

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

The public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing the burden, to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.
PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.

1. REPORT DATE (DD-MM-YYYY) 04/19/2017	2. REPORT TYPE Master's Thesis	3. DATES COVERED (From - To) SEP16 - APR 17
--	--	---

4. TITLE AND SUBTITLE REFORMING DOD INFORMATION TECHNOLOGY ACQUISITION: CONSIDERATIONS FOR AGILE IMPLEMENTATION STRATEGIES	5a. CONTRACT NUMBER N/A
	5b. GRANT NUMBER N/A
	5c. PROGRAM ELEMENT NUMBER N/A

6. AUTHOR(S) Ryburn, Sean H., Major, USMC	5d. PROJECT NUMBER N/A
	5e. TASK NUMBER N/A
	5f. WORK UNIT NUMBER N/A

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) USMC Command and Staff College Marine Corps University 2076 South Street Quantico, VA 22134-5068	8. PERFORMING ORGANIZATION REPORT NUMBER N/A
--	--

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S) N/A

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release, distribution unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT
Over the past 10-15 years, agile software development practices emerged as the industry best practice for software development and information technology governance. Recent DoD IT acquisition reform efforts created opportunities to adopt agile principles and methods to reduce costs, control project schedule, and incrementally deliver effective software while remaining responsive to changing end-user requirements. This research provides a review of IT acquisition methods in the DoD, an overview of the principles and benefits of agile methods, and provides recommendations for executive leadership interested in implementation strategies to incorporate agile software development on defense IT programs.

15. SUBJECT TERMS
Defense Acquisition, DoD Information Technology Acquisition; Agile; Agile Software Development; Agile Principles and Methods; Defense Business Systems

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON	
a. REPORT	b. ABSTRACT	c. THIS PAGE			USMC Command and Staff College	
Unclass	Unclass	Unclass	UU	53	19b. TELEPHONE NUMBER (Include area code) (703) 784-3330 (Admin Office)	

*United States Marine Corps
Command and Staff College
Marine Corps University
2076 South Street
Marine Corps Combat Development Command
Quantico, Virginia 22134-5068*

MASTER OF MILITARY STUDIES

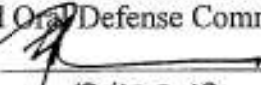
TITLE:
**Reforming DoD Information Technology Acquisition:
Considerations for Agile Implementation Strategies**

SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF MILITARY STUDIES

AUTHOR:
Maj Sean H. Ryburn, USMC


AY 16-17

Mentor and Oral Defense Committee Member: Dr. Benjamin Jensen

Approved: 

Date: 19 APR 17

Oral Defense Committee Member: Dr. Paul Gelpi

Approved: 

Date: 17 MAR 17

DISCLAIMER

THE OPINIONS AND CONCLUSIONS EXPRESSED HEREIN ARE THOSE OF THE INDIVIDUAL STUDENT AUTHOR AND DO NOT NECESSARILY REPRESENT THE VIEWS OF EITHER THE MARINE CORPS COMMAND AND STAFF COLLEGE OR ANY OTHER GOVERNMENTAL AGENCY. REFERENCES TO THIS STUDY SHOULD INCLUDE THE FOREGOING STATEMENT.

QUOTATION FROM, ABSTRACTION FROM, OR REPRODUCTION OF ALL OR ANY PART OF THIS DOCUMENT IS PERMITTED PROVIDED PROPER ACKNOWLEDGEMENT IS MADE.

Table of Contents

	Page
DISCLAIMER.....	ii
LIST OF FIGURES.....	iv
LIST OF TABLES.....	v
EXECUTIVE SUMMARY.....	vi
INTRODUCTION.....	1
Introduction.....	1
Purpose of the Research.....	3
Problem Statement.....	3
Significance of the Research.....	6
Research Scope and Limitations.....	6
LITERATURE REVIEW.....	9
DoD IT Acquisition: Brief History and Policy.....	10
Traditional Software Development Models in the DoD.....	12
DoD Acquisition Reform: Agile Software Development Practices in the DoD.....	19
Agile Software Development: Principles and Methods.....	23
Types of Agile Methods.....	29
Organizational Benefits of Adopting Agile Methods.....	30
CRITICAL CONSIDERATIONS FOR AGILE IMPLEMENTATION STRATEGIES IN THE DOD.....	35
Preparing to Implement Agile Principles and Methods.....	35
Team Environment and Organizational Culture.....	39
Agile Contracting and Oversight.....	48
CONCLUSION.....	52
BIBLIOGRAPHY.....	56

List of Figures

	Page
Figure 1. Waterfall Methodology.....	13
Figure 2. Spiral Software Development Model.....	17
Figure 3. New Acquisition Process for Information Technology.....	21
Figure 4. Comparison of Agile and Waterfall Development.....	26
Figure 5. Dimensions Affecting Method Selection.....	37
Figure 6. Hierarchical Organizations.....	45
Figure 7. Cooperative Organizations.....	46

List of Tables

Page

Table 1. Comparison of Traditional Principles to Agile Principles.....	29
--	----

Executive Summary

Title: REFORMING DOD INFORMATION TECHNOLOGY ACQUISITION: CONSIDERATIONS FOR AGILE IMPLEMENTATION STRATEGIES

Author: Maj Sean H. Ryburn

Thesis: Implementing agile software development methods in DoD information technology (IT) acquisition programs extends beyond the technical engineering discipline; success requires a collaborative effort across a program management office and between all involved stakeholders to transition teams to an agile framework and align organizational culture with agile principles. To be effective, the DoD must adopt agile IT governance practices, not just agile software development methods. Program managers implementing agile software development strategies must focus on three critical areas to ensure success: deciding and preparing to transition to agile principles and methods; developing an effective team environment and organizational culture; and preparing an adequate contracting strategy that supports the iterative nature of agile software development.

Discussion: Over the past 10-15 years, agile software development practices and project management methods emerged as industry best practices to address inherent shortfalls in traditional software development methods that resulted in the over-budget and late delivery of poorly functioning software that failed to meet end-user requirements. Agile software development is a collective term used to describe a set of values and principles focused on creating effective teams within an organization that are able to respond to change and succeed in an uncertain environment. Agile principles and methods describe adaptive management approaches used to ensure responsiveness to end-user needs, incrementally deliver functional software, and improve program management and IT governance. Agile software development practices are effectively employed throughout industry—from small entrepreneurial environments to larger more established firms—and have demonstrated quantitative results to control costs, prevent schedule delays, improve software performance, and address customer satisfaction. The DoD has consistently struggled with the timely, cost-effective, and efficient acquisition of information technology as a result of an acquisition process that is based on weapon systems procurement and tends to proscribe a traditional approach to software development. Recent DoD acquisition reform efforts have created opportunities to improve the efficiency and effectiveness of DoD IT acquisition and encourages use of industry best practices such as agile software development. As a result, DoD IT programs have begun to incorporate agile principles and methods, but the new DoD guidance lacks specific instruction for developing effective agile implementation strategies.

Conclusion: Agile implementation is recommended for DoD IT acquisition, particularly for defense business systems. To be effective, executive leadership must tailor agile implementation strategies to incorporate effective upfront strategic planning, develop a collaborative environment and organizational culture aligned with agile principles, and prepare adequate contracting strategies that support the dynamics and flexibility of agile software development.

Introduction

The Department of Defense (DoD) must improve efforts and strategies to implement agile software development methods in information technology (IT) acquisition and program lifecycle management. Agile software development and project management methods are widely accepted industry *best practices* that have been adopted to address inherent shortfalls in software development projects that traditionally resulted in the over-budget and late delivery of poorly functioning software that fails to meet end-user requirements. As software became more complex, traditional design and development techniques such as the “waterfall” software development life-cycle (SDLC) model failed to adequately meet organizational needs. As a result, the information technology (IT) industry struggled to effectively develop and deploy usable systems and software within acceptable cost and schedule constraints. In response, agile methodologies were developed to improve software development and project management processes.

Agile methods achieved initial success in small to medium-sized high-technology and startup companies, however over the past decade larger organizations have successfully incorporated agile practices into their businesses to improve IT project performance. Today, agile methods are successfully implemented by thousands of organizations worldwide as *best practice* for software development projects to control costs, prevent schedule delays, improve software performance, and address customer satisfaction. Prominent organizations that use agile methodologies for software development and project management include Nokia Siemens Networks, Yahoo!, Google, Microsoft, BT, Bankwest and Suncorp, for example.¹ The proven benefits of implementing agile methods resulted in adoption throughout private industry—

including high technology, finance, and telecommunications—and includes many larger more established firms rather than limited to the entrepreneurial environment of smaller high-technology start-ups. Given the successful results from agile implementation in large organizations in the private sector, the DoD has begun attempts to adopt agile methods in the DoD IT acquisition process to address similar longstanding problems that have plagued the DoD's IT acquisition efforts.

The Department of Defense has historically struggled with similar challenges experienced by the private sector in the cost-effective and efficient acquisition of IT capabilities that effectively meet end-user requirements. As a result, efforts to improve DoD IT acquisition attempt to leverage similar agile software development techniques to improve the acquisition and lifecycle management process. Initiatives to implement agile methodologies in DoD acquisition, however, have not yet uniformly demonstrated drastic improvements and may not necessarily be effective for all defense IT system acquisition scenarios. Furthermore, challenges in implementing agile software development principles and methods in the federal government often result from project management activities that neither adequately transition program management office teams to an agile framework nor properly align organizational culture to accept agile principles, rather than technical engineering concerns.² The intent of this research is to outline the transition from traditional SDLC models to agile methods in the DoD and develop a conceptual framework—based on observations and industry *best practices*—for program managers to improve agile software development implementation strategies within enterprise DoD defense business systems.

Purpose of the Research

The purpose of this research is to review existing business and academic literature and available case studies on the effective implementation of agile methodologies to develop a framework to improve the chances for successful agile implementation for the acquisition of defense business systems. Agile has emerged as the leading industry software development methodology and has been successfully implemented by a variety of businesses. As agile software development methods designed to address the complexity in software development, improve performance, and control costs matured in the private sector, the DoD implemented reforms in the IT acquisition process intended to achieve similar results. These reforms implemented an incremental approach to IT acquisition that encourages adoption of agile methods in system design and software development. Many of the larger defense business systems that support enterprise-level business activities such as acquisition, financial management, logistics, strategic planning and budgeting, installations, and human resource management share similar characteristics to those enterprise business systems deployed in the private sector that have benefited from agile software development methods. By understanding and leveraging those lessons learned by industry when employing agile methodologies for the acquisition of enterprise IT systems, DoD program managers and acquisition professionals can improve the likelihood of successfully implementing agile methods on DoD programs.

Problem Statement

For the DoD, information technology has become increasingly integrated in weapon systems and is a critical enabler for the business management systems that enable the effective operation of the defense enterprise.³ Both hardware and software technology continues to

advance rapidly as predicted by Gordon Moore in 1965 when he observed that the number of transistors on an integrated circuit doubles every two years and projected that this growth correlates to processing capacity.⁴ Known as “Moore’s Law,” this projection of rapid technological advancement and growth has been remarkably accurate and contributes to the increased complexity in software development and program management. To address this complexity, firms in the private sector have successfully implemented various agile software development methods while the DoD continues to use a deliberate acquisition process that can take 7-10 years from planning to delivery.⁵

The DoD fiscal year 2017 (FY17) budget request is \$38.2 billion for IT investments, which includes priorities in modernizing DoD networks, improving cybersecurity and building the IT workforce.⁶ Despite the enormous costs, these IT investments frequently result in multi-million-dollar cost-overruns and excessive schedule delays that deliver solutions that are too late or inadequate to meet the requirements of the user community. Furthermore, by the time defense systems are fielded, the software is often approaching obsolescence. This is a particular concern for networked enterprise defense business systems that leverage commercial off-the-shelf products because the software must remain up-to-date to ensure adequate vendor support and compliance with IT security policy. Cost-overruns, schedule-delays, and the poor performance of defense business systems are the result of the cumbersome DoD acquisition process that proscribes a linear approach to systems development and product life-cycle management. This linear approach was developed to manage the full lifecycle acquisition of hardware intensive weapon systems, which resulted in traditional “waterfall” software development methods where defined capabilities are developed and integrated into the hardware in a sequential order. For software intensive systems, particularly defense business systems, the traditional “waterfall”

method for software development consistently fails to result cost-effective solutions. Despite several attempts at reform, the acquisition of IT systems in the DoD continues to be problematic and is particularly slow with the acquisition of large defense business systems, the largest of which are categorized by the DoD as Major Automated Information Systems (MAIS).⁷

Efforts to reform the DoD IT acquisition process resulted in new acquisition guidance, outlined in DoD Instruction (DoDI) 5000.02, that emphasizes incremental delivery and encourages the use of agile methods for software development.⁸ As a result, DoD IT programs have begun to incorporate aspects of agile software development to improve program performance. However, the new guidance lacks specific instruction for developing an effective strategy for implementing agile software development methods on DoD IT programs. Implementing agile software development methods in DoD IT programs extends beyond the technical engineering discipline; success requires a collaborative effort across a program management office and between all involved stakeholders to transition teams to an agile framework and align organizational culture with agile principles. To be effective, the DoD must adopt agile governance practices, not just agile software development practices. To ensure success, program managers must understand the fundamental principles of agile software development, how agile software development methods are incorporated in the DoD acquisition process, and must take proactive steps to cultivate an environment that supports agile principles and methods. Program managers implementing agile software development strategies must focus on three critical areas to ensure success: deciding and preparing to transition to agile principles and methods; developing an effective team environment and organizational culture; and preparing an adequate contracting strategy that supports the iterative nature of agile software development.

Significance of the Research

The DoD has implemented an incremental approach for IT acquisition, and encourages projects to take advantage of the benefits of agile development methods, especially in defense business system programs that use state-of-the-practice commercial technologies.⁹ However, agile methods have not been entirely embraced throughout the acquisition workforce, nor have practitioners received sufficient information and guidance on agile to transition from traditional development models to an agile approach. Agile software development is a relatively new concept for the DoD, and many within the acquisition workforce lack sufficient understanding, training, and experience to develop an effective strategy for implementing agile that addresses common implementation challenges. This research provides a comprehensive review of the traditional software development models, outlines the principles and benefits of agile principles and methods, and provides specific guidance in the critical areas of strategic planning, organizational culture, and contracting to assist program managers in developing agile implementation strategies on DoD programs. Traditional software development models failed to adequately deliver effective software; to adopt agile methods in DoD IT acquisition, program managers must develop effective agile *implementation strategies* to take advantage of the benefits of agile software development.

Research Scope and Limitations

This research focuses on a comprehensive review of existing literature that includes business literature and case studies from private sector agile implementation, as well as available literature recently published to provide guidance to the federal and defense acquisition workforce to leverage agile best practices to reduce costs and improve performance of defense business

system programs. DoD IT programs consistently fail to deliver cost-effective solutions. To address inefficiencies in the processes for DoD IT acquisition that are created by the required compliance with federal acquisition regulations, the 2010 National Defense Authorization Act (NDAA) provided guidance to implement new processes to streamline and expedite IT acquisition. Subsequently, in 2010 the Office of the Secretary of Defense (OSD) released a report titled “A New Approach for Delivering Information Technology Capabilities in the DoD” that recommends implementation of agile software development principles and methods based on industry best-practices to reduce program life-cycle costs and deliver increments of capability to the end-users. Proper implementation and adherence to these reforms are critical for the timely and cost-effective acquisition of IT, particularly in regard to the procurement of larger defense business systems such as those classified Major Automated Information Systems (MAIS).

Agile software development is not specifically a technical engineering process; to be effective, program managers must proactively implement agile principles within program management office (PMO) organizational culture. Program managers (PM) must be engaged throughout the process and become familiar with agile methods and best practices. The PM must adequately train and prepare *all* members of the PMO to implement agile, and understand how agile principles are used to adapt and respond to the inevitable change in user requirements while adequately preparing government contracts to support contractual ambiguity often created with agile methods. This research is focused on providing a comprehensive review of DoD IT acquisition reform efforts leading to adoption of agile methods, as well as government and industry best practices for agile strategic planning, changing organizational culture, and contracting for agile on a defense business system. This research is limited to observations and an applied literature review; due to time and scope constraints, no empirical data collection was

conducted to support this research. A significant body of literature exists to support agile software development implementation in the private sector, the DoD must leverage private sector knowledge and experiences to effectively implement agile, particularly on defense business system that share similarities with those in the private sector.

Literature Review

Agile software development methods were introduced in the private sector to address reoccurring problems in software development and service delivery within the IT industry, including cost-overruns, schedule delays, poor performance and dissatisfied end-users. Over the past 15 years, agile has become recognized as an IT industry *best practice* for software development and program management methods. Agile describes software development and engineering techniques used for effective software development, but more importantly, agile includes the principles and methods recommended for implementing an adaptive management approach that ensures responsiveness to end-user needs, incremental delivery of useable software, and time-boxing techniques for effective program management of software intensive programs.

A large body of research and literature is currently available to help guide individuals seeking to incorporate agile methods into existing software development and project management processes within their organization. The major themes of this literature are predominately focused on best practices for agile implementation in the private sector. Since 2010, the DoD recognized the importance of incremental software development and recommends incorporating industry best practices on DoD programs such as agile software development methods. Given the increasing recognition placed on agile software development by the DoD, several reports and various literature focuses on recommendations for agile implementation on federal IT projects and within the DoD acquisition community. Notably, the Carnegie Mellon Software Engineering Institute (SEI) has contributed to this body of research and in 2016 published an update to a 2010 report “Considerations for Using Agile In DoD Acquisition” that provides guidance for DoD agile implementation strategies.¹⁰ As agile becomes more popular

within the defense industry, a growing body of literature and research is becoming available that analyzes best practice strategies for implementing agile software development on defense IT programs. This research contributes to that growing body of literature by reviewing implementation strategies, specifically applied to defense business systems, while focusing on critical areas that executive leadership and program managers must influence to facilitate successful agile implementation strategies, rather than a focus on the technical engineering aspects of agile software development.

This literature review provides a comprehensive background of the problems with DoD IT acquisition and the history of IT acquisition reform, how traditional software development models (i.e. “waterfall” and “spiral” development) failed to adequately achieve efficient software in the DoD, and provides a review of recent DoD reform efforts to incorporate agile in DoD acquisition processes. The importance of this literature review is to give the reader a comprehensive review of the history and trends of DoD IT acquisition to provide an understanding of the recent efforts focused on incorporating agile into DoD IT programs.

DoD IT Acquisition: Brief History and Policy

For many years, senior leaders within the Department of Defense, Congress, and industry have struggled with the DoD’s inability to deliver timely and effective information technology (IT) solutions within established budget estimates. Systemic causal problems within the DoD acquisition process have been identified in numerous studies and reports; common conclusions include:

- The DoD acquisition process, outlined by DoD Directive (DoDD) 5000.01, is based on weapon systems and are too slow and costly;

- Program requirements and available technology change faster than capabilities can be delivered to the warfighter;
- DoD IT projects are rarely delivered on time and within budget, while many are cancelled before delivery.¹¹

The Defense Acquisition System is governed by DoD Directive (DoDD) 5000.01 “The Defense Acquisition System,” and is managed through comprehensive guidance disseminated in the Defense of Defense Instruction (DoDI) 5000.01 “Operation of the Defense Acquisition System.” These documents provide the foundational guidance for the acquisition of weapon systems—and information technology—for the DoD. These documents, known collectively as the “DoD 5000 series regulations,” prescribe a linear approach to systems development and product life-cycle management and are intended to provide oversight and guidance on defense acquisition activities.

After consolidating IT acquisition management under the DoD 5000 series regulations in 1996 under the Information Technology Management Reform Act (ITMRA), the DoD struggled to provide affordable and effective military capabilities, particularly with respect to larger weapon system programs. Notable weaknesses in the DoD 5000 series directives included the inadequate assessment of technological maturity prior to system design and development, insufficient government review and oversight throughout the development phases, and inadequate execution of operational testing. Consequently, various attempts were made to reform the acquisition process, which introduced additional process steps and reviews in the defense acquisition system.¹² Many argued, however, that the introduction of these reforms had an adverse impact on IT programs that are regulated by the same DoD 5000 directives.

Consequently, these reforms further inhibited IT acquisition within the DoD, particularly with

the overemphasis on documentation and acquisition oversight for Commercial Off The Shelf (COTS) solutions that are often used in defense business systems programs. In 2009, the National Research Council found that “IT program managers...indicated that DoD 5000 processes dramatically increase the time to deliver solutions, especially those available as commercial off-the-shelf (COTS) solutions.”¹³ In effect, the burden posed by additional oversight mandated by the DoD 5000 series reforms reinforced inefficient processes for IT acquisition by institutionalizing formalized oversight and review (i.e. milestone reviews) and increased documentation requirements that subsequently necessitate larger associated support teams and resources.

Traditional Software Development Models in the DoD

Traditional software development models, used in both the DoD and the private sector, were an accepted industry *best practice* to establish effective management over software development activities in an effort to address the increasing complexity of software. The longstanding acquisition guidance outlined in the DoD 5000 series directives proscribes a linear, sequential approach to systems development and product life-cycle management. This linear approach to program acquisition, which is the acquisition model used frequently for major hardware systems, correlates to the traditional “waterfall” software development lifecycle model (SDLC) in which “well-defined increments of capability or technology are designed, developed, and fielded in a pre-specified order.”¹⁴

Traditional waterfall software engineering methodology was one of the first attempts to formalize software project management and originated from the work of Winston Royce in 1970.¹⁵ The term *waterfall* was used to describe a sequential software development process in

which a software development project flowed through a series of phases: system requirements, software requirements, analysis, program design, coding, testing, operations.¹⁶ The sequential process outlined in the waterfall methodology provided both developers and management a formalized series of sequential stages, beginning with requirements identification and moving to design, implementation, testing, and deployment (see Figure 1).¹⁷ As increasing computing power significantly increased the complexity of software design, substantial costs were incurred for the “rework” of failed software; the benefit for implementing the waterfall methodology provided a rigid systemic approach that enforced extensive documentation throughout the SDLC that allowed the effective management of large software systems.

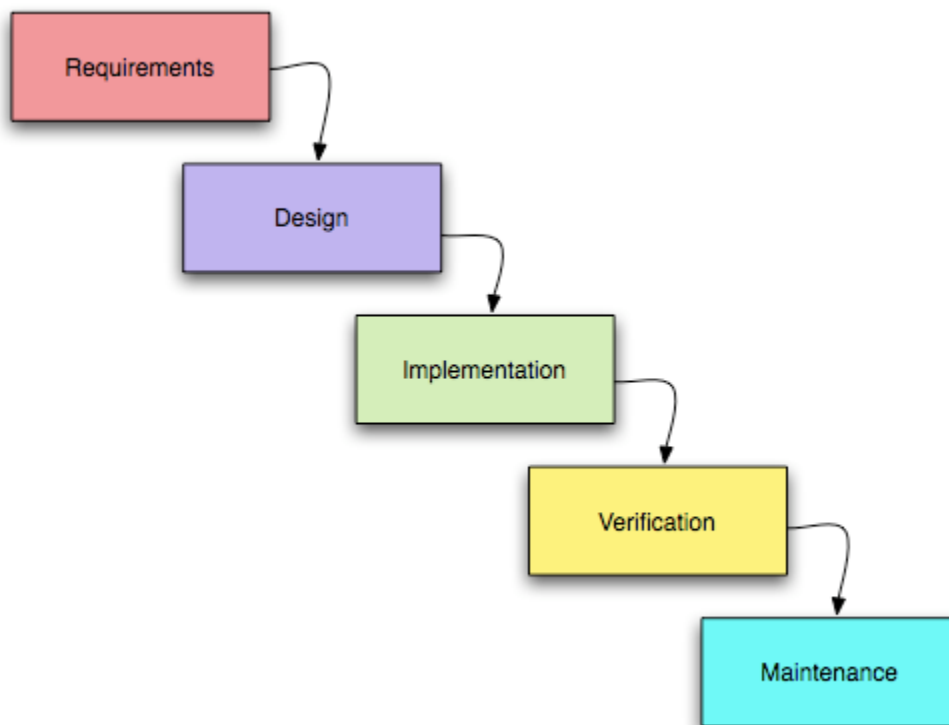


Figure 1: Waterfall Methodology (Source: Hughey, 2009)

Although Royce had originally argued for an iterative approach between the sequential waterfall steps in his original published article, the DoD adopted the linear and sequential

waterfall approach and incorporated the process into military standards for software development. The DoD adopted DoD Standard (STD) 1679A for software development, although its origins were in hardware and weapon systems development and reflects the predominance of waterfall SDLC in DoD software development and management.¹⁸

Despite early successes, the waterfall SDLC model failed to address persistent problems with DoD software development. The heavy reliance on project documentation and an inability to effectively define and understand system requirements upfront continued to result in software development efforts that were over-budget, behind schedule, and failed to deliver needed capability to the warfighter. The traditional waterfall model assumes that all system requirements can be identified and documented at the beginning of a project when, in reality, “IT systems can be fraught with uncertainty and ambiguity making it difficult to accurately define the end state upfront.”¹⁹ Because software is intangible or abstract, as compared to hardware or weapon system procurement, users are often unable to adequately define system requirements upfront. Additionally, the level of technical complexity in software design and coding, as well as interrelationships in software code and external interfaces, often result in unforeseen issues that are difficult to predict at the outset of the project.²⁰

Additional criticism of the traditional waterfall SDLC model is the inflexibility of the sequential methodology characterized by up-front planning, excessive formal documentation and functional development teams that develop the product in a linear approach. The waterfall approach tends to result in a handoff from one functional team to another at the end of each defined phase, which hinders creativity and delays decisions on the project to the end of a development phase, causing inflexibility and reducing the ability to adapt changes as they become apparently necessary.²¹ This approach to IT governance inhibits the projects ability to

course correct and, given the large number of stakeholders whose equities must be satisfied at each decision point at the conclusion of a development phase, often results in extended timelines and delayed testing efforts that create cost-overruns and schedule delays in the project.

Throughout the 1990's, the private sector experienced similar issues with traditional software development that resulted in a significant failure rate for software development projects. The use of waterfall approaches in industry was intended to reduce business risk in project delivery by ensuring each sequential step in the development process was completed to management's satisfaction before additional expenditures were incurred. However, IT industry leaders found that waterfall approaches tended to actually *increase* business risk of IT project failure by "mandating big upfront documentation, discouraging responsiveness to changing requirements as the project evolved, (and) creating "silos" of ownership that reduced communication across project team members."²² Furthermore, industry leaders soon identified that the most significant impact to project risk was that waterfall development approaches delay the delivery of tangible business outcomes until the end of the project when the software is delivered to the end-user, which is the point when necessary modifications to the software, due to changing business requirements, are the most costly.²³

In addressing the challenges in IT system acquisition in the DoD, several studies criticized the waterfall SDLC model specifically, including a 1987 Defense Science Board study that emphasized the need for DoD software acquisition regulations and management methodologies should align with industry best-practices that practice iterative software development and prototyping rather than the traditional waterfall SDLC model:

DoD Directive 5000.29 and STD 2167 codify the best 1975 thinking about software, including a so-called "waterfall" model calling for formal specification, then request for bids, then contracting, delivery, installation, and maintenance. In the decade since the waterfall model was developed, our discipline has come to recognize that setting the requirements is the most difficult and crucial part of the software building process, and one that requires iteration between the designers and users. In best modern practice, the early specification is embodied in a prototype, which the intended users can themselves drive in order to see the consequences of their imaginings. Then, as the design effort begins to yield data on the cost and schedule consequences of particular specifications, the designers and the users revise the specifications.²⁴

These criticisms led to attempts by the DoD to adopt an iterative, incremental development (IID) methodology, articulated in the "spiral model" for software development originally outlined by Barry Boehm in 1985.²⁵ The spiral model attempted to address the inefficiencies of the traditional waterfall model by incorporating an iterative approach to software development with frequent end-user involvement (See Figure 2). The attraction of the spiral approach was that the methodology was the first SDLC method that "explicitly took a risk-driven approach to software development, in contrast to the document driven, and the code driven approaches" of the traditional waterfall method.²⁶ The DoD attempted to incorporate the iterative spiral development model in defense IT acquisition processes, and in 2000 the DoD Instruction (DoDI) 5000.02 was revised (with further clarification in a 2002 USD AT&L memorandum) to identify evolutionary acquisition and spiral software development methods as the *preferred* approach for IT acquisition based on multiple delivery increments and multiple cycles within each delivered increment.²⁷

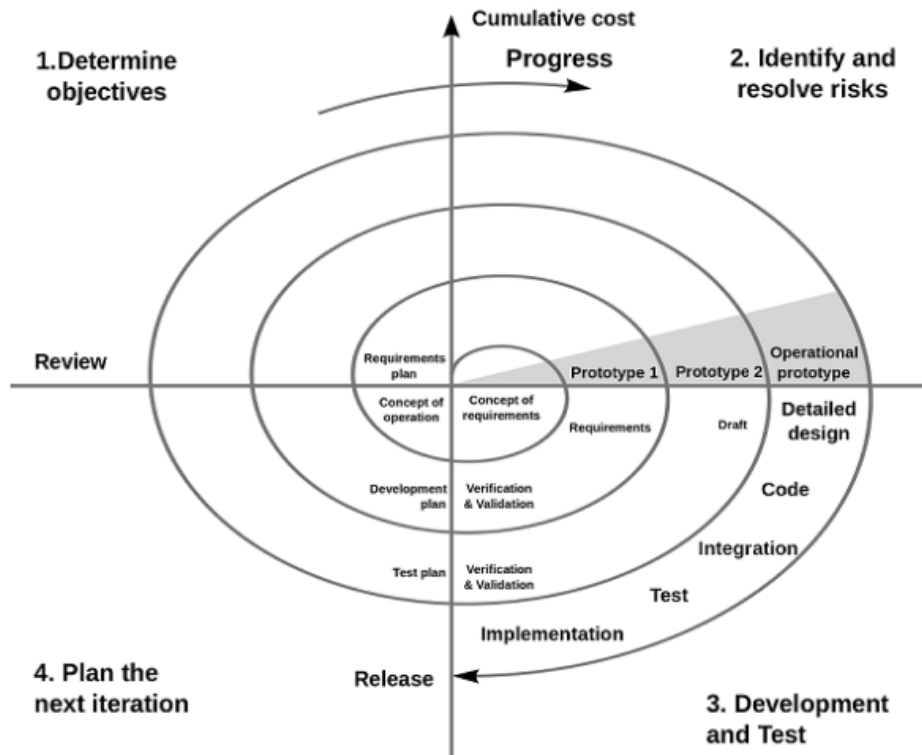


Figure 2: Spiral Software Development Model (Source: <http://www.seowebdesign.com/the-spiral-model-of-software-development/>)

Despite efforts to reform DoD IT acquisition processes, in practice the iterative “spiral development” method failed to receive demonstrable results in improving cost, schedule and performance metrics for DoD IT acquisition. Software practitioners found the methods heavily process based and with a heavy emphasis on documentation requirements. The ability to learn and quickly apply the spiral methodology was perceived by many as a limitation, as well as being labor intensive and time consuming, thus extending and slowing the development process.²⁸ Other significant limitations included an inability to adapt to the changing circumstance in organizations and projects or to be able to adequately adapt to changing customer requirements.²⁹ Furthermore, because IT acquisition remained guided by the DoD 5000 series regulations that are dominated by a hardware and weapons systems mentality, each

phase in the evolutionary acquisition governance model repeats critical decision milestones and engineering design reviews (i.e. Preliminary Design Review [PDR] and Critical Design Review [CDR]), hallmarks of the traditional waterfall SDLC model. Given the number of Milestone Decision Authority (MDA) reviews and decision points in each evolutionary cycle, the *preference* for an evolutionary acquisition approach for large IT acquisition programs failed to translate into actual implementation. Indeed, the 2009 National Research Council study observed that “the iterative incremental software approach development model...requires heroics on the part of the program manager (PM) and MDA to apply iterative, incremental development (IID) successfully within the DoDI 5000 framework.”³⁰

The failure to adopt an iterative, incremental development approach and successfully implement the spiral development methodology in DoD IT acquisition during the first decade of the twenty-first century resulted in the cultural persistence of traditional waterfall SDLC mentality in the acquisition workforce. This mentality has been reinforced by the reliance on a hardware and weapon-system centric governance and oversight model mandated by the DoD 5000 series regulations. As described by the 2010 National Research Council study, the traditional waterfall SDLC model tacitly remains as the oversight structure and process that governs DoD IT acquisition:

In the DoD, a significant structural factor leading to this failure to meet end-user expectations is the persistent influence over many decades of what is characterized as the waterfall software development life cycle (SDLC) model—despite a body of work that is critical of the waterfall mentality...and the issuance of directives identifying models other than the waterfall approach as the preferred approach. *The waterfall*

*model...remains at least implicit in the oversight structure and processes that govern IT acquisition programs in the DoD today (emphasis added).*³¹

DoD Acquisition Reform: Agile Software Development Practices in the DoD

Military and political leaders consistently recognize that the process by which the federal Government, and in particular the DoD, acquires IT solutions contributes to cost-overruns, schedule delays, and inadequate performance and must change. Concern over IT procurement effectiveness does exist throughout the federal Government; A 2008 Government Accountability Office (GAO) study, for instance, found that a significant number of federal IT projects (totaling \$25.2 billion in expenditures in FY2008) were “poorly planned, poorly performing or both,” and that 48 percent of the federal government’s major IT projects have been rebaselined at least twice.³² For the DoD, several studies have reached the similar conclusion that managing IT acquisition through the DoD 5000 series regulations contributes to the DoD’s inability to deliver timely and effective IT solutions.³³ The most notable of the studies in the longstanding discussion about how to fix DoD’s IT acquisition problems was the 2009 comprehensive report conducted by the Defense Science Board (DSB) that warned: “The deliberate process through which weapon systems and information technology are acquired by DOD cannot keep pace with the speed at which new capabilities are being introduced in today’s information age—and the speed with which potential adversaries can procure, adapt, and employ those same capabilities against the United States.”³⁴

The significance of the 2009 DSB report was the conclusion that the conventional DoD acquisition process—managed through the DoD 5000 series regulations—is “too long and cumbersome to fit the needs of the many IT systems that require continuous changes and

upgrades.”³⁵ The DSB task force concluded that it would not be effective to tailor the DoD 5000 series regulations and existing acquisition processes to meet the special characteristics for IT capability procurement, and recommended that the DoD implement a unique acquisition process designed specifically to meet the special characteristics of information technology (See Figure 3). The unique, tailored DoD IT acquisition process recommended by the DSB task force is modeled on successful commercial best-practices, designed for the rapid acquisition of IT capabilities, and supports the continuous upgrades and improvements necessary to sustain IT capabilities. The task force recommended the delivery of meaningful increments of system capability in approximately 18 months or less, which allows prioritization of requirements based on end-user needs while in balance with technical readiness and maturation.³⁶ Additionally, the incremental approach for capability release potentially permits the early operational release of functional capability, rather than excessively delaying capability release until program milestone decisions (i.e. Initial Operational Capability [IOC]), which was typical for IT programs managed under DoD 5000 series acquisition regulation. Furthermore, the recommended unique DoD IT acquisition process permits and encourages the ability to accommodate changes driven by end-user experiences and clarified requirements, particularly through the “active engagement of the user community throughout the acquisition process.”³⁷

The DSB task force recognized that in contemporary IT projects, changing circumstances often cause requirements to evolve from the original descriptions outlined in pre-milestone “A” system requirements documentation to support the changing needs of an evolving environment. This reality is frequently the case for defense business systems because a broad community of end-users often refine and clarify system requirements during the development and testing stages of the program. The DSB task force recommended a *new* process that allows “desired

capabilities” to be traded-off against program costs and initial operational capabilities (IOC) to deliver the most effective and usable capability to the end-user in the timeliest manner. The DSB task force further recommended the implementation of successive prototyping to “support an evolutionary approach,” and highlighted the importance of early and continuous involvement of key stakeholders throughout the design, development and testing of the IT capability and prior to Milestone Build Decision (equivalent to the MDA approval of the Capability Development Documentation in accordance with DoD 5000 series regulation). Industry best practices, such as the principles and methods of agile software development, influenced the incremental approach to IT acquisition described in the 2009 DSB report. These principles of agile development focus on an iterative development approach and the incremental deployment of usable software.

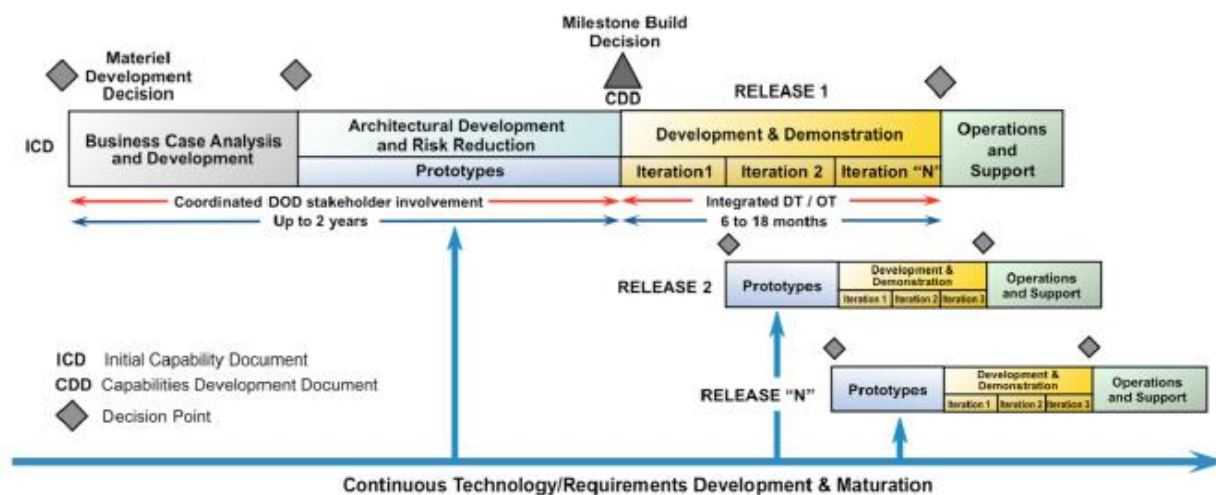


Figure 3: New Acquisition Process for Information Technology (Source: DSB, pg. xi)

The recommendation by the DSB to develop and implement a new acquisition process for information technology systems was required by Congress in Section 804 of the FY 2010 National Defense Authorization Act (NDAA). Specifically, Congress required that the new IT acquisition process be designed to include:

- Early and continual involvement of the user;
- Multiple, rapidly executed increments or releases of capability;
- Early, successive prototyping to support an evolutionary approach; and
- A modular open-systems approach.³⁸

In response to the FY 2010 NDAA, the DoD release a report, “A New Approach for Delivering Information Technology Capabilities in the DoD,” that provided an update and plan for the implementation of a new acquisition process for IT procurement and initially targeted defense business systems for implementation. The report set the stage for the implementation of Agile processes within the DoD and focused on: (1) delivery early and often, (2) incremental and iterative development and testing, (3) rationalized requirements, and (4) flexible/tailored processes.³⁹ Additionally, the DoD Chief Information Officer’s (CIO) modernization plans and the FY 2013 budget priorities outlined by the White House CIO identify the government’s need to establish processes and “agile teams” to support the effective acquisition of IT for the DoD.⁴⁰ With both legislation and internal DoD policy in place, the DoD IT acquisition reforms facilitate and encourage implementation of agile software development methods to more effectively develop and deliver needed IT capabilities to the end-user while reducing program lifecycle costs.

Because agile software development focuses on the iterative release of software while recognizing and accommodating changing requirements, these methods can be particularly effective for defense business systems that often support a broad user community. To ensure agile software development is not the latest “buzzword” in the defense acquisition community and is effectively adopted within DoD IT programs, the acquisition workforce and particularly program managers leading IT acquisition organizations must understand agile principles and

develop effective strategies for implementation. To be effective, program managers must adopt agile *governance practices* within their organization rather than just the technical aspects of agile software development. The following section provides a review of agile software development principles and methods as well as the benefits of adopting agile experienced by organizations in the private sector.

Agile Software Development: Principles and Methods

Agile is a collective term to describe principles and methods that have recently emerged to increase the “relevance, quality, flexibility, and business value of software solutions.”⁴¹ Agile methodologies describes the adaptive management approaches that were developed specifically to address reoccurring problems in software development and service delivery within the IT industry, including cost-overruns, schedule delays, poor performance and dissatisfied end-users—similar problems experienced with traditional DoD IT system acquisition processes.

During the 1990’s, the IT industry experienced a significant failure rate of software development projects due to cost-overruns, missed deadlines, poorly performing deliverables and dissatisfied customers. Leaders within the commercial IT industry identified three key factors that contributed to the systemic failures: over-planning, insufficient communication and overreliance on an “all-at-once” software delivery.⁴² To address these issues, 17 leading software developers met in 2001 to share and develop better processes to develop software. The result of the meeting was a general consensus over industry best-practices for effective software development practices and the creation of the *Agile Manifesto*, which describe the overarching concepts and principles of Agile software development methodology:

We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:

Individuals and interactions over processes and tools

Working software over comprehensive documentation

Customer collaboration over contract negotiation

Responding to change over following a plan.

That is, while there is value in the items on the right, we value the items on the left more.⁴³

The premise behind agile methodology is that the most effective and efficient means to deliver workable software requires *direct feedback* from the end-user's actual *hands-on review* of working capabilities within the IT system, rather than reliance on detailed system requirement specification documentation.⁴⁴ Agile software development describes an iterative method of managing software development and IT projects—based on concepts and principles from the Agile Manifesto—that remains flexible to change and focuses on interaction with the customer. Rather than reliance on traditional upfront project planning and detailed documentation, agile methods implement incremental planning to adapt to the most current information available in an anticipated changing and dynamic environment. Agile methods encourage an iterative development process that focuses on frequent delivery of high quality, functional software to *effectively* meet the needs of the end-user. Agile is based on management processes that improve the delivery of capability as an alternative to the “water fall,” or traditional sequential development methods. These processes focus on small flexible teams that iteratively develop and

improve software throughout the SDLC using customer input and feedback to converge on acceptable solutions. The major factors focus on: (1) early customer involvement, (2) iterative development, (3) self-organizing teams, and (4) adaption to change.⁴⁵

Agile approaches use *time-boxing* techniques that are intended to help development teams during capability development respond to anticipated unpredictability by incorporating empirical end-user feedback to incremental and iterative work cadences referred to as “sprints.” Whereas traditional waterfall methods use a plan-driven approach that identifies all requirements and develops a work breakdown structure upfront, agile methods break up tasks into smaller chunks to be developed incrementally using fixed-length intervals for developing those features. This time-boxing technique recognizes that system requirements are often not known or well-defined upfront, and given the pace with which IT capabilities change and the inherent complexity of software, end-user requirements often evolve throughout the design and development stages of the project. Agile time-boxing advantages include ensuring team members focus on specific work within a defined time-period, increased productivity, and improved awareness and project oversight of the development effort.⁴⁶ Traditional “waterfall” methods managed under the DoD 5000 series regulations reinforce delivery of outdated systems based on requirements that were defined 5-10 years prior during pre-Milestone “A” system design specifications. Agile methods, on the other hand, encourage continuous end-user feedback throughout the development process and manage development work using iterative time-box techniques. This ensures efficient delivery of incremental usable software in small short increments rather than the sequential phases of traditional waterfall methods that deliver a full product at the end of the project (see Figure 4).

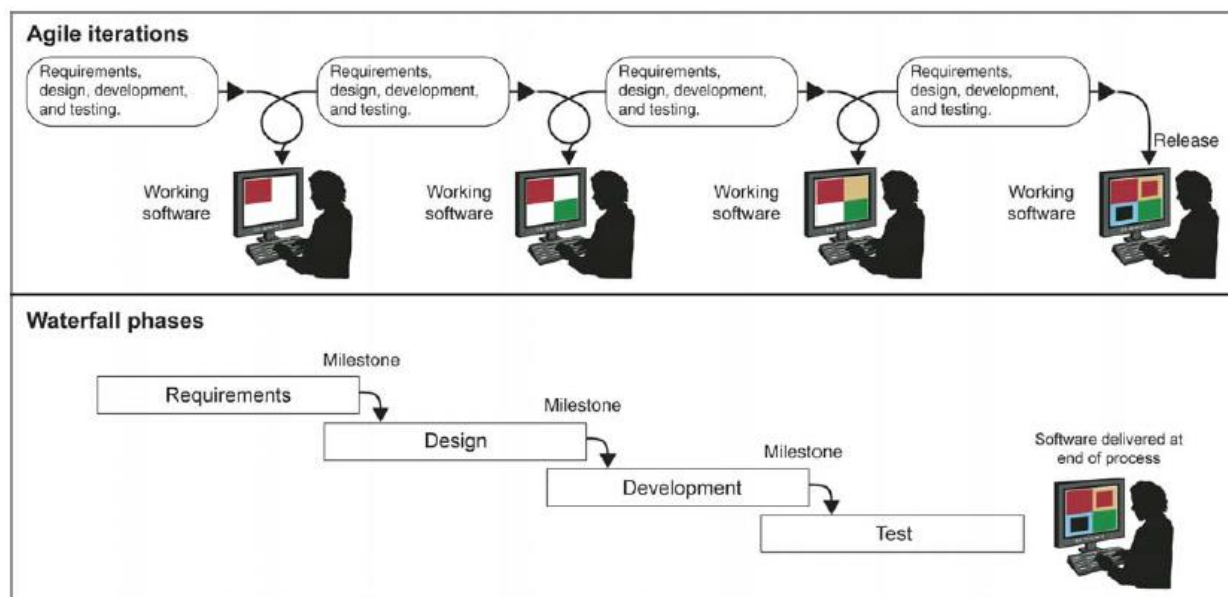


Figure 4: Comparison of Agile and Waterfall Development (Source: GAO (1), 7)

The Agile Alliance, which is the guiding body for the *Agile Manifesto* and continues to support the organizational implementation of agile methodologies, provides the following 12 principles that underlie the Agile Manifesto:

Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.

Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.

Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.

Business people and developers must work together daily throughout the project.

Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.

The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.

Working software is the primary measure of progress.

Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.

Continuous attention to technical excellence and good design enhances agility.

Simplicity—the art of maximizing the amount of work not done—is essential.

The best architectures, requirements, and designs emerge from self-organizing teams.

At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.⁴⁷

The agile principles provide a set of *values* for effective software development rather than prescriptive management processes guiding the technical managerial requirements for software development. Agile methods and the principles that guide them recognize the importance of individual interactions between stakeholders in the software development process and incorporate principles that are fundamentally people-oriented and team-oriented. From the guiding agile documents, two of the four items articulated in the Agile manifesto and five of the 12 principles are *people-oriented*, which emphasizes the importance of the “human element” in effective software development, and underscores the importance of delivering value to the customer rather than strict adherence to traditional management processes.⁴⁸

Implementing agile methods to software development projects also provides the opportunity to address technical risks as early in the project lifecycle as possible, which reduces the potential for unforeseen schedule delays or cost overruns. The emphasis on incremental planning for iterative delivery of system capability provides the opportunity for quicker end-user feedback. By obtaining end-user feedback early and often, project teams can make small system changes to the software as the need is identified rather than requiring larger changes later in the project lifecycle, after significant investment and realization that the system does not support end-user requirements.⁴⁹ Furthermore, an early understanding of project risks and ability to

implement incremental adjustments to base system software code prevents necessary system changes and modifications later in the project lifecycle, during system integration testing for example. Agile methods minimize the impact of inevitable changing requirements. Software modifications implemented late in the project lifecycle to accommodate changing requirements typically incur significant cost and schedule impacts due to the complexity of the mature system and the level of effort required to implement a modification and conduct appropriate regression testing to ensure the modification did not have unforeseen impacts throughout the system architecture. Therefore, program risk is significantly reduced by accommodating new or changing requirements based on continuous end-user feedback at the earliest opportunity during the design and development phases of the project.

Agile software development describes generalized *values* that encourage stakeholder collaboration and *principles* that seek to avoid an over-emphasis on rigorous processes, comprehensive documentation, contractual negotiation, defined requirements, and adherence to a development plan. Agile implementation *methods* attempt to establish a framework that remains adaptive to a changing environment and, therefore, differ from those of traditional “waterfall” methods that rely on a more formalized *plan-based* approach to managing the software development effort. Table 1 provides a comparison between a plan-based traditional approach and adaptive agile software development methods.

Traditional Principles	Agile Instantiation
Plan the work—especially the budget, schedule, and deliverables—to the maximum extent possible before beginning any design or code.	<ul style="list-style-type: none"> • Near-term plans contain more detail, while plans further out on the time horizon contain fewer details. • The overall vision is broken down into a roadmap, which is further broken down into release plans, which are further broken down into sprint or iteration plans, which are further broken down into daily plans. • Requirements are prioritized. • Cost and schedule estimates are prepared for each capability at a high level. Relative estimation versus absolute estimation is employed. • Frequent planning sessions (at the beginning of each iteration) result in detailed, high-fidelity plans. • Risks are assessed and risk mitigation influences planning.
Lock down requirements to prevent gold-plating and scope creep.	<ul style="list-style-type: none"> • No requirements can be added to an iteration once it has started. • New requirements are evaluated by the stakeholders and prioritized thus preventing gold-plating and scope creep.
Institute multiple reviews to provide senior leadership oversight as well as to serve as gates for continued work.	<ul style="list-style-type: none"> • The customer is involved in all aspects of planning and testing. Customer (in the form of the product owner) is involved daily. • There are reviews at the end of each iteration that serve as gates to further work.
Move forward in a step-by-step, sequential manner and only when all parts of the previous steps were complete.	<ul style="list-style-type: none"> • The code base is integrated and tested daily. • The code base must pass all tests before and after integration. Regression testing is typically done each night.
Capture all details with extensive documentation.	<ul style="list-style-type: none"> • There is an overall plan. • There are requirements descriptions. • There are cost and schedule estimates. • There are risk assessments. • There is training material (as appropriate). • There is documentation (as appropriate). • There are lessons learned (based on retrospectives).

Table 1: Comparison of traditional principles to agile principles (Source: Palmquist et al., 16)

Types of Agile Methods

Agile is not a single approach nor a prescriptive standard for software development, rather agile methods describe a family of development processes that embody the values and principles expressed by the Agile Manifesto. As defined by the Agile Alliance, “Agile Software Development is an umbrella term for a set of methods and practices based on the values and principles expressed in the Agile Manifesto. Solutions evolve through collaboration between self-organizing, cross-functional teams utilizing the appropriate practices for their context.”⁵⁰

There is no single correct way to “implement agile,” rather program managers can adopt agile methods and practices that best suit the unique circumstances and environment of their program. Agile methods include various software development techniques and management frameworks; the most widely used agile methodologies within industry include Scrum, Extreme Programming (XP), Kanban, and Test Driven Development (TDD). According to Version One’s most recent 2016 “State of Agile Development Survey,” out of all agile methodologies in use, 70% of respondents stated they incorporated Scrum or Scrum/XP hybrid, 39% stated they used Kanban, and 33% stated their organizations incorporated TDD agile methodologies in IT projects.⁵¹

More recently developed agile methodologies include scaling frameworks such as the Scaled Agile Framework (SAFe), which is intended to implement agile principles at an enterprise level. SAFe extends agile methods to the portfolio level and introduces a more strategic perspective to the practice, rather than focus on improvements at the software developer level. Using SAFe, large hierarchical organizations can develop a roadmap that outlines the appropriate roles for executives and project managers, beyond the development team, to succeed in implementing an agile approach to system development. Furthermore, SAFe is intended to synchronize the alignment, collaboration, and delivery for a large number of agile teams across an enterprise.⁵² Given the organizational complexity of defense IT acquisition processes, within DoD defense contractor community SAFe is considered the most adopted scaling framework.⁵³

Organizational Benefits of Adopting Agile Methods

Agile has emerged as the leading industry software development methodology and has been successfully implemented by a variety of businesses in industries varying from high-

technology start-up companies to mature enterprises in the financial services industry, professional services industry, telecommunications, and manufacturing for example. The Scrum agile methodology is used by a number of diverse organizations worldwide, including Adobe, Barclays Global Investors, the BBC's New Media Division, BellSouth, Bose, Capital One ® , GE, Google, Microsoft, Motorola, Nokia Siemens Networks, SAP ® , State Farm ® and Yahoo!⁵⁴

Successful implementation of agile practices in the commercial sector has reduced likelihood of IT project budget overruns and schedule delays while adapting to evolving end-user requirements to deliver needed capability that provides business value to the customer. In 2016, organizations that adopted agile methodologies reported that the top benefits of adopting agile include: accelerated product delivery, enhanced ability to manage changing priorities, increased productivity, and enhanced software quality.⁵⁵ Cooke outlines both “strategic” and “tactical” benefits for adopting agile methods in industry software development and IT project management processes, which correlate to benefits that could be realized in the implementation of agile practices for defense business systems. At the strategic level, benefits include:

- Ongoing risk management
- Ongoing control of budget expenditures
- Rapid delivery of tangible outcomes
- Strong Alignment with business requirements
- Focus on the highest-priority features
- Responsiveness to business change
- Transparency in status tracking
- Substantially higher-quality outputs

- Greater employee retention
- Minimized whole-of-life software costs⁵⁶

Tactical benefits identified include:

- Production of more valuable outputs per resource hour
- Earlier identification of technical issues
- Less rework and “throw-away” work
- Reduced need to create—and maintain—detailed documentation
- Greater flexibility
- More staff autonomy
- Greater job satisfaction
- Greater opportunity for innovation
- Reduced dependency on paper status reports
- Reduced demand for ongoing support and maintenance⁵⁷

Research evaluating the implementation of agile software development and IT management methodologies in industry has demonstrated quantitative improvements in reduced costs, improved schedule performance, and improved software quality for enterprise IT delivery and management. An extensive survey comparing the quantitative costs and benefits between traditional (i.e. waterfall) methods with agile methods in industry was conducted by Rico; the study summarizes 69 scholarly studies and academic research on the basis of cost, schedule, productivity, quality, and satisfaction to develop a quantitative return on investment (ROI) comparison. The study found that, on average, agile methodologies reported a 29% better cost, 91% better schedule performance, 97% better productivity, 50% better quality, 400% better end-

user satisfaction, and a 470% improved ROI than traditional development methods.⁵⁸ The findings of the exhaustive research suggest that “agile methods elevate business value beyond just the activities of creating working software at regular intervals—Agile Methods go on to define business value in terms of ROI.”⁵⁹

A survey conducted in 2006 that sampled 4,232 IT industry respondents compared adoption of new agile approaches to traditional software development methodologies and demonstrated observations that agile software development methods and techniques contributed to organizational improvement. 60% of respondents who had adopted agile methodologies in their organizations reported increased productivity, 66% reported increased quality, and 58% reported improved business stakeholder satisfaction.⁶⁰ A subsequent comparative survey conducted in 2008 that sampled 642 IT industry respondents indicated that 82% reported increased productivity, 87% reported enhanced quality, and 78% reported improved business stakeholder satisfaction. The 2008 survey also reported 72% of respondents observed reduced costs of system development.⁶¹

More recent survey data was collected between July and November, 2015 by Version One—international in scope and considered one of the most comprehensive overviews of the use of agile development practices—and analyzed 3,880 responses. Among the participants, 87% found agile improved the ability to manage changing priorities, 85% found increased team productivity, and 84% found improved project visibility.⁶² Collectively, these surveys demonstrate the improvements and anticipated gains in the areas of productivity, system quality, stakeholder satisfaction, and reduced development cost for organizations in the private sector that adopted agile methods. The number of enterprises that have adopted agile software development and project management methodologies continues to increase and is driven

primarily by a desire for accelerated product delivery and enhancing the enterprise's ability to handle changing priorities.⁶³

The federal government and defense industry has begun to implement agile in IT acquisition and governance and has achieved similar successes as the private sector. Examples of agile implementation within the defense industry include the Coast Guard Logistics Information Management System (CG-LIMS),⁶⁴ the Global Combat Support System-Joint (GCSS-J),⁶⁵ and the Department of the Air Force Patriot Excalibur (PEX) program.⁶⁶ By adopting agile methods, these programs have successfully controlled costs and remain responsive to their broad user communities. The DoD must continue to adopt agile software development practices, particularly for defense business systems that share many commonalities with IT projects in the private sector, to reduce costs, improve performance, and remain responsive to the developing needs of the end-users. To ensure success, program managers must develop effective strategies that prepare organizations to transition to agile principles and methods, cultivate an effective team environment and organizational culture, and prepare an adequate contracting strategy that supports the iterative nature of agile software development. The following section provides a framework to address those critical considerations to facilitate an effective agile implementation strategy.

Critical Considerations for Agile Implementation Strategies in the DoD

The potential benefits for adopting agile practices in industry have been demonstrated and over the last decade industry enterprises embraced agile methods to reduce costs, accept and adjust to changes in project scope and requirements, improve customer satisfaction, and provide timely delivery of functional software to the end-user. The agile process values constant collaboration, frequent delivery of functional capability, and accepts the continuous evolution of requirements.⁶⁷ The DoD must transition from traditional waterfall SDLC models and embrace agile software development principles and methods to improve the IT acquisition process. To be effective, DoD executives and program managers leading initiatives to adopt agile methods must develop effective implementation strategies within the constraints of the DoD acquisition policy, and provide guidance that adequately prepares teams for agile while aligning organizational culture with agile principles. To be effective, the DoD must adopt agile governance practices, not just change software development practices. To develop and apply effective agile implementation strategies, program managers must focus on three critical areas to ensure success: deciding and preparing to transition to agile principles and methods; developing an effective team environment and organizational culture; and preparing an adequate contracting strategy that supports the iterative nature of agile software development.

Preparing to Implement Agile Principles and Methods

Program managers must decide if “going agile” aligns with organizational goals and must prepare a *strategic plan* if adopting agile is a recommended course of action. Agile methods are not necessarily the most effective approach for all DoD IT acquisition projects. The DoD should embrace agile principles and methods for some programs and traditional waterfall methods for

others, but decisions should default toward application of agile methods.⁶⁸ Traditional plan-driven approaches to software development in national security systems or weapon systems, for example, may be preferred due to stable requirements, higher system complexity and a lower risk tolerance due to the sensitivity of the project. Safety-critical systems, such as the software development efforts associated with DoD weapon system acquisition, may require a traditional, more rigid, and plan-driven approach to ensure the system accurately meets desired requirements. However, because defense business systems by nature provide enterprise level IT services supporting the *business requirements* of a broad user community whose needs may change with the environment, agile principles and methods are generally recommended. In considering to “go agile,” program managers must decide to what extent and which agile practices will be most effective given program objectives and scope.

Depending on an IT project’s specific needs, a hybrid combination of agile and plan-based “waterfall” methods may be the most appropriate software development lifecycle strategy.⁶⁹ Boehm and Turner conclude that there are five critical factors for determining the suitability for adopting agile methods in particular project situations and provide a useful tool that can be adopted within the DoD to help determine whether an agile or traditional plan-driven approach (i.e. waterfall approach) is more appropriate (see Figure 5).⁷⁰ An assessment of the planned project and existing team based on five critical factors—size, criticality, dynamism, personnel, and culture—can help determine if a given acquisition project is a good fit for agile implementation or traditional plan-driven methodologies, and can improve the likelihood that the project will be a success. Program managers leading agile implementation on defense business systems must recognize when factors may inhibit agile implementation and address shortfalls in strategic planning. Furthermore, program managers must learn to blend traditional plan-based

“waterfall” methods with adaptive agile principles and practices in the right proportions to fit a given situation.⁷¹

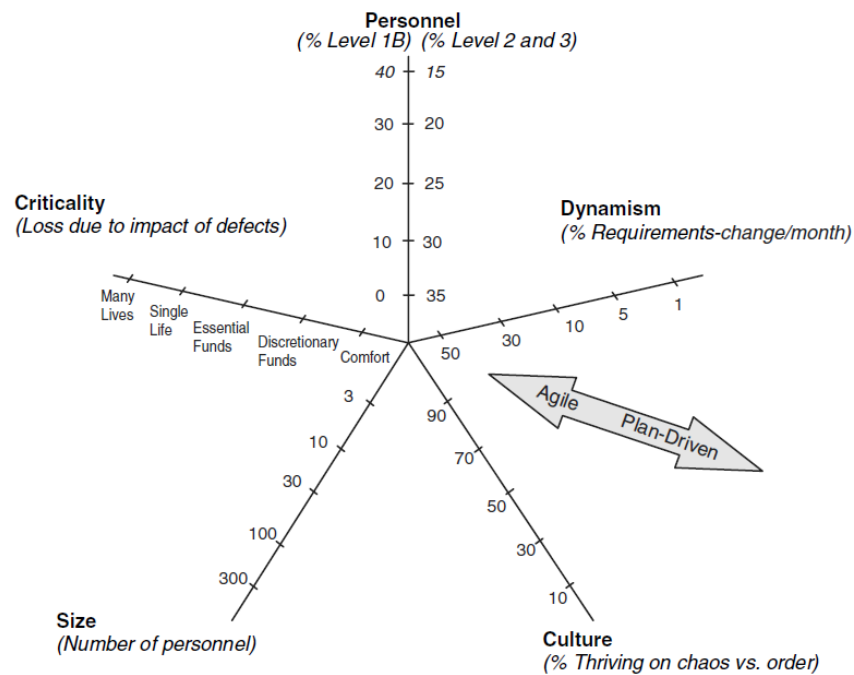


Figure 5: Dimensions Affecting Method Selection (Source: Boehm & Turner, 2004, pg. 56)

Strategic planning describes the overall implementation plan for an organization and prepares the organization to adopt or transition to an agile approach. Importantly, the decision to adopt agile principles and methods on DoD defense business systems can occur throughout the program’s lifecycle, not necessarily at an early decision point in the programs lifecycle. Strategic planning for transitioning a project from traditional methods to agile should follow organizational change management disciplines to ensure the transition to agile is effectively communicated to all stakeholders.⁷² Following change management discipline to transition organizations managing defense business systems includes establishing a sense of urgency,

creating a guiding coalition to lead the transition, and developing and communicating a clear vision and strategy for adopting agile.⁷³ These initial steps prepare organizations for the cultural change necessary to successfully adopt agile principles and methods. Furthermore, strategic planning should consider a *gradual* migration that combines agile with existing methods. For example, program managers can conduct pilots on specific projects or increments within the program to gradually transition to agile. Finally, while Scrum agile methods are gaining popularity within the defense industry, strategic planning must include the decisions on which agile practices, or combination of methods, best fit the particular program and tailor those approaches to the specific requirements of the defense business system.

Finally, to develop an effective agile implementation strategy for DoD IT acquisition, program managers must prepare to incorporate appropriate guidance *upfront* to create expectations for the level and type of documentation and criteria that meet the statutory requirements and policy guidance from the defense acquisition system. While implementing agile methods encourage iterative development and release schedules and have an advantage of flexibility over traditional approaches, DoD IT acquisition will remain constrained by acquisition policy outlined in the DoDI 5000.02 and the statutory requirements of the Federal Acquisition Regulations (FAR). Therefore, consideration of what agile processes and methods to use must occur early in the acquisition lifecycle for a program or increment to ensure that contractually binding documents, such as Request For Proposals (RFP) or the Statement of Work (SOW), adequately correspond to and support those practices and methods.⁷⁴ Furthermore, due to the emphasis on an iterative approach to system design and development, implementing an agile approach will likely result in incomplete documentation expected during the standard DoD acquisition milestone reviews (e.g. Preliminary Design Review (PDR), Critical Design Review

(CDR), etc.). Lapham et al addresses these concerns and recommends that the Systems Engineering Management Plan (SEMP) and Software Development Plan (SDP) provide specific guidance for when certain activities and reviews occur in order to align those activities to relative agile iterations.⁷⁵ Furthermore, the report recommends that “The PMO should communicate frequently with the contractor about program goals, provide timely feedback on iterative software capabilities as they are developed, and work with the contractor to ensure changes are well understood and smoothly incorporated as required and approved.”⁷⁶ Additionally, program managers must coordinate with the lead program engineer to consider how to integrate agile principles and practices within systems engineering disciplines and define expectations for documentation and artifacts from the program’s systems engineering technical review (SETR) process. Agile implementation strategies must provide the appropriate guidance upfront that accounts for policy and requirements of the DoD acquisition system and FAR, and facilitate a collaborative environment between the government PMO and contractor to ensure the success of the project.

Team Environment and Organizational Culture

Promoting the team environment and developing an organizational culture that accommodates agile methodologies and principles while accepting an agile “mindset” is critical for success. To effectively implement agile, program managers must lead change to organizational culture to accept agile governance practices, rather than proscribe “agile” software development practices. Implementing agile is a shared mindset across the team that embodies the principles of the Agile Manifesto and Agile Principles.⁷⁷ To effectively develop an organizational culture that adopts agile principles and methods, Program Managers should focus on: 1) training and coaching teams; 2) fostering “agile teams” through stakeholder collaboration;

3) empowering small, autonomous cross-functional teams; 4) creating balance between the demands of the defense acquisition system and the responsiveness of agile software development; and 5) understanding and influencing organizational culture to adopt an agile *mindset*.

Properly training and coaching teams to implement agile methods in DoD IT acquisition projects, while creating a culture that overcomes the resistance to change and actively adopts agile principles is critical for success. A 2016 report from SEI found that, despite significant use of agile methods in the commercial sector, relatively few defense acquisition professionals are familiar with agile or possess the necessary experience to implement agile methods.⁷⁸ Indeed, agile-related training offered by the Defense Acquisition University (DAU) remains insufficient; no DAU courses specifically focus on agile methodologies for program management⁷⁹ and only three courses (IRM 202, SAM 301, ENG 301) introduce agile-related material to the acquisition workforce. In the private sector on the other hand, professional training for implementation of agile principles and practices is widely available: the Program Management Institute (PMI) has created a new PMI-ACP (Agile Certified Practitioner) certification, the ScrumAlliance® provides a number of professional certifications including Certified ScrumMaster (CSM) and Certified Scrum Professional (CSP), and various education institutions offer courses in agile implementation to include offering Masters certificates in agile principles.

The emphasis on professional development and agile training in the private sector originates from examples of successful agile implementation strategies that reinforced the importance of agile *training* and *coaching* throughout implementation. For example, in 2005 Yahoo! successfully integrated agile methodologies in organizational software development and project management initiatives for several products ranging from new development projects to

established infrastructure (e.g. Yahoo! Mail). Lessons learned from the Yahoo! agile roll-out, which are applicable for initiatives in the DoD, included the importance of proper *training* for the agile teams and *coaching* throughout the implementation process.⁸⁰ The Yahoo! agile roll-out focused on implementation of Scrum agile methods within the organization; the success of the roll-out encouraged other teams within the organization to attempt to “go agile,” but without the proper training or coaching to properly implement and execute the Scrum methodologies. As a result, those teams failed to take adequate steps to change and improve on traditional waterfall methods. Effective agile implementation strategies in the IT industry, therefore, emphasize adequate training and proper coaching of development teams to ensure a successful transition from traditional plan-based methods to agile software development methods.

The implication and recommendation for DoD projects attempting to incorporate agile methods is to ensure *upfront comprehensive training*—such as Certified ScrumMaster training—is available for acquisition practitioners on the project. Furthermore, when implementing Scrum agile methods, dedicated agile coaches must be available and integrated into the IT project to facilitate key events such as the iteration planning, daily stand-up meetings, retrospectives, and working with team members as needed.⁸¹ Hands-on assistance from an agile subject matter expert will be required during the early stages of implementation; a 2010 Mitre report recommends including agile training in the project plan and hiring an agile coach to assist and guide the project team through the initial stages of agile implementation.⁸² A reinforcing example in the DoD is the Air Force Patriot Excalibur (PEX) program, which emphasized proper upfront coaching and team development when attempting to implement agile software development practices within the PMO. One significant factor that facilitated the success of the PEX program to adopt agile methods that reduced the software release cycle and better met end-

user requirements was the PMOs commitment to outsource professional consultants and halt production in order to sufficiently train the PEX team how to implement agile methods.⁸³ Outsourcing an agile coach to facilitate initial agile training is recommended. Additionally, ensuring an informed agile “advocate” is integrated into the PMO staff will enhance the chances of program success. The persistent presence of that advocate facilitates answering daily questions, helps immediately resolve issues with the agile process, and helps ensure the program runs efficiently when incorporating agile methods into PMO processes.⁸⁴ The program’s agile “advocate” can facilitate the integration of agile principles and methods and between the government PMO and designated contractor, but program managers should ensure the advocate is empowered and has the legal authority to direct the contractor. To effectively integrate agile principles and methods throughout the PMO team, program managers must ensure upfront comprehensive training is available and an agile advocate is identified to consistently manage the implementation, which avoids reverting to traditional software development methods.

Developing the appropriate environment and culture within the DoD Program Management Office (PMO) that fosters collaboration between system end-users, the government acquisition community, and industry partners to accommodate agile principles and methods is essential for implementing agile methodologies in DoD IT acquisition. In his retrospective report “Lessons in Agility from Internet Based Development,” Scott Ambler highlights the importance of open communication within the development team that encourages dialog and sharing of ideas and information.⁸⁵ To ensure the development team obtains the *direct feedback* from the end-user’s *hands-on review* of working software, agile implementation efforts within DoD IT programs will require a team environment where developers work in a *collaborative environment* that supports open dialog with both government PMO leads and the designated system subject

matter experts (SME) who fundamentally understand the business needs and system requirements. Program managers must invest in hiring effective SMEs with recent and relative field experience who can act as user surrogates, and ensure that these PMO SMEs are integrated into routine agile practices (i.e. daily Scrum meetings) on the program. Furthermore, the program manager should ensure that mechanisms are put into place that facilitate dialog between the development team and DoD system end-users in the field. The PMO should proactively host reoccurring *user forums* to elicit new needs or recommendations from end-users and capture those requirements in system “user stories.” An example of an opportunity to conduct that necessary dialog in the United States Marine Corps occurs at reoccurring Operational Advisory Group (OAG) meetings held to discuss program lifecycle implementation strategies and capture end-user feedback from operating force components.¹ During the user forums, end-users should be invited to observe all system demonstrations where they have an opportunity to provide critical feedback, directly from the end-user perspective, that can become binding with PMO approval.⁸⁶ Program managers and leadership must ensure that the culture within the PMO is developed that facilitates collaboration among all project stakeholders and ensures the software development teams have the opportunities and access to the end-user’s direct feedback in order to effectively adjust to clarified or changing system requirements.

Project managers leading change and implementing agile methods in DoD IT acquisition projects must focus on establishing *small* effective teams that operate in an environment that facilitates the active dialog, independence and relative autonomy required in an agile framework.

¹ The Global Combat Support System-Marine Corps (GCSS-MC) PMO recently implemented agile principles for software development and maintenance activity and routinely disseminates information and receives user feedback through the biannual Deputy Commandant (DC) Installations and Logistics (I&L) led GCSS-MC OAG.

Jeff Sutherland, a co-creator of the agile Scrum method, advocates implementing a set of practices that facilitates individual and team autonomy to make decisions on how to take action, and that project managers allow sufficient ability for the team to improvise, within acceptable project constraints, in order to adjust to changing requirements and meet system end-user demands.⁸⁷ Small teams of 7 to 18 people who are empowered to decide what capability to deliver and how to produce it was recommended by the GAO as a best practice for adopting agile methods.⁸⁸ Defense IT projects also typically rely on distributed teams that may be dispersed across the country, or world, supporting various needs of the Enterprise. Ensuring collaborative tools (e.g. VerionOne or JIRA software development tools) are available and adequately integrated into business practices is essential to establishing the collaborative environment necessary for successful implementation of agile methods. Program managers implementing agile methods in the DoD must ensure small teams are empowered, adequately structured, and have the appropriate resources to implement agile software development methods.

Adapting organizational culture in defense IT programs to incorporate agile principles and methods requires balance between the demands of the defense acquisition system and the responsiveness of agile software development. Program managers leading change will need to balance the demands of *hierarchical* organizational culture with the flexibility of a *cooperative* organizational culture. Organizational culture within the defense acquisition workforce tends to be influenced by a *hierarchical organization* where people in certain positions have clear divisions of responsibility and authority and are responsible for discrete activities. These organizations tend to attempt to control change through rigid application of procedures and tools to capture, track and manage change.⁸⁹ The nature of defense acquisition tends toward this type

of environment. Examples of IT governance practices in defense business systems include configuration control boards (CCB) or the rigid application of the SETR process to document and manage software development efforts and software baseline changes in IT programs.

Cooperative organizations, on the other hand, describe an environment where ad hoc teams develop to resolve problems and directly address end-user requirements. Cooperative organizations value the satisfaction of their end-users and react to the changing requirements of the organization and end-users. The organizational culture in high-tech startup companies, where a flattened structure encourages collaboration, tends to be defined as cooperative organizations. However, because cooperative organizations focus predominately on customer satisfaction, processes are not well defined to keep track of changes, identify and track costs, or clearly establish schedule constraints, which can result in unpredictable project budgets and schedules and, consequentially, increase project risk.⁹⁰ Figures 6 and 7 illuminates the differences between the standardized, sequential approach in hierarchical organizations compared to the dynamic, ad-hoc teamwork in cooperative organizations.

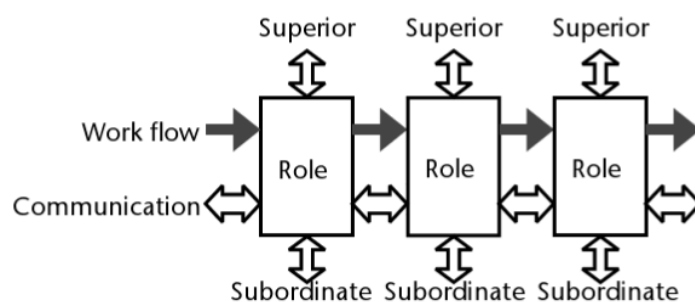


Figure 6: Hierarchical Organizations (Source: Koch, 10)

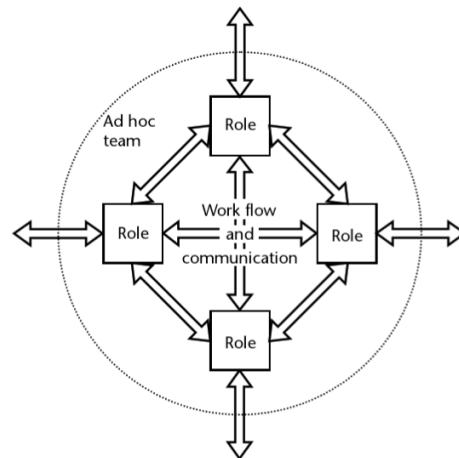


Figure 7: Cooperative Organizations (Source: Koch, 11)

Agile methods require an environment that encourages the collaboration, ad-hoc teamwork, and the responsiveness characteristic of *cooperative organizations*, which can create challenges for hierarchical controlling type organizations attempting to implement agile. However, agile methods do have certain hierarchical elements: to ensure roles are defined between project teams, to facilitate iterative planning processes (i.e. daily Scrum meetings), or control prioritization of work (i.e backlog prioritization), for example. Program managers implementing agile on defense business systems must create a balance between the demands of *hierarchical organizations*—the cost and schedule management requirements inherent in the defense acquisition system—and the need for an atmosphere that encourages initiative, collaboration, and responsiveness to changing end-user requirements. Because implementing agile methods will result in project plans that do not accurately capture all system requirements, and assumes that both the development team and the end-user will learn and adjust to the evolving needs, program managers must accept that those initial project plans are approximations, and subject to inherent cost and schedule risks. To control those risks, many agile methods use *time-boxing techniques*, such as a Scrum “Sprint,” that define beginning and

end dates for a team to accomplish specific tasks within a software release iteration, and allows the customer to prioritize the functionality that will be delivered.⁹¹ To successfully implement agile principles and methods in defense business system acquisition projects, program managers must create a collaborative and responsive organizational culture that integrates the programmatic tools necessary to satisfy the requirements of the defense acquisition system.

Successfully implementing agile principles and methods will require influencing organizational culture to adopt an “agile mindset.” The defense acquisition workforce is invested in the use of traditional water-fall models for the acquisition of defense weapon systems and equipment, and as a result a significant segment of the acquisition community is comfortable with traditional waterfall methods and skeptical of agile.⁹² Program Managers adopting agile methods must recognize that an agile *culture* is quite different from the traditional “waterfall culture.” Culture drives how an organization collectively acts and responds to change and is ingrained through organizational structure, leadership structure, reward system, communication and decision making styles, and the staffing model.⁹³ As previously discussed, agile methodology provides a set of *values* for effective software development *not* a menu of prescriptive management procedures for DoD software development projects. Successfully adopting agile requires overcoming the potential resistance to a change in culture within organization, and developing a mindset change throughout the organization—rather than just a practice change. Program managers should consider implementing change models (e.g. Kotter’s “Eight Step Process”) to incrementally introduce and reinforce agile principles within the PMO. Visible *artifacts* of agile culture, such as user story boards or project burndown charts, should be displayed to foster a collaborative agile environment. An important value behind agile is openness and transparency to ensure that essential information is allowed to flow easily and

automatically throughout the team using the concept of an *information radiator*.⁹⁴ Information radiators are large, highly visible displays of both progress and problems that are continuously updated to ensure all team members are kept informed real-time as work is completed. These artifacts and communication tools help facilitate collaborative self-organization within the agile team. Adopting the Agile Manifesto to the program's unique circumstances and ensuring the artifact is displayed in the work place further helps develop shared values throughout a team. Program managers attempting to adopt the principles and methods of agile must recognize that success requires change in organizational culture, which demands a persistent and dedicated effort.

Agile Contracting and Oversight

Finally, program managers implementing agile software development methods must prepare an adequate *contract strategy* that supports the iterative nature of agile software development. Implementing agile software development practices on defense business systems requires a *flexible contract vehicle* and adequate oversight mechanisms that effectively document and monitor contract performance. The fundamental principles outlined in the Agile Manifesto that guide agile implementation require flexibility based on user feedback and accepts changing requirements. Whereas DoD acquisition projects that follow the traditional waterfall systems engineering approach assume the product's end-state is well known and the requirements are stable at a relatively early stage in the project, IT acquisition projects implementing an agile approach will not have a clearly defined end-state upfront. Therefore, traditional contract strategies that align with waterfall development methods and require specific artifacts and time-driven milestone reviews may not support the frequency of changes and product demonstrations of an iterative agile approach.⁹⁵ Agile accepts constant re-prioritization and changing system

requirements, which inherently creates uncertainty and increases risk for DoD IT programs that leverage traditional contracting approach. Effectively implementing agile in DoD IT acquisition, therefore, requires an appropriate contracting approach.

Methods of contracting for traditional waterfall acquisition projects will generally seek to develop a baseline upfront and control changes; contracting for agile projects on the other hand should encourage and welcome the continuous evolution of system requirements.⁹⁶ Program managers implementing agile should consider aligning funding to specific software releases that correspond to a predetermined time-box, rather than focus on aligning funding to functionality. Program managers should consider Indefinite Delivery Indefinite Quantity (IDIQ) Delivery Orders for agile projects to increase flexibility to adjust to evolving requirements and changing operational needs. IDIQ vehicles are managed similar to a service contract, where the service provided is the “as-required” software development evolution, rather than a product contract.⁹⁷ Cost reimbursement contracts, rather than Fixed Price (FP) contracts, are recommended for government agile contracting efforts to accommodate evolving requirements and the expected redirection and reprioritization of resources throughout the development process that an FP contract would likely fail to accurately capture.⁹⁸ There is no “one-size-fits-all” approach in developing a contract strategy on an agile project; program managers and contracting officers must work collaboratively to develop a strategy that accepts change as part of the norm.

Contract oversight on agile programs also creates unique conditions that can challenge acquisition professionals and contracting officers more familiar with traditional methods. Agile approaches tend to limit documentation to that which adds direct value to the development and sustainment of the system. Each iteration of software release in an agile framework will have documentation describing the “definition of done,” and system architecture and design

documentation is updated “mid-step” during the process. The result is often that the proposed documentation under an agile contract is considerably light compared to traditional methods.⁹⁹ Therefore, when deciding the appropriate contract vehicle for agile projects, the program manager must coordinate with the contracting officer *upfront* to demonstrate how project documentation relates to program execution.¹⁰⁰ Additionally, given the emphasis on iterative development work and acceptance to changing requirements, the program manager and contracting officer or representative must facilitate a collaborative partnership between the PMO and contractor to ensure a high level of transparency and sharing of information related to the project, which increases a shared understanding of program risks and accuracy of project estimates.¹⁰¹ Contract strategies for agile projects must ensure project documentation requirements are identified upfront and information is shared between all stakeholders to ensure transparency and reduce project risk.

A final consideration program managers must consider when developing contract strategies for agile projects is how to ensure performance measuring and tracking. Traditional methods to determine contractual cost/schedule estimates and measure performance rely on metrics that described software size, such as Source Lines of Code (SLOC) or “functional points.” Under a traditional approach, the volume of code produced will therefore be used as an indicator of project progress. In an agile approach, software sizing is not used as a basis for cost/schedule estimates, rather the preference is to focus on the time-box (i.e. Scrum Sprint) iterations that have a predetermined, fixed schedule and cost. Agile Cost/schedule estimates and performance metrics are based on the number of “story points” the software development team delivers at the end of each time-box iterations (i.e. Sprints). This approach relies on end-user prioritization and relative-effort sizing techniques to ensure the maximum value for the end-user

is delivered in each iteration, or software release.¹⁰² To measure performance and identify appropriate metrics for a contract data requirements list (CDRL), program managers must identify what data are available from the contractor's automated tools that meet objectives for monitoring progress and software quality. Effective agile contract strategies must ensure adequate performance measuring and tracking metrics are identified upfront to ensure transparency and effective contractual oversight.

Conclusion

Recent acquisition reform efforts created opportunities in the DoD to improve the efficiency and effectiveness of software development efforts and encourages use of industry best practices such as agile software development methods. As a result, DoD IT programs have begun to incorporate aspects of agile software development to improve program performance. However, the new DoD guidance lacks specific instruction for developing an effective strategy for implementing agile software development methods on DoD IT programs. Defense business systems that provide enterprise level IT services supporting the business requirements of broad DoD user communities share many similarities to business systems used in the private sector that successfully leverage agile principles and methods, and should therefore transition from traditional software development methods to an adaptive agile approach.

Successfully implementing agile software development techniques on defense business systems requires more than adopting new technical engineering techniques. Rather, success requires that executive leadership and program managers develop agile implementation strategies that focus on developing a collaborative environment between all involved stakeholders to transition teams to an agile framework and align organizational culture with agile principles. To be effective, program managers must adopt agile governance practices, not just agile software development practices. To ensure success, program managers must develop agile implementation strategies that incorporates strategic planning to prepare organizations to transition to agile; develop and cultivate a collaborative agile team environment and organizational culture; and prepare appropriate contract strategies that support the dynamics of agile software development. Traditional software development methods consistently failed to deliver timely, cost-effective software that adequately met end-user needs. The proven benefits of agile software

development—if implemented correctly on defense business systems—can control costs, prevent unnecessary project delays, improve software performance, and address customer satisfaction for DoD IT programs. Now is the time for defense business systems that provide enterprise-level IT services to proactively implement agile software development best practices and improve results for *all* involved stakeholders.

¹ Jamie Lynn Cooke, *Everything You Want to Know about Agile: How to Get Agile Results in a Less-than-Agile Organization*, (Cambridge, GB: IT Governance Publishing, 2012), 31

² Government Accountability Office (1), *Effective Practices and Federal Challenges in Applying Agile Methods*, (Washington DC: Government Accountability Office, 2012), 15

³ National Research Council, *Achieving Effective Acquisition of Information Technology in the Department of Defense*, (Washington, DC: National Academies Press, 2010), vii

⁴ Gordon E. Moore, “Cramming more components into Integrated Circuits,” *Proceedings of the IEEE* 86, no. 1 (January 1998): 1, <https://www.cs.utexas.edu/~fussell/courses/cs352h/papers/moore.pdf>

⁵ Colin Clark, “Army Cyber Chief Meets Buyers In Pursuit Of Faster Acquisition,” *Breaking Defense*, last modified October 22, 2013, <http://breakingdefense.com/2013/10/army-cyber-chief-meets-buyers-in-pursuit-of-faster-acquisition/>.

⁶ Cheryl Pellerin, “CIO Priorities Include Cybersecurity, Innovation, Retaining IT Workforce,” *DoD News, Defense Media Activity*, last modified March 22, 2016, <https://www.defense.gov/News/Article/Article/702488/cio-priorities-include-cybersecurity-innovation-retaining-it-workforce>.

⁷ National Research Council, 1

⁸ US Department of Defense, *A New Approach for Delivering Information Technology Capabilities in the Department of Defense*, (Washington DC: Office of the Secretary of Defense, November 2010): 4 & 12

⁹ *Ibid*, 12

¹⁰ Suzanne Miller and Dan Ward, *Update 2016: Considerations for Using Agile in DoD Acquisition*, Hanscom, MA: Software Engineering Institute: 2016

¹¹ John M Gilligan, “A Roadmap for Reforming the DoD’s Acquisition of Information Technology,” *Software Tech News* 15, no. 1 (retrieved: November 2016): 1

¹² National Research Council, 29

¹³ *Ibid*, 29

¹⁴ US Department of Defense Science Board, *Final Report of the Defense Science Board Task Force on Department of Defense Policies and Procedures for the Acquisition of Information Technology*, (Washington DC: Office of the Undersecretary of Defense for Acquisition, Technology, and Logistics, March 2009): 35.

¹⁵ Winston W Royce, “Managing the Development of Large Software Systems.” *Proceedings of the IEEE Westcon, Washington, D.C. 1970* (August, 1970): 1-9.

¹⁶ *Ibid*, 1-9

¹⁷ Daniel E. Schoeni, “Long on Rhetoric, Short on Results: Agile Methods and Cyber Acquisitions in the Department of Defense,” *Santa Clara Computer and High Technology Law Journal* 31, no. 3, (July 1, 2015): 398

¹⁸ National Research Council, 48

¹⁹ Carlton Northern et al, *Handbook for Implementing Agile in Department of Defense Information Technology Acquisition*, (Mitre Corporation: December 15, 2010): 2-3, <http://www.dtic.mil/docs/citations/ADA546756>.

²⁰ *Ibid*, 2-3.

²¹ *Ibid*, 2-4

²² Jamie L. Cooke, 35

²³ *Ibid*, 35

²⁴ Brooks, F.P. “Report of the Defense Science Board Task Force on Military Software.” (Washington DC: Office of the Secretary of Defense, 1987): 3

-
- ²⁵ National Research Council, 49
- ²⁶ Subhas Misra, "Agile Software Development Practices: Evolution, Principles, and Criticisms." *The International Journal of Quality & Reliability Management* 29, no. 9 (2012): 972-980, 974.
- ²⁷ National Research Council, 49
- ²⁸ Subhas Misra, 974.
- ²⁹ Ibid, 974
- ³⁰ National Research Council, 50
- ³¹ Ibid, 47
- ³² Government Accountability Office (2), *OMB and Agencies Need to Improve Planning, Management, and Oversight of Projects Totaling Billions of Dollars*, (Washington DC: Government Accountability Office, 2008), <http://www.gao.gov/new.items/d081051t.pdf>.
- ³³ John M Gilligan, 2
- ³⁴ US Department of Defense Science Board, 1.
- ³⁵ Ibid.
- ³⁶ Ibid, x
- ³⁷ Ibid, xi
- ³⁸ National Defense Authorization Act for Fiscal Year 2010, HR 2647, 111th Cong., Congressional Record 111, no. 288, (October 7, 2009).
- ³⁹ US Department of Defense, *A New Approach for Delivering Information Technology Capabilities in the Department of Defense*, (Washington DC: Office of the Secretary of Defense, November 2010): 4
- ⁴⁰ See Broadus, pg. 6.
- ⁴¹ Jamie Lynn Cooke, 29
- ⁴² Ibid, 32
- ⁴³ Beck et al, "Manifesto for Agile Software Development," *Agile Alliance*, 2001, <http://agilemanifesto.org/>
- ⁴⁴ Jamie L. Cooke, 23
- ⁴⁵ David F Rico, What is the ROI of (ROI) of Agile Methods?, 2, <https://pdfs.semanticscholar.org/7453/b6083e71a3caf48168f5c3af9df2d09a4dbc.pdf>
- ⁴⁶ Charles G. Cobb, *The Project Manager's Guide to Mastering Agile: Principles and Practices for an Adaptive Approach*, (Somerset, US: John Wiley & Sons, Incorporated, 2015), 90
- ⁴⁷ The Agile Alliance, "12 Principles Behind the Agile Manifesto," <https://www.agilealliance.org/agile101/12-principles-behind-the-agile-manifesto>
- ⁴⁸ Carlton Northern et al, 3-7
- ⁴⁹ Jeff Sutherland, *Scrum: The Art of Doing Twice the Work in Half the Time*, (New York: Crown Business, 2014), 197
- ⁵⁰ Agile Alliance, <https://www.agilealliance.org/agile101/>
- ⁵¹ VersionOne, Inc., "10th annual state of Agile survey: State of Agile development," VersionOne, 2016, <http://info.versionone.com/state-of-agile-report-thank-you.html>
- ⁵² "Overview of the Scaled Agile Framework® for Lean Software and Systems Engineering," Scaled Agile, Inc. Whitepaper, 1, <http://www.scaledagileframework.com/introduction-to-safe/>
- ⁵³ Suzanne Miller, 9
- ⁵⁴ Jamie L. Cooke, 65
- ⁵⁵ VersionOne, Inc. (2016). 10th annual state of Agile survey: State of Agile development. Retrieved January 11, 2017 from <http://info.versionone.com/state-of-agile-report-thank-you.html>.
- ⁵⁶ Jamie L. Cooke, 37-39
- ⁵⁷ Ibid, 39-43
- ⁵⁸ David F. Rico, 3.
- ⁵⁹ Ibid, 5
- ⁶⁰ Scott Ambler, "Agile Adoption Rate Survey," (March 2006), <http://www.ambyssoft.com/scottAmbler.html>.
- ⁶¹ Ibid.
- ⁶² VersionOne Inc., 2.
- ⁶³ Ibid, 2
- ⁶⁴ Miller, 7
- ⁶⁵ Government Accountability Office (1), 29

-
- ⁶⁶ Suzette Johnson, et al, "The Business Case for Agile Methods," *Association for Enterprise Information*, 5, <http://www.afei.org>
- ⁶⁷ Murray, Anna P. "The Complete Software Project Manager: Mastering Technology from Planning to Launch and Beyond" (Hoboken, NJ: John Wiley & Sons, 2016): 34, <http://site.ebrary.com/lib/usmcu/docDetail.action?docID=11151664&ppg=34>.
- ⁶⁸ Suzanne Miller, v
- ⁶⁹ Steven M. Palmquist et al, *Parallel Worlds: Agile and Waterfall Differences and Similarities*, Hanscom, MA: Software Engineering Institute: 2016, 3
- ⁷⁰ Boehm, B & Turner, R, *Balancing Agility and Discipline: A Guide for the Perplexed*. Addison-Wesley, 2004, 54-57
- ⁷¹ Charles G. Cobb, *The Project Manager's Guide to Mastering Agile: Principles and Practices for an Adaptive Approach*, (Somerset, US: John Wiley & Sons, Incorporated, 2015), 10
- ⁷² Government Accountability Office (1), 10
- ⁷³ John P. Kotter, *Leading Change*, (Boston: Harvard Business Review Press, 2012), 22
- ⁷⁴ Suzanne Miller, vi
- ⁷⁵ Mary Ann Lapham, et al (1), *Agile Methods and Requests for Change (RFC): Observations from the DoD*, Hanscom, MA: Software Engineering Institute: 2014, 9
- ⁷⁶ Ibid, 17
- ⁷⁷ Mary Ann Lapham, et al (2), *Agile Methods: Selected DoD Management and Acquisition Concerns*, Hanscom, MA: Software Engineering Institute: 2011, 15
- ⁷⁸ Miller, 14
- ⁷⁹ Amber R. Oar, et al. "DoD Acquisition Reform: Embracing and Implementing Agile," *Air and Space Journal* 31, no. 1, (2016): pg, 10, www.au.af.mil
- ⁸⁰ Gabrielle Benefield, "Rolling out Agile in a Large Enterprise." *IEEE*, (2008): 5, <http://static1.1.sqspcdn.com/static/f/447037/6486321/1270929190703/YahooAgileRollout.pdf>
- ⁸¹ Ibid, 2
- ⁸² Carlton Northern et al, 4-20.
- ⁸³ Johnson, et al, 5-6
- ⁸⁴ Miller, 43
- ⁸⁵ Scott W. Ambler "Lessons in Agility from Internet-Based Development." *IEEE Software* 19, no. 2 (2002): 66-73.
- ⁸⁶ Miller, vii
- ⁸⁷ Sutherland, 41-70
- ⁸⁸ Government Accountability Office (1), 11
- ⁸⁹ Alan Koch, *Agile Software Development: Evaluating the Methods for Your Organization*, (Norwood, US: Artech House Books, 2004): 10
- ⁹⁰ Ibid, 12
- ⁹¹ Ibid, 12
- ⁹² Miller, 17.
- ⁹³ Mary Ann Lapham, et al (2), 21
- ⁹⁴ Charles G Cobb, 139
- ⁹⁵ Government Accountability Office (1), 12
- ⁹⁶ Koch, 18
- ⁹⁷ Mary Ann Lapham, et al (2), 34
- ⁹⁸ Ibid, 33
- ⁹⁹ Eileen Wrubel and Jon Gross, *Contracting for Agile Software Development in the DoD*, Hanscom, MA: Software Engineering Institute: 2015, 23.
- ¹⁰⁰ Ibid, 24
- ¹⁰¹ Cobb, 102
- ¹⁰² Eileen Wrubel and Jon Gross, 27

Bibliography

- Ambler, Scott. "Agile Adoption Rate Survey." March 2006.
<http://www.ambysoft.com/scottAmbler.html>.
- Ambler, Scott W. "Lessons in Agility from Internet-Based Development." *IEEE Software* 19, no. 2 (2002): 66-73.
- Beck, Kent, Mike Beedle, Arie van Bennekum, Alistair Cockburn, Ward Cunningham, Martin Fowler, James Grenning, Jim Highsmith, Andrew Hunt, Ron Jeffries, Jon Kern, Brian Marick, Robert C Martin, Steve Mellor, Ken Schwaber, Jeff Sutherland, & Dave Thomas. "Manifesto for Agile Software Development." *Agile Alliance*, 2001.
<http://agilemanifesto.org>.
- Benfield, Gabrielle. "Rolling out Agile in a Large Enterprise." *IEEE*, (2008),
<http://static1.1.sqspcdn.com/static/f/447037/6486321/1270929190703/YahooAgileRollout.pdf>.
- Boehm, Barry W. and Richard Turner. *Balancing Agility and Discipline*. Boston MA: Addison-Wesley, 2006.
- Broadus, William. "The Challenges of being Agile in DOD." *Defense AT&L* 42, no. 1 (Jan, 2013): 4-9.
- Brooks, F.P. *Report of the Defense Science Board Task Force on Military Software*. Washington DC: Office of the Secretary of Defense, 1987.
- Chin, Gary. *Agile Project Management: How to Succeed in the Face of Changing Project Requirements*. Saranac Lake, US: AMACOM Books, 2003.
- Clark, Colin. "Army Cyber Chief Meets Buyers In Pursuit Of Faster Acquisition." *Breaking Defense*. October 22, 2013. <http://breakingdefense.com/2013/10/army-cyber-chief-meets-buyers-in-pursuit-of-faster-acquisition/>.
- Cobb, Charles G. *The Project Manager's Guide to Mastering Agile: Principles and Practices for an Adaptive Approach*. Somerset, US: John Wiley & Sons, Incorporated, 2015.
- Cockburn, Alistar. *Agile Software Development: The Cooperative Game*, 2nd ed. Boston, MA: Addison-Wesley, 2007.
- Cooke, Jamie Lynn. *Everything You Want to Know about Agile: How to Get Agile Results in a Less-than-Agile Organization*. Cambridge, GB: IT Governance Publishing, 2012.
- Gilligan, John M. "A Roadmap for Reforming the DoD's Acquisition of Information Technology," *Software Tech News* 15, no. 1 (November 2016),
<https://www.csiac.org/journal-article/a-roadmap-for-reforming-the-dods-acquisition-of-information-technology/>

- Holcombe, Mike. *Running an Agile Software Development Project*. Hoboken, US: John Wiley & Sons, Incorporated, 2008.
- Hughey, Douglas. *The Traditional Waterfall Approach*. 2009.
<http://www.umsl.edu/~hugheyd/is6840/waterfall.html>.
- Johnson, Suzette, Richard Cheng, Stosh Misiazek, Stephanie Greytak, James Boston. "The Business Case for Agile Methods." *Association for Enterprise Information*.
<http://www.afei.org>.
- Koch, Alan. *Agile Software Development: Evaluating the Methods for Your Organization*. Norwood, US: Artech House Books, 2004. <http://replace-me/ebraryid=10082017>.
- Kotter, John P. *Leading Change*. Boston: Harvard Business Review Press, 2012.
- Lapham, Mary Ann, Michael Bandor, Eileen Wrubel (1). *Agile Methods and Requests for Change (RFC): Observations from the DoD*. Hanscom, MA: Software Engineering Institute: 2014, <http://repository.cmu.edu/cgi/viewcontent.cgi?article=1776&context=sei>.
- Lapham, Mary Ann, Suzanne Miller, Lorraine Adams, Nanette Brown, Bart Hackemack, Charles (Bud) Hammons, Linda Levine, Alfred Schenker (2). *Agile Methods: Selected DoD Management and Acquisition Concerns*. Hanscom, MA: Software Engineering Institute: 2011, https://resources.sei.cmu.edu/asset_files/TechnicalNote/2011_004_001_15335.pdf.
- Miller, Suzanne and Dan Ward. *Update 2016: Considerations for Using Agile in DoD Acquisition*. Hanscom, MA: Software Engineering Institute: 2016,
<http://www.sei.cmu.edu/reports/10tn002.pdf>
- Misra, Subhas. "Agile Software Development Practices: Evolution, Principles, and Criticisms." *The International Journal of Quality & Reliability Management* 29, no. 9 (2012): 972-980.
- Moore, Gordon E. "Cramming more components into Integrated Circuits," *Proceedings of the IEEE* 86, no. 1 (January 1998): 1,
<https://www.cs.utexas.edu/~fussell/courses/cs352h/papers/moore.pdf>.
- Morris, Langdon, Moses Ma, and Po-Chi Wu. *Agile Innovation: The Revolutionary Approach to Accelerate Success, Inspire Engagement, and Ignite Creativity*. Somerset, US: John Wiley & Sons, Incorporated, 2014.
- Murray, Anna P. *The Complete Software Project Manager: Mastering Technology from Planning to Launch and Beyond*. Hoboken, NJ: John Wiley & Sons, 2016.
- National Research Council. *Achieving Effective Acquisition of Information Technology in the Department of Defense*. Washington, DC: National Academies Press, 2010.
- Northern, Carlton, Kathleen Mayfield, Robert Benito, and Michelle Casagni. *Handbook for Implementing Agile in Department of Defense Information Technology Acquisition*. Mitre Corporation: December 15, 2010, <http://www.dtic.mil/docs/citations/ADA546756>.

- Oar, Amber R., John P. Fitzsimmons, James P. Guthrie, Raymond A. Hoffman, Andrew J. Metzger, Carl J. Nelson, Taylor J. Olson, and Matthew J. Postupack. "DoD Acquisition Reform: Embracing and Implementing Agile." *Air and Space Journal* 31, no. 1, (2016): <http://www.au.af.mil/au/afri/aspj/article.asp?id=315>.
- Opelt, Andreas, Boris Gloger, and Wolfgang Pfarl. *Agile Contracts: Creating and Managing Successful Projects with Scrum*. Somerset, US: John Wiley & Sons, Incorporated, 2013.
- Overview of the Scaled Agile Framework for Lean Software and Systems Engineering*. Scaled Agile, Inc. Whitepaper. <http://www.scaledagileframework.com/introduction-to-safe/>.
- Palmquist, Steven M., Mary Ann Lampham, Suzanne Miller, Timothy Chick, Ipek Ozkaya. *Parallel Worlds: Agile and Waterfall Differences and Similarities*. Hanscom, MA: Software Engineering Institute: 2016, https://resources.sei.cmu.edu/asset_files/TechnicalNote/2013_004_001_62918.pdf.
- Pellerin, Cheryl. "CIO Priorities Include Cybersecurity, Innovation, Retaining IT Workforce." Accessed: Nov 14, 2016. <http://www.defense.gov/News/Article/Article/702488/cio-priorities-include-cybersecurity-innovation-retaining-it-workforce>.
- Ren, Chiang H., Stephen Busch USAF, and Matthew Prebble. "Improving the Initiation of Acquisition Activities for Automated Information Systems." *Defense AR Journal* 17, no. 4 (2010): 418-435.
- Rico, David F. *What is the ROI of (ROI) of Agile Methods? Analysis of XP, TDD, Pair Programming, and Scrum*, 2008. <https://pdfs.semanticscholar.org/7453/b6083e71a3caf48168f5c3af9df2d09a4dbc.pdf>
- Royce, Winston W. "Managing the Development of Large Software Systems." *Proceedings of the IEEE Westcon, Washington, D.C. 1970* (August, 1970): 1-9.
- Schoeni, Daniel E. "Long on Rhetoric, Short on Results: Agile Methods and Cyber Acquisitions in the Department of Defense." *Santa Clara Computer and High Technology Law Journal* 31, no. 3 (July 1, 2015): 398
- Schwaber, Ken, & Jeff Sutherland. *Software in 30 Days : How Agile Managers Beat the Odds, Delight Their Customers, and Leave Competitors in the Dust*. Hoboken, US: John Wiley & Sons, Incorporated, 2012.
- Sutherland, Jeff. *Scrum: The Art of Doing Twice the Work in Half the Time*. New York: Crown Business, 2014.
- US Congress. House. *National Defense Authorization Act for Fiscal Year 2010*. HR 2647. 111th Cong., Congressional Record 111, no. 288, (October 7, 2009).
- US Department of Defense. *A New Approach for Delivering Information Technology Capabilities in the Department of Defense*. Washington DC: Office of the Secretary of Defense, November 2010. <https://acc.dau.mil/CommunityBrowser.aspx?id=412417>.

- US Department of Defense Science Board. *Final Report of the Defense Science Board Task Force on Department of Defense Policies and Procedures for the Acquisition of Information Technology*. Washington DC: Office of the Undersecretary of Defense for Acquisition, Technology, and Logistics, March 2009.
- US Government Accountability Office (1). *Effective Practices and Federal Challenges in Applying Agile Methods*. Washington DC: Government Accountability Office, 2012. <http://digital.library.unt.edu/ark:/67531/metadc303144/>.
- US Government Accountability Office (2). *OMB and Agencies Need to Improve Planning, Management, and Oversight of Projects Totaling Billions of Dollars*. Washington DC: Government Accountability Office, 2008. <http://www.gao.gov/new.items/d081051t.pdf>.
- VersionOne, Inc. “10th annual state of Agile survey: State of Agile development.” VersionOne, 2016. <http://info.versionone.com/state-of-agile-report-thank-you.html>.
- Wrubel, Eileen and Jon Gross. *Contracting for Agile Software Development in the DoD*. Hanscom, MA: Software Engineering Institute: 2015.