Agile development has proven tremendously difficult not only for the DoD but also for many other federal agencies. The July 2012 GAO Report on *Effective Practices and Federal Challenges in Applying Agile Methods* cites “Procurement practices may not support Agile Projects” as a key challenge area. Contracting for Agile development presents a unique challenge to the government not often encountered in the private sector because commercial firms often rely on in-house staff to execute the Agile practices, whereas the government must obtain Agile development support through a contract arrangement.

This poses several challenges for the government. First, the government contracting process emphasizes competition and is guided by a set of policies and laws articulated in the Federal Acquisition Regulation (FAR). Government programs cannot simply choose any Agile development contractor they like, but must follow a specific set of contracting processes and protocols to obtain contracted support in a fair and transparent manner. These government contracting laws and regulations have resulted in long contracting timelines that in themselves pose significant difficulty for government implementation of Agile. A competitive IT contract can take over a year to award. This prevents execution of the Agile development process, which relies on short delivery cycles and time-boxed schedules (Lapham et al., 2011).

Next, the government contracting process requires programs to define the contract requirements upfront in a Statement of Work (SOW) or Performance Work Statement (PWS) so that a contractor can prepare a technical and cost proposal against the SOW/PWS requirements. The government uses the contractor’s proposal to determine the contract scope, schedule, and cost. Herein lies one of the biggest obstacles to Agile implementation. One of the key tenets of Agile development is a dynamic requirements process that does not lock down requirements. The government therefore confronts the very difficult challenge of figuring out how to define requirements in a SOW/PWS to award the development contract, without locking down the technical requirements to a point where the contractor has no flexibility during the execution of the Agile development process (Balter, 2011).

Following contract award, successful Agile development depends on the ability to reprioritize requirements as program staff “learn” throughout the development process and re-scope the effort as needed. Today, however, post-award management of the contract is often inconsistent. In some cases, the contractor has minimal oversight and management and government–contractor interaction occurs only during infrequent reviews. By contrast, Agile requires very close management of the government–contractor relationship, with frequent, often daily, interaction between them.

Lastly, the award of a government contract today often relies on the strength of the proposed technical solution. Under Agile, the government and contractor together determine the technical solution in the course of executing the Agile development process. Thus, contract award should be based on the strength of the development team and the team’s experience using Agile practices.

Solutions

Given the disparity between traditional contracting practices and the needs of Agile, the government has encountered difficulties in contracting for Agile development. However, programs should consider the following solutions.

First, programs must plan contracts well in advance of the proposed Agile development. In many cases today, contracting can become the long-lead item in the development process if it is not properly considered in the upfront planning process.

Second, the program must determine if it will use a service or a product contract. A service contract is highly recommended because this vehicle would provide the program with greater flexibility to modify requirements along the development process (Modigliani & Chang, 2014). A service contract is more flexible for Agile efforts than a product contract because it describes requirements in terms of the people and time required to execute the development process rather than locking down the technical details of the end-product deliverable. However, this strategy assumes the program is the lead systems integrator and is responsible for overall product rollout and delivery. If the government expects the contractor to act as the systems integrator, determine the release schedule, and be accountable for overall product delivery, then a product-based contract in which the government describes overall delivery outcomes and objectives is more practical. However, this scenario would make it difficult for the government to execute a true Agile process, because changes to requirements in the course of development, or a change to the delivery schedule, will require a contract negotiation that could affect the Agile process. If the government does execute a product-based contract, it should pursue an indefinite-delivery, indefinite-quantity (IDIQ) contract vehicle and define product-based task orders based on either the release or the sprint level, depending on which level has the best-defined technical requirements (e.g., user stories). The program must carefully balance the advantages of a service versus a product contract based on a determination of government versus contractor responsibilities for the Agile processes.

Next, the program must determine the type of contract vehicle and strategy. Some cases require a separate stand-alone contract; in others, the government could leverage an existing contract vehicle. Programs must conduct thorough market research to determine if an existing contract vehicle could meet their needs. When analyzing existing contract vehicles, a program must review the contract scope to ensure it can support Agile processes and evaluate the capabilities of the contract awardees to determine if they have Agile expertise and experience and if the labor categories and rates are compatible with the program's level of complexity (Lapham et al., 2011).

Lastly, the program should focus on the competition strategy to be used for the initial award as well as for follow-on task orders and awards. This will help determine how to scope the contract or task order for each contract action. In some cases, the program would benefit from bundling a set of releases into a single contract action to minimize the number of contract activities during the development process. However, the program should balance this against the need to maintain continuous competition throughout the program lifecycle to keep rates low and receive the best value for products and services.

Using a Contract Vehicle to Support Agile Program Structure

As stated above, a services contract may represent a good strategy for a program seeking to acquire the skills and expertise of a developer to participate in a government-led Agile team. The program can pursue a separate stand-alone contract for Agile support services, or can consider leveraging an existing contract vehicle such as a GSA Schedule to acquire Agile support services on a task order basis. This strategy works well for a program that will need consistent Agile support to develop a single product, but is not recommended when pursuing a product-based contract, because the program would have to define requirements too far in the development process to gain the benefits of an Agile process (Modigliani & Chang, 2014). As illustrated in Figure 8, such a program would require consistent support throughout the development of several release cycles.

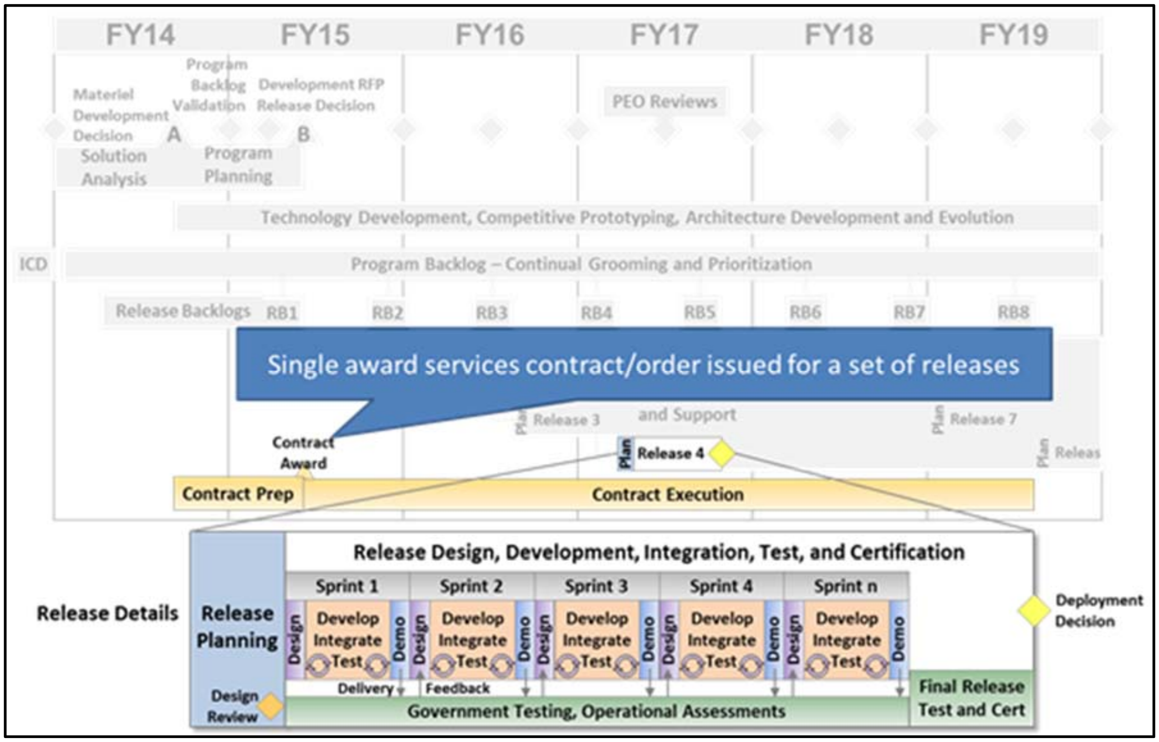

FY14
FY15
FY16
FY17
FY18
FY19
Materiel Development Decision
Program Backlog Validation
Development RFP Release Decision
A
B
PEO Reviews
Solution Analysis
Program Planning
Technology Development, Competitive Prototyping, Architecture Development and Evolution
ICD
Program Backlog – Continual Grooming and Prioritization
Release Backlogs
RB1
RB2
RB3
RB4
RB5
RB6
RB7
RB8
Single award services contract/order issued for a set of releases
Plan
Release 3
and Support
Plan
Release 7
Contract Award
Plan
Release 4
Plan
Releas
Contract Prep
Contract Execution
Release Details
Release Planning
Release Design, Development, Integration, Test, and Certification
Sprint 1
Sprint 2
Sprint 3
Sprint 4
Sprint n
Design
Develop Integrate Test
Demo
Delivery
Feedback
Deployment Decision
Design Review
Government Testing, Operational Assessments
Final Release Test and Cert

Figure 8. Single-Award Services Contract

A multiple-award IDIQ contract can allow a program to use several development contractors. This strategy would enable the program to maintain continuous competition for future task orders and/or execute parallel development. Under this strategy, the program would award IDIQ contracts to two or more qualified vendors to compete on individual task orders, as illustrated in Figure 9. The program office would have to work closely with the contracting office to streamline contract timelines to enable rapid execution of task orders. This could be achieved by using standardized business practices, templates, and streamlined selection criteria. Past performance on task orders would become a weighted selection criterion for future work, further motivating contractor performance.

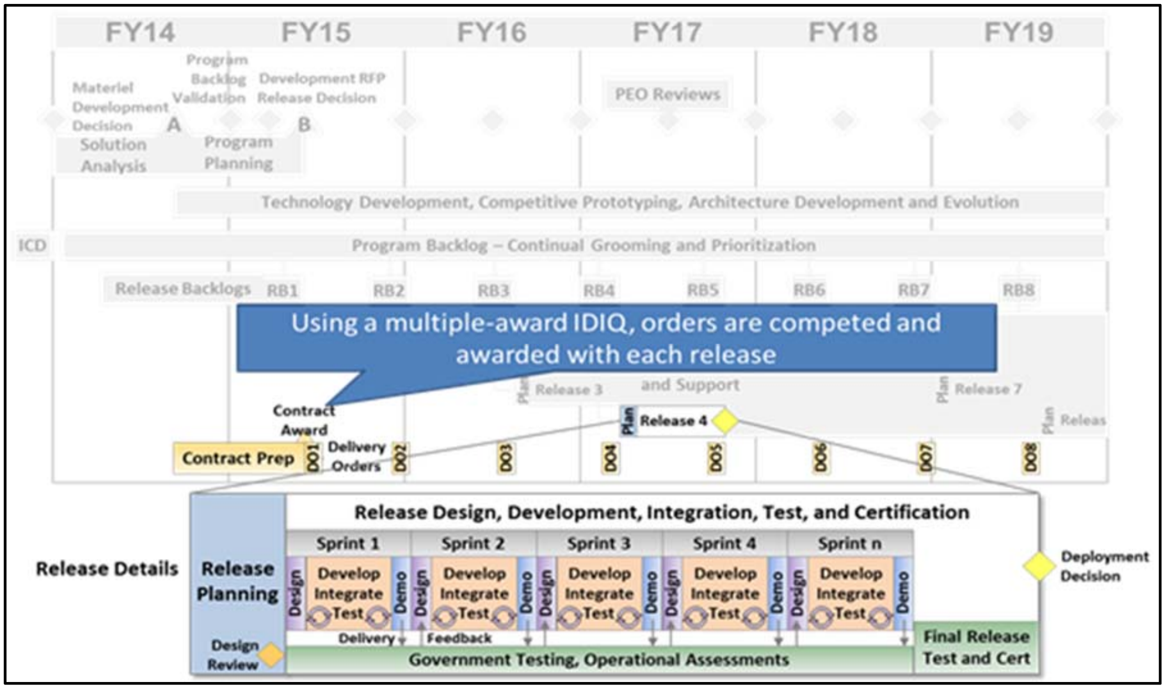

FY14
FY15
FY16
FY17
FY18
FY19
Materiel Development Decision
A
Program Backlog Validation
Development RFP Release Decision
B
PEO Reviews
Solution Analysis
Program Planning
Technology Development, Competitive Prototyping, Architecture Development and Evolution
ICD
Program Backlog – Continual Grooming and Prioritization
Release Backlogs
RB1
RB2
RB3
RB4
RB5
RB6
RB7
RB8
Using a multiple-award IDIQ, orders are competed and awarded with each release
Plan
Release 3
and Support
Release 7
Releas
Contract Award
Plan
Release 4
Contract Prep
DO1
Delivery Orders
DO2
DO3
DO4
DO5
DO6
DO7
DO8
Release Details
Release Planning
Release Design, Development, Integration, Test, and Certification
Sprint 1
Sprint 2
Sprint 3
Sprint 4
Sprint n
Design
Develop Integrate Test
Demo
Delivery
Feedback
Design Review
Government Testing, Operational Assessments
Final Release Test and Cert
Deployment Decision

Figure 9. Multiple-Award IDIQ Strategy

However, this strategy can also complicate integration and require increased resources to award and manage multiple contracts and developments (Lapham et al., 2011). To mitigate the integration risks of using two or more vendors, the government must dedicate time and effort to developing a rigorous architecture, interfaces, standards, and systems engineering processes. Each vendor should have active representation on the systems engineering Integrated Product Team to ensure a common understanding and maturation of these systems engineering elements throughout development. To foster coordination across vendors, the program should require the use of a common tool suite in the Request for Proposals process, and should also identify an initial set of required metrics each vendor must collect and report. In accordance with the contract, within the first 90 days of contract award, the vendors must submit to the program office an agreed-upon updated set of metrics proposed for review and approval.

If the program has reached a more mature stage of development with clearly defined releases, it may be feasible to execute product-based task orders. If requirements are dynamic and the program is in the initial stages of executing Agile, it would make more sense to use a service task order and compete the task orders for a set of releases.

Summary

The focus on iterative development and frequent capability deployments makes Agile an attractive option for many DoD IT acquisition programs, especially time-sensitive and mission-critical systems. However, Agile differs so profoundly from traditional development practices that the DoD must overcome significant challenges to foster greater Agile adoption. The DoD cannot expect individual programs to tailor current acquisition processes on their own, because the complexities of the DoDI 5000.02 processes do not lend themselves to obvious solutions, let alone solutions that accommodate processes so fundamentally different from current DoD practices. Following the guidance offered in this paper would better equip programs to tailor the DoDI 5000.02 for Agile execution. As they face the next challenge of defining requirements in a way that meets rigorous JCIDS standards, programs can use the IT Box model outlined in this paper to enable the speed and flexibility required for Agile requirements. Lastly, programs can utilize the contracting strategies presented in this paper to acquire development support and utilize flexible contract vehicles that support Agile practices.

This paper has offered potential solutions to these key challenges in order to aid programs in laying a foundation for successful Agile implementation. As Agile adoption continues to take root and expand across programs, the DoD would benefit from additional guidance and training to ensure consistent and pervasive success in Agile IT acquisition.

References

Agile Manifesto. (n.d.). Principles behind the Agile Manifesto. Retrieved from http://agilemanifesto.org/principles.html

Balter, B. J. (2011, Fall). Towards a more Agile government: The case for rebooting federal IT procurement. *The Public Contract Law Journal.*

Broadus, W. (2013, January–February). The challenges of being Agile in DoD. Defense *AT&L Magazine*, 5–9.

Chairman of the Joint Chiefs of Staff (CJCS). (2012, January 19). *Manual for the operation of the Joint Capabilities Integration and Development System.* Retrieved from http://jitc.fhu.disa.mil/jitc_dri/pdfs/jcids_manual_19jan12.pdf

Defense Acquisition University. (2015, March 20). Milestone Document Identification (MDID). Retrieved from https://dap.dau.mil/mdid/Pages/Default.aspx?ms=2&acat=2&acatsub=1&source=All&type=chart0

Defense Information Systems Agency. (2015, March 20). GCSS-J. Retrieved from http://www.disa.mil/Mission-Support/Command-and-Control/GCSS-J/About

Defense Science Board. (2009). *Report of the Defense Science Board Task Force on Department of Defense policies and procedures for the acquisition of information technology*. Retrieved from Office of the Under Secretary of Defense for Acquisition and Technology website: http://www.acq.osd.mil/dsb/reports/ADA498375.pdf

Duquette, J., & Bloom, M. (2008). *Transitioning Agile/Rapid acquisition initiatives to the warfighter.* The MITRE Corporation.

GAO. (2012). *Software development: Effective practices and federal challenges in applying Agile methods* (GAO-12-681). Washington, DC: GAO.

Lapham, M. A. (2012, January/February). *DoD Agile adoption*. Retrieved from CrossTalk website: http://www.crosstalkonline.org/storage/issue-archives/2012/201201/201201-Lapham.pdf

Lapham, M., Miller, S., Adams, L., Brown, N., Hackemarck, B., Hammons, C., … Schenker, A. (2011). Agile methods: *Selected DoD management and acquisition concerns.* Software Engineering Institute.

Lapham, M. A., Williams, R., Hammons, C., Burton, D., & Schenker, A. (2010). *Considerations for using Agile in DoD acquisition*. Software Engineering Institute.

Modigliani, P., & Chang, S. (2014, March). *Defense Agile acquisition guide*. The MITRE Corporation.

Northern, C., Mayfield, K., Benito, R., & Casagni, M. (2010). *Handbook for implementing Agile in Department of Defense information technology acquisition.* McLean, VA: The MITRE Corporation.

Office of the Secretary of Defense. (2010, November*). A new approach for delivering information technology capabilities to the Department of Defense.* Retrieved from Office of the Deputy Chief Management Officer website: http://dcmo.defense.gov/documents/OSD%2013744-10%20-%20804%20Report%20to%20Congress%20.pdf

ACQUISITION RESEARCH PROGRAM
GRADUATE SCHOOL OF BUSINESS & PUBLIC POLICY
NAVAL POSTGRADUATE SCHOOL
555 DYER ROAD, INGERSOLL HALL
MONTEREY, CA 93943

www.acquisitionresearch.net