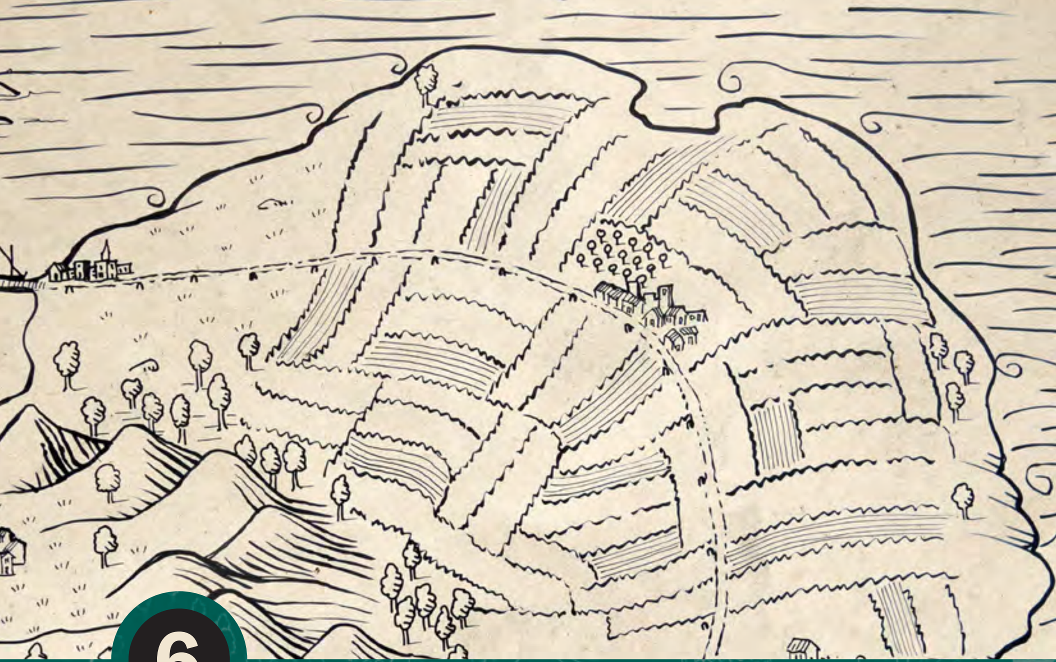

6

Acquisition & Management Concerns for Agile Use in Government Series

Adopting Agile in DoD IT 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

Adopting Agile in DoD IT Acquisitions

Why Change to Agile Methods?

Changing the practices of an organization is rarely easy. The further the new practices and their methods are from current practice, the more difficult, time-consuming, and resource-consuming the change is.

In a previous booklet, we pointed out several of the reasons that acquisition and development organizations are adopting Agile methods. Our interviewees exhibited a wide variety of reasons for considering Agile methods. The ones who experienced the greatest success were those for whom the change was important in a particular, rather than a general way. Where the motivation for change was less particular and more general, the changes did not stick as well, nor were the expected benefits realized. The two biggest reasons we have seen within DoD for moving to Agile are

1. *A burning platform:* If the program does not change its current development practice to improve outcomes, it is likely to be cancelled.
2. *Urgency of delivery:* An operational need that cannot wait for traditional delivery times is mission-critical enough to warrant a different acquisition approach.

In the programs that we interviewed, there were other motivations for beginning the change to Agile practices. For the organizations that have been using Agile methods for some time, we heard some common themes that characterized their continuing motivation for change, including

- a sense of true accomplishment when they delivered a release that they knew incorporated functionality the end user needed
- a short time span for seeing the differences their work made to their end users
- encouraging (often laudatory) user feedback that clearly communicated the value of their approach
- consistent ability to meet or exceed user expectations
- previous inability to deliver value within agreed time spans and costs

The above themes were seen among both developers and acquirers. However, all the organizations agreed that their culture and practices needed to change if they wanted the benefits they were seeking from adopting Agile methods. The next section discusses a way to gauge the size of the needed adjustment.

How Big a Challenge is Your Adoption of Agile Practices?

Paul Adler, a well-known researcher in the field of organizational behavior, has postulated a continuum of change that predicts how difficult it will be for an organization to change its practices. In its simplest version, it looks at factors that indicate increasing level of difficulty, and correlated with that, time to change. The least impacting change tends to be a skills change, followed by procedural change, structural change, strategy change, and the most difficult and time-consuming, culture change.[1]

It is clear in our interview population and other experience that different organizations adopt Agile practices to different levels in terms of Adler's progression. All projects we interviewed at least had skill and procedure changes. Many of them incorporated structure changes due to different kinds of interactions with users than in the past. Some also changed their business strategy, and several of them changed their culture to be more amenable to Agile practices. In the book *Becoming Agile,* the authors differentiate "doing agile" versus "being agile" [Sidky 2009]. From Adler's viewpoint, they differentiate "being Agile" as involving the culture change to adopting the values of the Agile manifesto published in 2001 [Adler 1990]. We discussed Agile cultural norms in *Booklet 2: Agile Culture in the DoD.* In that booklet, we introduced some of the differences between Agile culture and more traditional development cultures. We will address that knowledge, as well as other elements that affect the pace and success of change, in this booklet.

Adoption Assumptions Table for Agile Methods

As part of a readiness and fit assessment (RFA) for adopting Agile methods, a series of adoption factors that include those from Adler have been articulated. RFA can help you to understand how far away your own organization's practices, values, and culture are from what is typical in an organization that has embraced Agile practices and culture [Garcia 2006].

Cultural expectations related to an environment that has successfully adopted Agile methods are divided into two columns. The first lists expectations of what would be seen in a development organization adopting Agile methods. The second lists expectations of what would be seen by customers (or in the DoD case, acquirers) of a solution being built using Agile methods. This is a starting point list based on our observations of Agile projects in the DoD context. The kinds of behaviors and attributes that are listed in Table 1 are ones that we have seen in at least some of the DoD programs that were adopting Agile methods. By providing both developer and acquisition personnel expectations, we hope to help acquisition program offices understand the cues they are getting from their development contractors.

As more DoD projects adopt Agile practices, we expect Table 1 to evolve, since particular instantiations of these factors depend on both the product and environmental context.

[1] See CMU/SEI-2011-TN-002, Appendix G, for elaboration of these factors and their effect on time and complexity of change efforts.

Table 1: Adoption Expectations for Agile Culture, Practice, and Skill for Developers and Acquirers

ADOPTION FACTOR	AGILE CULTURE EXPECTATIONS—DEVELOPER	AGILE CULTURE EXPECTATIONS—CUSTOMER/ ACQUIRER
Business Strategy (What types of business strategy are typical in this environment?)	customer-centric vs. technology-centric business strategy value to customer is paramount	incremental delivery is supported ongoing solution vs. single point product is needed/ supported
Reward System (What are the types of behaviors that are rewarded or punished?)	team performance is emphasized more heavily than individual mentoring is rewarded candor/truth is rewarded, not punished solving problems that are not directly tied to your iteration goals is NOT rewarded heroics are not encouraged	early functioning software release is rewarded vs. documents being "finished"
Sponsorship (What level and types of sponsorship behaviors are needed for adoption to succeed?)	requires local engineering management sponsorship and senior management buy-in that is sufficient to allow changed practices vision for change is shared by sponsor and teams engineering and senior management are willing to adopt "time box" approaches to managing	willingness to permit access to end-user subject matter experts throughout the development life cycle willingness to be an active partner in the co-creation of the software solution

▸Continued

Table 1, continued

Values (What cultural norms are exhibited in the activities and decisions of the organization?)	keep everything—software, documentation, overhead—as simple as possible sharing of information vs. “knowledge is personal power” learning is valued fail fast, learn fast is encouraged self-reflection via activities like retrospectives are encouraged strong team-based values shared team understanding of what “done” means developer isolation is frowned upon	metrics used for course correction, not punishment of individuals “gold-plating” is discouraged bloated software is discouraged minimizing waste is important
Skills (What skills and knowledge are prevalent in the individuals in the organization?)	working well with peers (especially for pair programming) translating user scenarios (stories) into test cases working directly with end users or surrogates different roles working together across boundaries (i.e., testers and developers working together from the beginning of the cycle) strong verbal communication skills ability to move on when product is sufficiently “done”	strong verbal communication skills facilitative leadership skills conflict resolution skills when dealing with diverse stakeholders ability to recognize when product is sufficiently “done” to move on working effectively with end users or surrogates
Structure (What kinds of structural mechanisms are used that contribute to the organization’s success?)	colocation; use of collaboration technology to compensate when co-location is not feasible team roles are flexible layering of teams when projects get too large for a single team; sometimes this becomes a hierarchy, sometimes it is a set of peer teams	porous boundaries between acquisition and development staff support for a strong user (surrogate) presence throughout development iterations and at release

History (How necessary is successful change history to the adoption of this set of practices?)	history of successful management changes is helpful in getting teams sufficient time to demonstrate new approach to senior management “burning platform” motivation is often useful in allowing teams to try something different from past practice	understanding that “silver bullets” don’t work is useful
Work Practices (What work practices distinguish this technology or methodology from others?) Note: Not all of these work practices are present in all variations of Agile methods.	test-driven development user scenarios creation pair programming daily builds; continuous integration strong version control project retrospectives iteration-based release management daily standup meetings collaborative planning among team, customers, and management use of velocity as a primary measure of team progress	active participation in construction and evolution of a product backlog active participation in progress reviews and demonstrations active participation in construction and evolution of user stories collaborative planning among team, customers, and management construction of contracting vehicles that support practices (see above) and deliverables typical of Agile methods

In earlier sections of this document, we explored some of the above attributes as well as others that play a role in adopting Agile within a DoD context. As more development organizations adopt Agile practices (some of the literature argues that Agile development has already reached a tipping point in the software industry and is becoming the dominant approach), more DoD acquisition organizations are likely to try to benefit from Agile [Leffingwell 2008]. The report to Congress responding to the 804 section requirements of FY 2010, although it does not explicitly call out Agile, embraces many of the tenets of Agile [OSD 2010]. The further organizations are from the attributes in the adoption expectations table above, the more difficult the transition to Agile practices will be. The 804 Implementation Task Force has started studying the DoD and service-level acquisition regulations to find or establish better support for responsive, iterative acquisition practices. Although the task force has found some supportive rhetoric, it also has found regulations that are more easily *interpreted* to prohibit some of the Agile attributes listed above; however, these interpretations are not codified into regulations.[2] For IT systems at least, changes in regulations that make it easier to adopt Agile methods are being contemplated, according to the 804 report [OSD 2010].

[2] Boston, J., et al. EX INVEST proposal to Office of the Secretary of Defense. 2010.

Approaches for Successfully Adopting Agile Practices

In interviewing projects that have adopted Agile, we consistently asked, “Did your approach to adopting Agile methods have to be different from your approach to adopting other new practices (e.g., new testing methodologies)?” The consistent answer to this question was, “No, the same things you have to pay attention to for any adoption must be paid attention to for adopting Agile practices.” In this section, although we will address some more general organizational change management approaches, we will include, where we have seen it, how these approaches were expressed in our interviewed projects.

There were a variety of approaches that the programs we interviewed took for adopting Agile practices. Most exhibited awareness of cultural issues, adopter populations, and finding or building the right adoption support mechanisms for their environment, even though they might not have expressed their journey in exactly those terms.

The change management topics we will focus on below are typical of the things that must be dealt with in adopting Agile practices:

- understanding your adopter population
- understanding the cycle of change
- understanding your adoption risks
- building transition mechanisms to mitigate adoption risks

Understanding Your Adopter Population

When adopting new practices, it is crucial that you understand the characteristics of the people—both as individuals and groups—you are trying to influence to adopt. Geoffrey Moore, a marketing researcher and business consultant, leveraged prior research that indicated that groups are differentiated in terms of their propensity to adopt a new technology, and can be characterized in terms of the types of communication and implementation mechanisms they need to feel comfortable with the technology [Rogers 2003]. He noted that, in conjunction with the adopter groups that had been previously defined, a “chasm” appeared to exist between two of the groups, from the viewpoint that getting a technology adopted by one group was more than incrementally different from the next [Moore 2002]. Figure 1 shows Moore’s version of the adopter populations curve.

The chasm appears between the Early Adopters and Early Majority groups. These two groups are very different in terms of the kinds of communication and implementation mechanisms that will influence them to adopt the new practices or technology. We observed most of the current DoD adopters of Agile to be Early Adopters, although when some programs began their adoption, they were even farther to the left. (For example, some of them were willing to put together their own tools for managing their product backlog and to measure and monitor their velocity.) Where we saw Early Adopter attitudes in the development staff and the acquisition staff, Agile methods had sufficient support to achieve successful use.

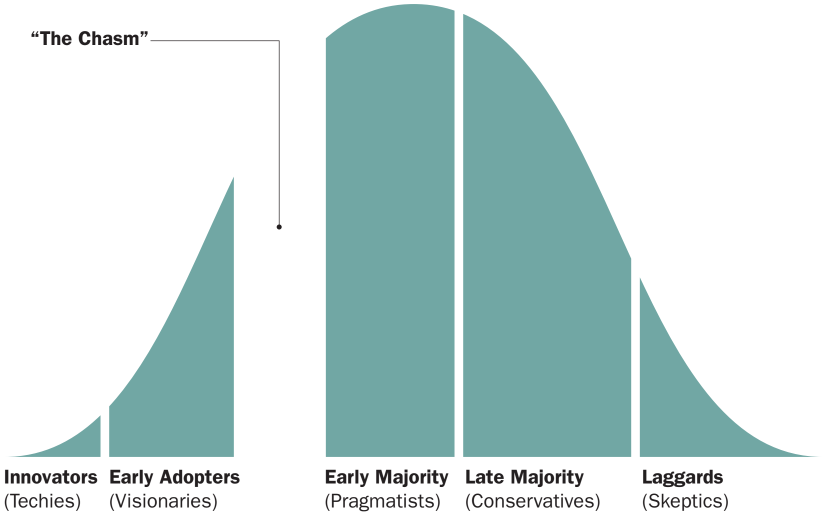

Figure 1: Geoffrey Moore's Adaptation of Rogers' Adoption Populations Curve[3]

The Early Majority adopters, on the other hand, are looking for a more explicitly proven benefit and prefer technologies or practices that are supported by training and tools. However, they are still willing to tailor practices to their own environment or domain.

Where we have seen the most conflict in DoD programs in transition toward Agile methods is one where Early or Late Majority acquisition staff had been rotated into a program that had been previously staffed with Early Adopter acquisition and development staff using Agile methods. In particular, plans that had been made for iterative mini-PDRs were reversed by the acquisition staff to a more traditional "one big PDR" event. As some of the recommendations from the 804 Report to Congress are implemented, we hope that more communication and implementation support mechanisms will be made available to Early and Late Majority acquisition staff.

What this meant to our interview population is that they have been instrumental in terms of finding ways to "work Agile" in an environment that demands artifacts and evidence based on *working traditional.* In other sections, particularly the Technical Milestones section of this report, we have elaborated some of the ways in which Early Adopters have made Agile methods successful in the DoD acquisition context. However, successful adoption across a wide spectrum of appropriate DoD programs will not occur until more communication and implementation support mechanisms are available to the Early Majority adopters (e.g., DAU courses on how to implement Agile methods in an acquisition program, or other similar guidance).

[3] The y-axis, not shown in these figures, represents the percentage of people typically in each group.

Late Majority (or Conservative) adopters will wait until it is inconvenient to not adopt a new set of practices. The Agile methods proponents are still several years away from penetrating the Late Majority software development community. Evidence of this shift will be when there are Agile variants specific to different domain sectors (like banking) and when Agile methods are the dominant approach being taught at the undergraduate level in software engineering.

All of the programs we interviewed exhibited either Innovator or Early Adopter characteristics in their leadership teams, where adoption style will make a huge difference. We are starting to encounter programs that have more traditional middle management even though the senior leadership is interested in adopting Agile methods. Where middle management exhibits more early- or late-majority characteristics, versions of Agile methods with more *packaging* will have to be used to help them make the transition. For example, an appendix to 5000.02 or Air Force Instruction (AFI) acquisition life cycle regulations that provides explicit guidance on contents and sequencing of technical milestones in a program using Agile would be a part of such packaging, along with tailoring guidance for known variations of program types. Support in terms of policies that visibly drive toward Agile methods is another example of more packaging for Agile approaches.

Understanding the Cycle of Change

A precept of organizational change is that any change—positive or negative—takes effort and time to incorporate into an individual's mental models and into a group's behavioral norms. Even for changes that we perceive as positive, there is a series of steps that we invariably go through (usually faster for positive changes) before achieving the results that are desired from a change in behavior.

Virginia Satir's elaboration of this change cycle has been adopted by well-known consultants and change practitioners in the software industry and is cited explicitly in some Agile literature [Weinberg 1997].

Figure 2 illustrates the basic phases of Satir's change cycle.

The Old Status Quo is disrupted by the introduction of a *foreign element*—a change that will require different skills, procedures, etc., and will presumably improve overall performance. This precipitates a stage of highly variable performance, as the individual or group tries to accommodate the change within its current frame of reference (for a positively perceived change) or tries to reject it (for a negatively perceived change). Chaos continues until the individual or group discovers a *transforming idea.* The transforming idea is something that allows the individual to perceive some intended benefit from the change and incorporates some idea of how to connect the change to his or her own behavior and values.

It is easy to think that, at this point, the New Status Quo, complete with improved performance, will be reached. However, a stage of integrating and practicing the new behavior, and fine-tuning it to meet the individual's context, will be needed before the New Status Quo, with its associated benefit in performance, can actually be reached.

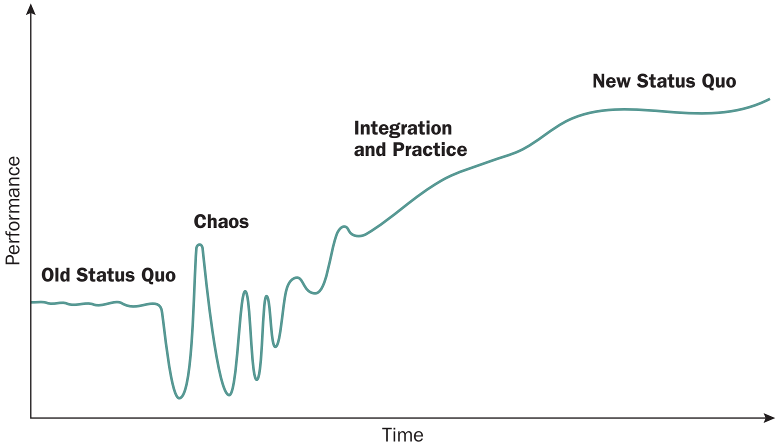

Figure 2: Satir Change Model

As a whole, DoD's adoption of Agile is centered in the Old Status Quo and Chaos sections of the graph, with a few programs that are in Integration and Practice, and a handful at best where Agile is the New Status Quo. In our interviews, we saw evidence of this cycle. It was common for organizations to phase in their adoption of Agile methods over at least a couple of years, allowing staff to get accustomed to a new set of practices (e.g., pair programming) before adding in another set (e.g., continuous integration). When this kind of phasing is done, while conscious of where the individuals generally are in relation to the Satir cycle, the Chaos stage can be minimized. In most organizations, the optional point to add in a new practice is shortly after a New Status Quo has been achieved. At that point, the new behaviors are fairly embedded, but have not become so ingrained that they are difficult to modify if needed. However, piecemeal adoption of Agile methods, if not paced wisely, has also resulted in lack of full adoption and lack of benefit realized.

In some cases, we saw an organization that was well on its way into Integration and Practice but was thrown back into Chaos when a new foreign element was introduced. In particular, backsliding was seen when they were asked to continue achieving the success they were seeing with Agile practices while maintaining the familiar technical milestone events of a single PDR and single CDR. From a Satir viewpoint, the solution to this dilemma is to find a transforming idea that allows both approaches (Agile and traditional PDRs) to coexist. If there is insufficient trust between the different parties advocating for the conflicting practices, finding that transforming idea could easily fail. We observed that failure firsthand in one of the programs with which we were involved.

Understanding Your Adoption Risks

Once you have some ideas about your motivation for change, the scope of the change, and the population you want to adopt the change, you can get more specific about identifying the particular adoption risks you are likely to face when adopting Agile development and acquisition approaches. One effective way of identifying adoption risks is to use Table 1 as a basis for analyzing how closely your current organization exhibits the kinds of attributes that are typically seen in Agile environments. This method has been successfully used in other areas, including CMMI adoption and complex commercial off-the-shelf software adoption (like enterprise resource planning (ERP) software) [Garcia 2006]. We briefly summarize below an organizational evaluation method called Readiness & Fit Analysis. For additional detail on applying Readiness & Fit Analysis (RFA), refer to Chapter 12 in *CMMI Survival Guide: Just Enough Process Improvement.*

By surveying or interviewing potential Agile team members, you can get an estimate of how close the organization's attributes are to the Agile attributes in Table 1. Usually a 1-to-5 scale is used. The results can be aggregated by group and reported via a histogram or kiviat (radar chart) diagram to get an overall perception of fit. Beyond the summary visualization, it is extremely valuable to ask participants, via either a survey or workshop or interview, for specific issues they see in a particular topic area that would impede adoption or, conversely, specific strengths they see that would improve the chances of successful adoption. These opportunities for leverage, issues, and risks (participants will naturally identify some of each) can be grouped into similar themes (which may or may not reflect the categories in the expectations table—often they do not), and mitigation planning can be done to address the issues.

There are dozens of adoption support mechanisms that can be acquired or developed to help an organization through an adoption of new practices. The mechanism used across the majority of our interviewed programs was external training and consulting. Sometimes this was conducted by the iconic agilists in the field (one group hired Ken Schwaber, a well-known and prolific agilist, as its consultant/trainer). Agile consultants and trainers usually addressed four particular areas of the assumptions table for Agile methods: practices, skills, sponsorship, and values. Other transition mechanisms used included facility changes (substitution of team rooms for individual cubicles), changes to program-level management directives, formation of Agile methods communities of interest, and lunch-and-learn activities where individuals from the team chose a topic about which to become more knowledgeable.

The most obvious of the four areas focused on by Agile consultants is practices. Training is a standard way of transferring knowledge about what practices are relevant to a particular situation and how to execute them. Successful use of practices depends, of course, on appropriate skills. The skills needed to adopt Agile methods, in terms of individual techniques, can be acquired through basic training. However, a more complete training series that includes education on Agile culture and values provides a basis for helping team members on an Agile program make decisions that are consistent with the Agile culture even if the situation was not exactly the same as those covered in the class. At least one of our interviewed organizations took this extra step with its training approach.

Another program used team coaching from the consultant during a real project as a way to help the team recognize, transition, and apply the values of an Agile culture. Both approaches worked for the groups involved.

The other role a good Agile consultant played in some of the programs we interviewed was as coach to the sponsors of the Agile methods adoption. Early-adopter sponsors often understand the broad concepts and goals of a set of methods like Agile methods, but may not be as certain as to how their typical behaviors will enable or create barriers for adoption. A good Agile consultant can be very useful in pointing out alternative behaviors for the sponsors that will be more productive in an Agile setting. At least one of the programs successfully used its consultant in that role.

Building Transition Mechanisms to Mitigate Adoption Risks

We introduced the concept of communication and implementation support mechanisms earlier. They are both examples of the more general category of transition mechanisms. Deciding which mechanisms must be built, and at what point in an adoption of new practices, is a common issue, regardless of the technology or practices being adopted. The timing of training, in particular, can have an effect on adoption pace. Training delivered too early cannot be acted on before the skills dissipate. Training delivered too late rarely is successful in refocusing negative impressions of the practices, especially ones that require new skills.

A model we have found useful for almost two decades in figuring out what adoption support mechanisms to build when is based on the adoption commitment curve, illustrated in Figure 3.

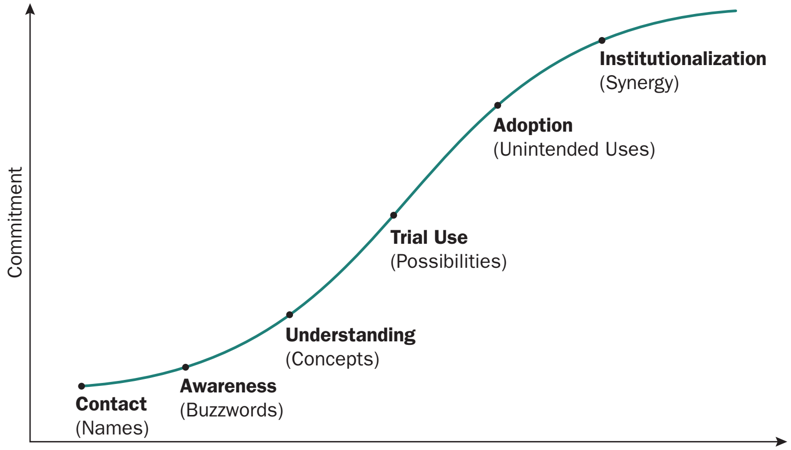

Figure 3: Adoption Commitment Curve [Conner 1983]

This curve expresses, in general, the difference in commitment that is needed by an individual or group to adopt a new technology or set of practices. The early, “easy” stages—contact, awareness, and understanding—primarily rely on communication mechanisms to help people move forward. The latter, “hard” stages—trial use, limited adoption, and institutionalization—rely on implementation support mechanisms that tend to be more specialized to the technology and the organization than the types of communication mechanisms that are used earlier in the adoption.

For Agile methods, some candidate transition mechanisms that might be useful at each stage of adoption in the DoD are proposed in Table 2. These are based on our team’s interviews with DoD programs and our knowledge of Agile patterns of adoption, and have not been reviewed for their ease or difficulty to implement in DoD or service regulations and directives. They are suggestive rather than exhaustive or complete.

Table 2: Candidate Transition Mechanisms for DoD Adoption of Agile Methods Keyed to Adoption Commitment Curve Stages

CONTACT/ AWARENESS	Articles in Crosstalk, Defense Acquisition News, and other publications on programs successfully using Agile methods	Conference tracks and workshops that highlight the benefits and risks associated with adopting Agile practices
UNDERSTANDING	Course modules in DAU acquisition courses discussing when and how to use Agile methods in DoD acquisition	Conferences and workshops that focus on implementing Agile concepts in a DoD acquisition environment
TRIAL USE	Standard language for Milestone Decision Authorities to use to except programs using Agile methods from waterfall practices that conflict with Agile methods	Provisions for programs attempting Agile methods for the first time to get continual direct access to their end-user communities
ADOPTION	Standard reporting templates for oversight reports expected in an Agile acquisition	Guidance on use of story points and other Agile estimation mechanisms in GAO Estimation Best Practices Guide
INSTITUTIONALIZATION	Language changes to DoD 5000.02 and service regulations that make clear when to use Agile methods and provide guidance on making Agile methods effective within an acquisition	Citation of the successful use of Agile methods in programs that are held up as models for DoD acquisition success (obviously only if the use of Agile methods is true!)

The adoption commitment curve helps to explain why training can sometimes be too early in an adoption cycle. Training is primarily used as an awareness-building mechanism (one-day Agile methods overview, for example), or as an understanding-building mechanism (five-day Agile methods course where students exercise all the skills associated with that particular method on case study projects or their own). When skills-focused training is used as an initial event, it is usually overkill—the potential adopters have not had sufficient exposure, contact, and awareness to the practices or technology to "make it their own." Most of the programs we interviewed used external training, primarily of the skill-building type, to jumpstart their Agile teams. Some of them used general training as a way to build awareness of the coming change with both managers and development staff. Others used conferences, readings, and guest speakers at meetings as a way to build awareness of Agile methods before embarking on skill-building training.

Coupling the adoption risks list generated via RFA with the adoption commitment curve is a powerful way of planning "just in time" adoption support mechanisms for your Agile adoption. As we gain more insight into adoption of Agile methods across a broader range of DoD programs, it may be possible to better characterize typical mechanisms that have been successful in different acquisition settings.

Conclusion

Although it is easy to say that Agile methods do not require any different methods for addressing organizational change, it is not so easy to actually apply those methods in demanding project settings, just as it is not easy to do so for other practices or technologies. Someday there may be a “GAO Best Practices Guide for Adopting Agile Methods,” similar to the GAO Guide for Estimation. In the meantime, we leave you with the following advice:

- Find and nurture good sponsors for your adoption.
- Understand your adoption population.
- Conduct some kind of readiness assessment that addresses organizational and cultural issues.
- Analyze what adoption support mechanisms you are likely to need for your context and build or acquire them before you get too far into your adoption.

For a starter list of resources for Agile adoption, several of which are focused on organizational change management, consult Appendix 1 of the 2011 Tech Note from which this booklet is drawn, *Agile Methods: Selected DoD Management and Acquisition Concerns* (CMU/SEI-2011-TN-002).

References

[Adler 1990]
Adler, P. S. & Shenhar, A. “Adapting Your Technological Base: The Organizational Challenge.” *Sloan Management Review 32,* 1 (1990): 25-37.

[Conner 1983]
Conner, D. & Patterson, R. “Building Commitment to Organizational Change.” *Training and Development Journal* (April 1983):18-30.

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

[Leffingwell 2008]
Leffingwell, D. *Scaling Software Agility.* Addison-Wesley, 2008.

[Moore 2002]
Moore, G. *Crossing the Chasm: Marketing and Selling Disruptive Products to Mainstream Customers.* Harper Business Essentials, 2002.

[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

[Rogers 2003]
Rogers, E. *Diffusion of Innovation,* 5th ed. Simon & Schuster, 2003.

[Sidky 2009]
Sidky, A. & Smith, G. *Becoming Agile in an Imperfect World.* Manning Publications Co., 2009.

[Weinberg 1997]
Weinberg, G. *Quality Software Management: Anticipating Change.* Dorset House, 1997.

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