1

Acquisition & Management Concerns for Agile Use in Government Series

# Agile Development and DoD Acquisitions

Software Engineering Institute
Carnegie Mellon University

# Acquisition & Management Concerns for Agile Use in Government

This booklet is part of a series based on material originally published in a 2011 report titled *Agile Methods: Selected DoD Management and Acquisition Concerns* (CMU/SEI-2011-TN-002).

The material has been slightly updated and modified for stand-alone publication.

**Booklet 1:** Agile Development and DoD Acquisitions

**Booklet 2:** Agile Culture in the DoD

**Booklet 3:** Management and Contracting Practices for Agile Programs

**Booklet 4:** Agile Acquisition and Milestone Reviews

**Booklet 5:** Estimating in Agile Acquisition

**Booklet 6:** Adopting Agile in DoD IT Acquisitions

# Agile Development and DoD Acquisitions

## Introduction

The source material for the original report that this booklet comes from included an extensive literature search on the topic of adopting Agile methods within a DoD environment. It was also based on interviews with a number of Agile corporate advocates, practicing Agile consultants, and personnel working on projects employing Agile methods.

In this booklet, we discuss what Agile is and why the DoD is interested in Agile, and we provide background for the report.

## What Is Agile?

Nothing better reflects the culture and values of the Agile community than the Agile Manifesto developed by the Agile Alliance. This alliance was formed in 2001. Members were searching for an alternative to documentation-driven, heavyweight software development processes. In doing so, they expressed their allegiance to a set of values promoting organizational models based on people, collaboration, and the creation of the types of organizational communities they wanted to work in.

Jim Highsmith zeroed in on the importance of values and culture for succeeding with these Agile methods and wrote, tongue-in-cheek: “At the core, I believe Agile Methodologists are really about ‘mushy’ stuff about delivering good products to customers by operating in an environment that does more than talk about ‘people as our most important asset’ but actually ‘acts’ as if people were the most important, and lose the word ‘asset’” [Highsmith 2009]. Therefore, in the final analysis, the meteoric rise of interest in and sometimes tremendous criticism of Agile methodologies is about the mushy stuff of values and culture.

The Manifesto for Agile Software Development (commonly referred to as the Agile Manifesto) states the following:

> 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. [Agile Alliance 2001]

In Agile terms, an Agile team is a self-organizing cross-functional team that delivers working software, based on requirements expressed commonly as *user stories,* within a short timeframe (usually 2–4 weeks). The user stories often belong to a larger defined set of stories that may scope a *release,* often called an epic. The short timeframe is usually called an iteration or, in Scrum-based teams, a *sprint;* multiple iterations make up a *release.* The team's progress toward completion of the iteration is measured via the team's *velocity.* While the code produced within an iteration is useable, it may not have enough functionality to be released to the end user until the multiple iterations that make up a release are completed.

In an environment employing Agile methods, working software is produced at the end of each iteration in an Agile project, and just enough documentation is produced to meet the needs of the team and its stakeholders. Many have speculated that the groundswell of interest in Extreme Programming, Scrum, and other Agile methods, is because the practices largely "define a developer community freed from the baggage of Dilbertesque corporations" [Agile Alliance 2001].

## Why the DoD Is Interested in Agile Methods

Robert Gates, the United States Secretary of Defense, said in a September 2008 speech, "Our conventional modernization programs seek a 99% solution in years. Stability and counterinsurgency missions—the wars we are in—require 75% solutions in months. The challenge is whether in our bureaucracy and in our minds these two different paradigms can be made to coexist" [Gates 2008]. This, and other similar statements by senior DoD officials, express a problem space that is also felt in commercial industry. In the commercial world, the challenge is how to get products to market faster than competitors do, while taking advantage of the latest technologies. In the DoD, the competitor is the adversary, and the consequences of providing competitive capabilities to warfighters too slowly are potential loss of life, not just loss of market share. In addition, one of our reviewers stated that with Agile, one is more likely to get a system that can continue to evolve over time as the customer's needs change. The easier it is to evolve a system, the more likely it is that life cycle costs will be lower, which is important with today's budget pressures.

Gates's concern is reflected in statements by other DoD officials and by Congress itself [OSD 2010]. In December 2010, the Association for Enterprise Information (AFEI) sponsored a one-day forum on the use of Agile methods in the DoD, with a keynote by the Honorable Elizabeth McGrath, Deputy Chief Management Officer of the Performance Improvement Office of the Department of Defense. In her remarks, McGrath noted that the current average time from idea to production release for a DoD information technology (IT) system is 81 months. Her office has coordinated a response to Congress for improved acquisition performance for IT systems that includes recommendations favorable to many of the Agile approaches that we have seen used successfully in DoD programs.[1]

[1] The report to Congress, A New Approach for Delivering Information Technology Capabilities in the Department of Defense, was written pursuant to Section 804 of the National Defense Authorization Act for Fiscal Year 2010.

### The Need for an Acquisition Tempo that Responds to Operational Tempos

These and other statements and activities in the DoD reflect recognition that we must successfully address the difference in the tempo of need (the tempo of the warfighter) and the tempo of provision (the tempo of the developer and the acquirer). A visualization of this challenge is illustrated in Figure 1.

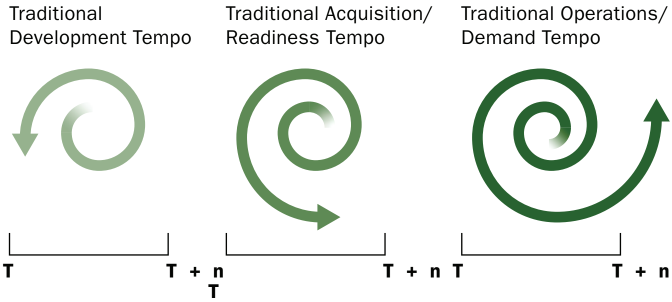
Traditional Development Tempo
Traditional Acquisition/ Readiness Tempo
Traditional Operations/ Demand Tempo
T
T + n
T
T + n
T
T + n

*Figure 1: The Disconnect Among Warfighter and Acquisition Tempos [Boxer 2009]*

Figure 1 shows the different tempos for traditional development, versus traditional acquisition/readiness, versus traditional operations/demand tempo. The "hotter" colors or larger spiral indicate higher tempo. For each, the timeline is the same but the amount that is accomplished varies as represented by the length of the spiral. This graphically depicts the differences in the amount of work that can be traditionally accomplished as opposed to the need or tempo that operations require. To increase the urgency of this problem, the current operations and demand tempo is accelerating to meet today's demands in the field. The DoD needs to get the tempo of work and tempo of operations more in sync. Slowing the operations tempo is not an option for this synchronization.

Addressing the disconnect between the warfighter/demand tempo and the acquisition/contracting tempo is not easy. The acquisition regulations, rules, and practices that have developed over the years to ensure that taxpayer dollars for DoD capabilities are being spent wisely mostly originated in a time when the U.S. was not engaged in such dynamic warfighting situations as today's. This same acquisition governance also reflects a time of building large, complex systems with minimal software reliance. Today, software-reliant systems are the norm instead of the exception.

Agile practices alone cannot solve the tempo issue. However, one of the common practices of Agile—to involve end users early and often throughout the development cycle and allowing them to change the priority of their needs—does address the tempo issue. By acknowledging that requirements are dynamic, not static, and by going directly to the end users who will be employing the provided capabilities, Agile helps to collapse the time lag between identification of a new threat or demand and its satisfaction. Agile also allows incremental software deliveries to the field as opposed to long delivery times associated with releasing all software at once. In Section 2, we will look more closely at Agile principles that affect tempo and their inherent challenges.

## The Need for Rapid Development of Quality Software Systems Within a Dynamic Environment

Another issue that drives the attraction of Agile in DoD contexts stems from the recognition of a need to increase the tempo of acquisition and development while, at the same time, maintaining high-quality software that ensures effective use of resources in providing needed capabilities. There have been many DoD initiatives that attempt to encourage the use of disciplined acquisition and development practices to obtain and maintain high-quality software—CMMI, Lean, and Six Sigma—are all examples that see both effective and ineffective use within the DoD's portfolio of projects.

Operational effectiveness, customer intimacy, and product innovation are the three strategies that market leaders in commercial industry pursue to achieve dominance. These are described in the book, *The Discipline of Market Leaders* [Treacy 1995]. Most methods used to improve high-quality software are focused on improving *operational effectiveness.*[2] Improving operational effectiveness generally focuses on improving the processes that are internal to the enterprise, as opposed to those that are focused on interactions with customers and end users.

Although Agile methods include very defined internal processes, their focus is actually on another dimension pursued by some market leaders—*customer intimacy.* Customer intimacy as a strategy focuses on deep understanding of a set of customer's needs and solution preferences, regardless of how well they fit with the performing organization's preferences. Operational effectiveness as a strategy focuses on optimizing the processes that produce the performing organization's products and services, with less regard for the deep understanding of customers. Gates's statement about needing a 75% solution in months reflects an acknowledgment that acquirers and developers who are not active in the operational space cannot be expected to provide complete solutions—the operational environment is not sufficiently static to support pre-definition of all the requirements. The Agile focus on direct involvement of end users throughout the development process is a direct reflection of this difference in strategy. At the AFEI DoD Agile Development Conference,[3] one of the recurring themes was how important the continual inclusion of end users was in successful projects using Agile. One of the authors has observed that outside the DoD, and even outside the U.S., organizations are finding that the use of Agile methods combined with other methods like CMMI is a powerful approach to achieving both customer intimacy and operational effectiveness.

[2] In the context of Treacy's book, operational refers to the fundamental processes that produce the work products and services of an organization. Their context goes beyond the military viewpoint of operations to include acquisition and development operations.

[3] NDIA/AFEI. Program, DoD Agile Development Conference. NDIA/AFEI, December 14, 2010, Alexandria, VA.

When organizations like the Software Engineering Institute (SEI) started addressing process discipline issues in order to obtain high-quality software in the 1980s, we often expressed a triangle made up of process, people, and technology, illustrated in Figure 2.

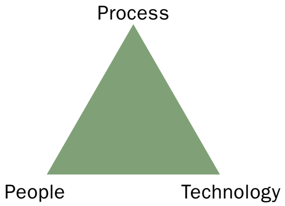

*Figure 2: 1980s View of Process Discipline*

As understanding of the role of process in supporting the key factors of market leaders—operational effectiveness, product innovation, and customer intimacy—evolved, a more accurate portrayal of the role of process discipline has evolved, as illustrated in Figure 3.

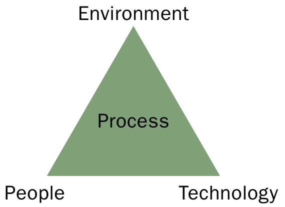

*Figure 3: Process Triangle Including Environment [Garcia 2006]*

This view of process sees process as an integrating function between technology, people, and their environment. When people and their skills change, the processes need to change; when technology changes, processes usually need to change too. And when the environment—the operational environment, the business or market environment—changes, then processes need to adapt to the new conditions. Incorporating the environment dimension as an explicit aspect to be accounted for in designing and adapting processes is consistent with the Agile view of the operational environment and its dynamism being the source of processes that are meant for adaptation.[4] Achieving high quality in the Agile context requires discipline in the process areas we are accustomed to focusing on, such as operational effectiveness, as well as a new focus on processes for customer intimacy.

[4] Watts Humphrey, one of the great proponents of process discipline and a consistent user of the original process triangle, commented in 2006 that this revised view of the influences on process solves some of the problems that he had experienced in communicating the benefits of disciplined processes.

## Achieving More Value with Limited or Shrinking Resources

Historically, the project triangle, also known as the Iron Triangle, is a depiction of the three project attributes or constraints that must be balanced to achieve a successful project outcome: cost, schedule, and scope.[5] As shown in Figure 4, each attribute is shown on the corners of the triangle, implying that how the three attributes are balanced will determine the "shape" of the project's focus. If one attribute is changed, the other two attributes will also be affected. For example, increased scope typically means increased time and increased cost; a tight time constraint could mean increased costs and reduced scope, and a tight budget could mean increased time and reduced scope. Sometimes a fourth attribute, quality, is included and shown in the center of the triangle as it is the ultimate result of the three other attributes. Typically, projects use these three measures (scope, cost, and schedule) to determine the success of the project. Completion within cost and schedule and providing all the scope is the definition of a successful project.

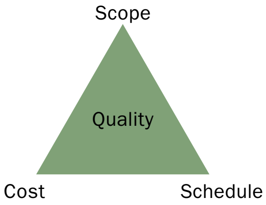

*Figure 4: Classic Iron Triangle*

However, software development projects often fail because the organization sets unrealistic goals for the Iron Triangle. An example of this came from one of our reviewers. If the government got a requirement to take a simple Hypertext PreProcessor (PHP)/mySQL-based forum type website that already exists in the .com and simply move it to the .mil, it could take $3-5 million and a year to complete. This would include, but not be limited to, documenting a new start, conducting a capabilities assessment, assigning a program manager, finding a host, doing the justification and approval, establishing contracts, getting the vendor and "approved" system for billing, briefing the required oversight groups, and so forth. If this type of requirement occurred within a commercial environment, it would take about two hours and less than $1,000.

"The fact that (particularly SIDRE [software-intensive innovative development and reengineering/evolution]) software development effort and duration cannot be estimated precisely means that it is unwise to try to lock a software project into simultaneously fixed budget, schedule, and feature content (as has been found in many fixed-price, fixed-requirements software development contracts)" [Critical Code 2010]. In the end, if the project team delivers at all, the quality of the delivered product suffers and the project is almost always late and over budget.

[5] The triangle is the historical representation of this idea as well as for process. The two triangles do not represent the same ideas but rather only use the same icon.

With the emergence of Agile, another view of the Iron Triangle has evolved. Jim Highsmith proposed the Agile Triangle as an alternative to measuring performance with the Iron Triangle because Agile is all about being flexible [Highsmith 2009]. Since value is based on capabilities that the users or stakeholders find valuable, scope is the cornerstone of the Agile Triangle. Scope should be considered first and cost and schedule should adapt to achieving the scope. This may or may not be possible, but it is the ideal. Because Agile processes and methods allow for flexibility, customers also gain more innovation value in that it is easier for them to be inventive or consider new ideas.

Highsmith has continued to evolve the initial Agile Triangle. The most important items to measure should be value and quality, within the constraints of the program (scope, cost, schedule). According to Highsmith, these are defined as

1. Value: Your project's value should be measured by the stakeholders and what they expect.

2. Quality: The quality part of the triangle means you can deliver a reliable product by adapting to the customer's needs.

3. Constraints: Here is where the three elements of the Iron Triangle appear—project scope, schedule, and cost.

The new Agile Triangle shown in Figure 5 illustrates these attributes.

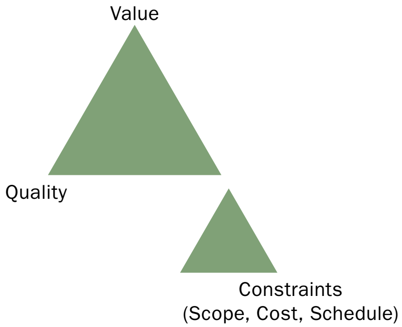

*Figure 5: New Agile Triangle, adapted from Jim Highsmith (http://www.jimhighsmith.com/2010/11/14/beyond-scope-schedule-and-cost-the-agile-triangle/).*

The new Agile Triangle changes the foundational trade-off elements to include value and quality, and keeps the old standards of cost, schedule, and scope in the constraints part of the triangle. This is another way in which Agile addresses Gates's need for a "75% solution in months." By putting the focus on value to end users through such approaches as continual end-user involvement, Agile's philosophy is poised to address explicit DoD needs.

# Conclusion

The Agile Manifesto proposed "better ways of developing software," by focusing on the so-called mushy stuff like delivering good products to customers. Over the years, a number of specific implementations of Agile methods have emerged, such as Extreme Programming and Scrum. The DoD is adopting some of these methods as a way to make defense acquisitions more effective and to align the acquisition tempo with the department's operational tempo.

Agile practices alone cannot solve the tempo issue, but taking steps to involve users early and often throughout the development cycle and allowing them to change the priority of their needs can go a long way towards improving outcomes. Thus, it is important for members of the defense acquisition community to familiarize themselves with Agile, to add it to their toolbox for current and future programs.

# References

[Agile Alliance 2001]
Agile Alliance. *History: The Agile Manifesto. http://agilemanifesto.org/history.html (2001).*

[Boxer 2009]
Boxer, P. & Garcia, S. *Limits to the Use of the Zachman Framework in Developing and Evolving Architectures for Complex Systems of Systems.* SATURN Conference, May 2009. Software Engineering Institute, 2009.
http://www.sei.cmu.edu/saturn/2009/images/Limit_use_Zachman_Framework.pdf

[Critical Code 2010]
Committee for Advancing Software-Intensive Systems Producibility, National Research Council. *Critical Code: Software Producibility for Defense.* The National Academies Press, 2010

[Garcia 2006]
Garcia, S. & Turner, R. *CMMI Survival Guide: Just Enough Process Improvement.* Addison-Wesley, 2006.

[Gates 2008]
Gates, R. M. Speech to National Defense University (Washington, D.C.) Monday, September 29, 2008. http://www.defense.gov/Speeches/Speech.aspx?SpeechID=1279 (2008). Accessed July 13, 2011.

[Highsmith 2009]
Highsmith, J. *Agile Project Management: Creating Innovative Products,* 2nd ed. Addison-Wesley, 2009.

[OSD 2010]
Office of the Secretary of Defense. *A New Approach for Delivering Information Technology Capabilities in the Department of Defense, Report to Congress,* November 2010, Pursuant to Section 804 of the National Defense Authorization Act for Fiscal Year 2010. United States Department of Defense, 2010. http://dcmo.defense.gov/documents/OSD%2013744-10%20-%20804%20Report%20to%20Congress%20.pdf

[Treacy 1995]
Treacy, M. & Wiersema, F. *The Discipline of Market Leaders: Choose Your Customers, Narrow Your Focus, Dominate Your Market.* Perseus Books, 1995.

This material is based upon work funded and supported by the Department of Defense under Contract No. FA8702-15-D-0002 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center.

The view, opinions, and/or findings contained in this material are those of the author(s) and should not be construed as an official Government position, policy, or decision, unless designated by other documentation.

References herein to any specific commercial product, process, or service by trade name, trade mark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by Carnegie Mellon University or its Software Engineering Institute.

NO WARRANTY. THIS CARNEGIE MELLON UNIVERSITY AND SOFTWARE ENGINEERING INSTITUTE MATERIAL IS FURNISHED ON AN “AS-IS” BASIS. CARNEGIE MELLON UNIVERSITY MAKES NO WARRANTIES OF ANY KIND, EITHER EXPRESSED OR IMPLIED, AS TO ANY MATTER INCLUDING, BUT NOT LIMITED TO, WARRANTY OF FITNESS FOR PURPOSE OR MERCHANTABILITY, EXCLUSIVITY, OR RESULTS OBTAINED FROM USE OF THE MATERIAL. CARNEGIE MELLON UNIVERSITY DOES NOT MAKE ANY WARRANTY OF ANY KIND WITH RESPECT TO FREEDOM FROM PATENT, TRADEMARK, OR COPYRIGHT INFRINGEMENT.

DM17-0009

## About the SEI

For more than three decades, the Software Engineering Institute (SEI) has been helping government and industry organizations acquire, develop, operate, and sustain software systems that are innovative, affordable, enduring, and trustworthy. We serve the nation as a federally funded research and development center (FFRDC) sponsored by the U.S. Department of Defense (DoD) and based at Carnegie Mellon University, a global research university annually rated among the best for its programs in computer science and engineering.

## Contact Us

Software Engineering Institute
4500 Fifth Avenue, Pittsburgh, PA 15213-2612

Phone: 412.268.5800 | 888.201.4479
Web: www.sei.cmu.edu | www.cert.org
Email: info@sei.cmu.edu