

AGLPM5 – Unit 1 - ACTIVITY 2: OBSERVE

Best Practices and Agile Adoption Barriers

This document is an excerpt from the book:
“Agile Project Management for Government “
Authored by Brian Werham
Published by Maitland & Strong
Reproduced with permission under license



PMCAMPUS.com / Mokanova Inc

© 2003-2016 Mokanova Inc and its licensors. All rights reserved for all countries.
PMCAMPUS.com is a trademark and service of Mokanova Inc.

PMI, PMBOK, PMP are owned and registered marks of the Project Management
Institute, Inc.

Single use only. Do not download, print, duplicate, and share

Can we legislate for agile? The DOD Experience

We want to work agile into our acquisition and systems engineering lifestyle.¹

Richard Spires
CIO, Department of Homeland Security

A major inhibitor to adoption of the agile approach in governments around the world is the regulations and 'best practice' guidance that have built up over the years. These have been centrally produced in order to improve technical development in diverse government bodies, but often they have just ended up stifling effectiveness.

This chapter looks at the regulations that apply to project management in the US Department of Defense (DOD), and how they either encourage or discourage agile adoption. It is possible for Government organizations to be agile despite some difficult and prescriptive regulations, but it is an up-hill struggle.

I start the story in 1988 with the impact of the publication of the DOD-STD-2167 standard which was widely, but mistakenly, interpreted as mandating a waterfall approach. The subsequent 2167A standard tried to clarify and stress the need for incremental delivery, but waterfall projects continued unabated even though efforts were made to sweep up all the regulations under an umbrella standard (Mil-Std-498).

Congress then got involved, and the Clinger-Cohen Act was passed in 1996 which begat the DOD-5000 series of regulations which tried to reinforce the need for evolutionary development. Unfortunately an inflexible approach to project management continued, so the National Defense Act of 2012 required the DOD to review its regulations yet again, and the result was the release of interim guidance called the "IT Box" approach, which attempts to reinforce the intent of the DOD-5000 series of guidance.

This sorry history of attempts to inculcate an agile military development approach shows that you cannot legislate for a change in attitudes. A leadership approach is needed to implement agile project management in the DOD and its suppliers, not further regulation.

Background

The US Department of Defense (DOD) is the USA's largest employer. It processes a huge

amount of information. As of 2010, it had more than 170,000 people working in management and IT. It processed business data on 7.3m people either on active duty, in the National Guard, working in civilian posts or receiving benefits.²

Government Procurement Processes prefer BDUF

The US military standard DOD-STD-2167 was mentioned briefly in the above discussion on waterfall standards.³ McDonald has charted the history of the rise and fall of this and other attempts to impose waterfall, top-down methods. He found that:

“Although military procurers could not directly impose such regulations on their suppliers, they could use contractual software development standards as a way to reach into private companies and perform the same disciplining function indirectly. Many in the military certainly believed that programmers were badly in need of such discipline.”⁴

The Joint Logistics Commanders (JLC) developed the DOD-STD-2167 standard as a default approach to software design for weapons systems – but flexibility was always intended, even if it was rarely achieved. The standard sets out clearly that:

“Software development is usually an iterative process, in which an iteration of the software development cycle occurs one or more times during each of the system life cycle phases.”⁵

In addition, the standard provided an appendix specifically requiring:

“Tailoring of activities, products, and reviews required during each software development phase.”⁶

The standard was approved in 1985, but sub-contractors still railed against the imposition of processes on their internal development processes. Many criticisms were leveled at the standard – especially a general misunderstanding that it forced a top-down approach and that it contained archaic requirements that were a decade out of date.

Within a year, in 1986, a replacement standard DOD-STD-2167A was drafted and, after discussion, was authorized in 1988. This revision removed all mention of the top-down approach that the earlier standard had appeared to mandate. The foreword made clear this new approach:⁷

“This standard is not intended to specify or discourage the use of any particular software development method. The contractor should select software development methods (for example rapid prototyping) that best support the achievement of contract requirements.”⁸

However, a recent report by the National Research Council found that:

“The DOD is hampered by a culture and acquisition-related practices that favor large programs, high-level oversight, and a very deliberate, serial approach to development and testing. Programs that are expected to deliver nearly perfect solutions and that take years to develop are the norm in the DOD. ... Moreover, the DOD’s process-bound, high-level oversight seems to make demands that cause developers to focus more on process than

on product, and end-user participation often is too little and too late. ... The key to resolving the chronic problems with the DOD acquisition of IT systems is for the DOD to adopt a fundamentally different process—one based on the lessons learned in the employment of agile management techniques in the commercial sector.”⁹

The impact of the Clinger-Cohen Act

The Clinger-Cohen Act (CCA) formed part of the Information Technology Management Reform Act which itself was part of the 1996 National Defense Authorization Act. The CCA transferred ultimate responsibility for IT developments to the OMB and required every federal agency to develop and maintain an Information Technology Architecture.

Because of the CCA, the DOD-5000 series of instructions were issued. These are a detailed set of instructions to government defense bodies dealing with project management of systems development. Some of these instructions are intended to facilitate incremental development, and therefore can be seen as ‘agile friendly’. Some of them seem to require a waterfall approach to projects, and therefore could inhibit agile adoption.

An example of an ‘agile friendly’ requirement is for systems to be built using a Modular Open Systems Approach (MOSA). This does not refer to what are, confusingly, commonly known as *open systems*. MOSA refers to systems that have clearly understood connections to other modules. The objective is to be able to build systems out of basic building blocks. They should have easy to understand interfaces that programmers can easily take apart and reuse in different configurations – like Lego[®] bricks. The MOSA approach is expected to help incremental implementation, and make upgrades easier once a system is implemented.¹⁰

The massive Defense Acquisition Guide (DAG) is a 900+ page long document that pulls all of the DOD-5000 series and supplementary guidance together in one place. It is reissued on a monthly basis as various items are revised. It defines two basic approaches to projects: *evolutionary acquisition strategies* and *single step strategies*. An *Acquisition Strategy* is required to justify which approach is planned for each project. The DAG states that the “DOD preference is evolutionary acquisition”. This new evolutionary life cycle was designed to specifically meet the requirements of the CCA which requires that:¹¹

- ◆ Each acquisition supports core, priority functions
- ◆ Outcome-based performance measures are linked to strategic goals
- ◆ Processes are redesigned to reduce costs, improve effectiveness and maximize the use of Commercial Off the Shelf (COTS) solutions
- ◆ The right government department or private company carries out each function
- ◆ An Analysis Of Alternatives (AOA), Return on Investment (ROI) and a whole Life Cycle Cost Estimate (LCCE) is made
- ◆ Measures and accountability for the progress of each program are established
- ◆ Each acquisition is consistent with an overall architecture called the Global

Information Grid

- ◆ Each program has a compliant information assurance strategy
- ◆ All important systems must be registered with the DOD Chief Information Officer (CIO).

Problems with the DOD Evolutionary Process

However, some of DOD-5000 is not so 'agile friendly'. The need to drive delivery in regular increments driven by immovable deadlines is not reinforced by the DAG. It does not clearly articulate the *timebox* concept that scope should be cut as required to meet timescales. Projects are still required to produce a great deal of pre-planning documentation before development work can be authorized to proceed, even on incremental projects:

“(Even in) an evolutionary (project), the Acquisition Strategy should fully describe the initial increment of capability (i.e., the initial deployment capability), and how it will be funded, developed, tested, produced, and supported. The Acquisition Strategy should preview similar planning for subsequent increments, and identify the approach to integrate and/or retrofit earlier increments with later increment improvements.”¹²

In 2009, the Defense Science Board (DSB) reported to Congress with an evaluation of the DOD's Information Technology (IT) developments. The report proposed a new acquisition process for IT and especially stressed the importance of continuous user participation and iterative development of technology.¹³ In parallel, the Congress Panel on Defense Acquisition Reform came to similar conclusions:

“In the context of the acquisition of IT, the Panel finds that the existing requirements process is ill-suited for the rapidly evolving nature of the IT marketplace which requires an iterative dialogue on requirements. The current process is too inflexible and prone to the kinds of over-specification that has long been an issue.”¹⁴

More attempts to control behavior by legislation ensued. Additional sections were added to the National Defense Authorization Act for fiscal year 2010 onwards. The requirements pulled towards and away from the agile approach. Section 804 required the Secretary of Defense to complete a review of the process for identification and acquisition by "an iterative approach to urgent operational needs".¹⁵ However, Section 805 still required extensive and specific processes for planning and oversight processes for the acquisition of major automated information systems. The Secretary of State was required to ensure that EVA is used to track all IT projects. Although some have attempted to use EVA on agile projects the tracking of spend against detailed, pre-planned activities is seen as many as a barrier to agile working (see page **Error! Bookmark not defined.**)¹⁶

Governing Requirements using the IT Box

The Joint Oversight Requirements Council (JORC), chaired by the Chairman of the Joint Chiefs of Staff, has provided a useful impetus for moving towards more incremental development.¹⁷ JORC gives overall priorities to new acquisitions across all military services, and in 2010, it released some guidance for business systems development projects to supplement DOD-5000 which included a technique called the *IT Box*. This defines how performance and cost ranges should be agreed and how authority for change approval should be delegated. The idea is to define clear boundaries and improve delegation to development projects.¹⁸

The IT Box is so called because it delineates four boundaries that control, inform, constrain, and direct a project. An Information Systems Initial Capabilities Document (IS ICD) is then produced to kick off the project with just enough design (EDUF), rather than too much. The IT Box is illustrated in **Figure 1**. It defines the governance of change control for the essential requirements for new software. It delegates detailed decisions to those responsible for development, while keeping control over strategic decisions.

The boundaries are flexible to the extent that the team responsible for development may negotiate changes as required, but remain strong enough to ensure consistency and control of projects from a strategic point of view. The four walls around the box of requirements are:¹⁹

- ◆ Governance of requirements: a statement as to how the contents and timing of each release will be agreed, tracked, and controlled – especially how the users and the team will collaborate. Of importance here is a statement as to how stakeholders from a multi-organization environment will interact and who will act as the final arbiter and decision-maker. Scrum defines a similar role of the *product owner*. DSDM defines the role of a business sponsor who would appoint a business visionary for day-to-day collaboration with the team.
- ◆ Operational Environment: This defines the constraints that exist for the hardware on which the computer system must run. Emphasis is placed here on determining the *total cost of ownership* (TCO) over the whole life of the use of the system, which might be for decades. This requires the breaking out of costs into annual estimates, including a budget for the final decommissioning of the solution when it is either no longer required, or has to be replaced. Worked into these costs must be the costs and timing of the *refresh cycle* to upgrade the systems as the hardware and operating software are upgraded and replaced over time.

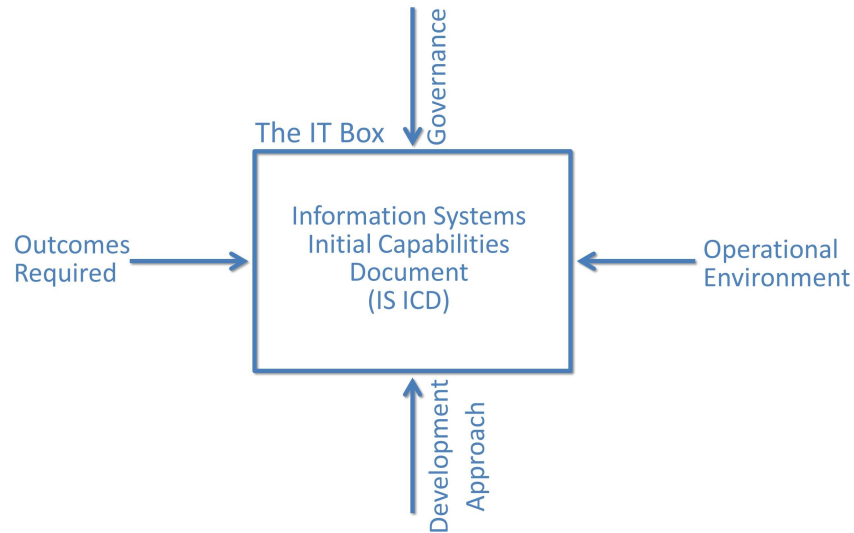


Figure 1: The DOD IT Box: a Framework for Capturing Essential IT Requirements

- ◆ **Development Approach:** This defines how the systems will be acquired and developed – not only must the cost of any contract with suppliers be included, but also a budget for the DOD to run the procurement and manage with suppliers.
- ◆ **Outcomes Required:** these define the effectiveness of the capabilities of the required system – in other words not just what will be delivered, but whether it has a positive operational impact. These outcomes are broad and shallow in definition. Broad, because the entire scope of the program must be included, not just the first phase. Shallow, because it is the minimum outcomes that must be met are focused on. This avoids the problems of either over-specifying an expensive solution, or committing to assumptions on what the technology can achieve. It is important to balance the need to avoid trying to achieve impossible outcomes with the need to explore potential opportunities to harness unrecognized technological possibilities that will emerge as development progresses.

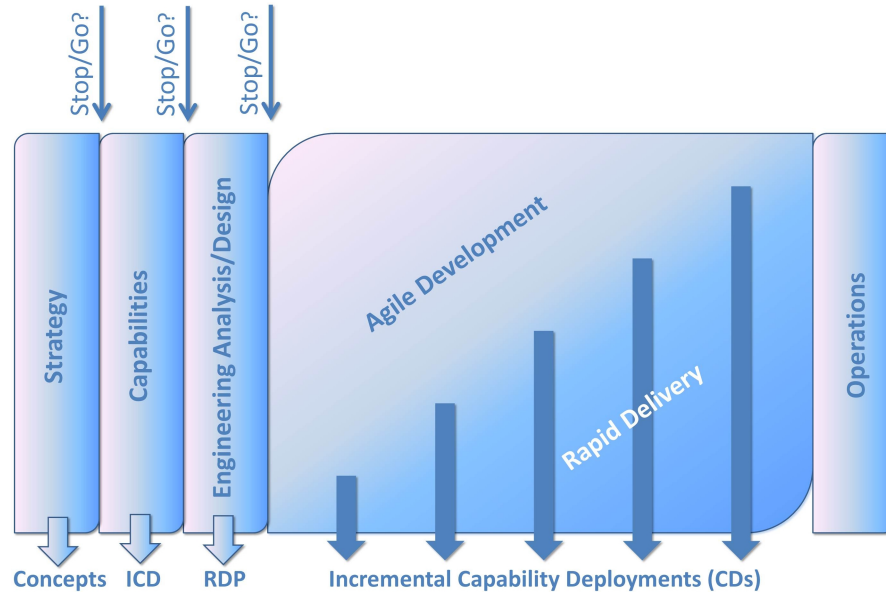


Figure 2: DOD Requirements/Acquisition Process Requirements process²⁰

Careful planning is required for each Capability Drop (CD) (see **Figure 2**). Software is broken into small, independent modules, and extensive use is made of re-useable *widgets*, which are modules that perform simple, common operations across many applications. The project board, not the team, agrees the timing and objectives for each CD. This means that the sponsor not the team has control over the sequencing of CDs and prioritization of development as a whole. The team, however, has freedom to decide how to meet those objectives. They do not need to seek top management permission to detailed changes to over-complex requirements specifications and detailed project plans.

One criticism of the use of the IT Box is that it could lead to the treatment of software development as a separate activity ring-fenced from hardware development and implementation planning by governance processes that could remain too inflexible. Projects using an agile approach within the IT Box concept must be careful to adopt an integrated team approach, and not have separate software and hardware test teams, for example.²¹

It is a hybrid model somewhere between waterfall and agile. The initial work is segmented into three planning phases (strategy, definition of capabilities, and engineering analysis/design). Before iterative development can get underway these three *gates* must be hurdled. Once these gates are passed, the possibility of change of direction is reduced. There is the need for a sign-off by the relevant MDA of an Information Systems Initial Capabilities Document (ICD), and then the production and agreement of further planning documents Requirements Definition Packages (RDPs) before development work can get under way. The development is further decomposed into Capability Drop (CD) documents that define each phase of delivery.

A more detailed look at the proposed iterative development required is shown in **Figure 3**. Each CD must consist of a lengthy prototyping phase with a discrete *stop/go decision* by the MDA before build can commence. These two phases may stretch up to 2 years in length before any deployment is achieved.

Phasing of delivery into pre-planned CD documents and the separation of prototyping from build and implementation may constitute a type of iterative development. However, is it really driven by emerging understanding of the solution? Or is it the substitution of a grand waterfall with a series of smaller (but not so small) document driven waterfalls? An improvement on the single step waterfall approach, but not in line with the 12 Agile Manifesto Principles since the need to “seek out requirements, even late in development” is not encouraged. The DOD still mandates that requirements for each CD must be fully developed before each step starts. The delivery timescales are in 12 to 18 month phases, which does not equate to the principle of delivery frequently, in weeks or months rather than years. Finally, the concept of a *sustainable pace of development* is not promoted.

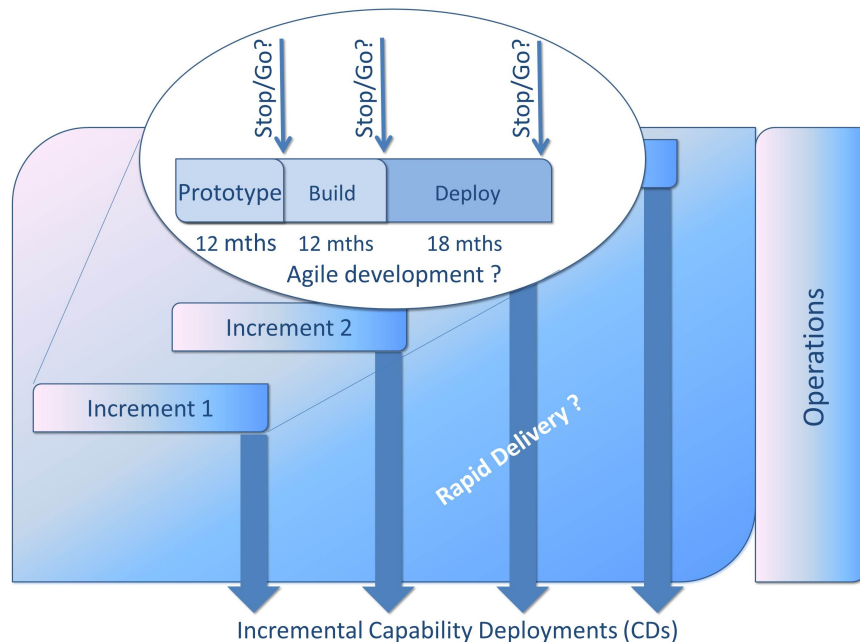


Figure 3: A criticism of the DOD BCL model is that it treats each phase of development as a mini-waterfall project

The agile approach encourages a steady and efficient output of work rather than the team starting and stopping at a series of artificial review gates. The sponsors, developers, and users should be able to maintain a constant pace indefinitely. At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly. Stephany Bellomo has highlighted the problems implied by the BCL:

“This directive straddles the new and the old acquisition processes. ... Clearly, this is interim guidance that is trying to move in the direction of new acquisition concepts, but is still holding rather tightly to the old.”²²

However, the SEI notes that although DOD-5000 does not preclude the use of agile:

“(The tactical and strategic benefits from agile are) not likely to occur without changes to the traditional DOD mindset.”²³

Table 1: Explanations of selected DOD Acronyms²⁴

Acronym	Definition
BCL	Business Capability Lifecycle: An interim approach for the development of defense business systems. Created as a supplement to DOD-5000, it is an integrated approach to producing JCIDS documents, and meeting the requirements of the Investment Review Board (IRB) and the Defense Business System Management Committee (DBSMC).
CD	Capability Drop: A lower level document that specifies the characteristics of a “widget” or “app” for partial deployment of the solution.
IS ICD	Information Systems Initial Capabilities Document: the minimum requirements for a new system “based upon what is achievable with today’s technology”.
JROC	The Joint Requirements Oversight Council reviews and agrees all Joint Capabilities Integration and Development (JCID) documents
KPP	Key Performance Parameters.
MDAPs	Major Defense Acquisition Programs
MDAs	Milestone Decision Authorities – The bodies authorized by DOD-5000 to tailor the regulatory information requirements and acquisition process.
OSD	Office of the Secretary of Defense: The staff function that supports the Secretary of Defense. This includes the Deputy, Under and Assistant Secretaries, and Directors.

Questions

1. Section 804 is part of US public law and requires specific development techniques to be used in defense projects. Is it desirable to state in law that particular methods should be used?
2. Do you feel that the 'IT Box' could work as an enabler for agile to flourish, or as a straightjacket that could stifle agility?
3. The 'IT Box' was a response to the requirements of the JROCM 008-08 memorandum.²⁵ Do you think that it meets those requirement

¹ {Agile development gaining in popularity #310}

² {Deputy Chief Management Officer 2010 #309: 3}

³ {DOD-STD-2167A 1988 #83: 53}

⁴ {McDonald 2010 #77: 37}

⁵ { DOD-STD-2167 1985 #84: 1}

⁶ { DOD-STD-2167 1985 #84: 78}

⁷ { DOD-STD-2167 1985 #84: 78}

⁸ { DOD-STD-2167A 1988 #83: 3}

⁹ {Achieving effective acquisition of information 2010 #314: ix /pageroman}

¹⁰ {DOD 2012 #402: 295–296} and {DOD 2012 #402: 48–49}

¹¹ {USD(AT&L) 2008 #323: 48, Table 8}

¹² {DoD 2012 #402: 48}

¹³ {Defense Science Board 2009 #307}

¹⁴ {PANEL ON DEFENSE ACQUISITION REFORM 2010 #319: 23}

¹⁵ {Congress 2009 #308: Sec. 804}

¹⁶ {Congress 2009 #308: Sec. 805 4259}

¹⁷ {Wills 2012 #312: 4}

¹⁸ The Joint Capabilities Integration and Development System (JCIDS) "IT Box" {Wills 2012 #312}, Command and Control Requirements Management {DOD Director of Joint Staff 2010 #325} and {Deputy Chief Management Officer 2010 #309}. At the time of publication this model was described as an 'interim' model yet to be incorporated into DOD 5000.02.

¹⁹ {Wills 2012 #312: 9–11}

²⁰ {Wills 2012 #312: 17}

²¹ {Wills 2012 #312: 12}

²² {Bellomo 2011 #306: 4}

²³ {Lapham 2012 #311}

²⁴ I collated and derived these from several sources. See the following for more information: {USD(AT&L) 2008 #323}; {Wills 2012 #312}; {J-8 2001 #326}

²⁵ {Wills 2012 #312: 4}