Improving Acquisition to Support the Space Enterprise Vision

Supplemental Appendixes on Acquisition Concepts

FRANK CAMM, BRIAN DOUGHERTY, THOMAS C. WHITMORE

Prepared for the United States Air Force
Approved for public release; distribution unlimited

RAND CORPORATION

For more information on this publication, visit www.rand.org/t/RR2626

Published by the RAND Corporation, Santa Monica, Calif.

Preface

The national security space community recognizes that the threat to the space domain is increasing and, as a result, the current U.S. space enterprise is not resilient enough to survive a conflict that extends to space.[1] At the time this report was written, Air Force Space Command (AFSPC) was leading an effort to build a space force that will prevail in the face of the threat. Specifically, AFSPC envisioned an integrated approach—the Space Enterprise Vision (SEV)—to building a force across all space mission areas, coupling the delivery of space capabilities to the warfighter with the ability to protect and defend space capabilities against emerging threats. Since that time, the U.S. Space Force has stood up as a separate service under the Department of the Air Force, and the implementation of the SEV has transitioned to the Space Force.

The SEV requires significant changes in how the Air Force designs, develops, acquires, and operates its space systems to build a future space enterprise. With respect to acquisition, the SEV demands more responsive acquisition and improved synchronization of systems across the space enterprise. AFSPC and the Space and Missile Systems Center asked RAND Project AIR FORCE (PAF) to help the Air Force identify ways to achieve those SEV acquisition goals. To that end, RAND PAF examined how space acquisition could be improved to support the SEV, focusing on three promising acquisition approaches to achieving the SEV goals: modular open systems architecture (MOSA), agile acquisition, and rapid prototyping. That research is documented in "Improving Space Acquisition to Support the Space Enterprise Vision." This document is a companion report, which provides additional background and reference information on MOSA, agile acquisition, and rapid prototyping. The challenges and recommendations identified in this report remain relevant to the Space Force as the new service aims to implement an alternative acquisition system to accelerate space acquisition and improve synchronization across space, ground, and user segments.

This research was sponsored by the Director of Strategic Plans, Programs, Requirements and Analysis, Headquarters Air Force Space Command; and the Vice Commander, Space and Missile Systems Center and was performed within the Resource Management Program of RAND PAF. The research was conducted between October 2016 and August 2017. This document should interest those involved in the planning and implementation of the SEV and those engaged in the acquisition of space systems.

[1] AFSPC Public Affairs, "AFSPC Commander Announces Space Enterprise Vision," April 11, 2016.

RAND Project AIR FORCE

RAND PAF, a division of the RAND Corporation, is the U.S. Air Force's federally funded research-and-development center for studies and analyses. PAF provides the Air Force with independent analyses of policy alternatives affecting the development, employment, combat readiness, and support of current and future air, space, and cyber forces. Research is conducted in four programs: Force Modernization and Employment; Manpower, Personnel, and Training; Resource Management; and Strategy and Doctrine. The research reported here was prepared under contract FA7014-16-D-1000.

Additional information about PAF is available on our website: http://www.rand.org/paf/

Contents

Figures

Tables

Acknowledgments

We thank Maj Gen Nina M. Armagno, Brig Gen Mark A. Baird, Maj Gen John E. Shaw, and Brig Gen Philip A. Garrant for sponsoring this work. We are also grateful for the support of our action officers, Maj Robert C. Small and Capt Shaun C. Cain, who opened many doors for us and provided helpful insights throughout the project. We also benefited from many discussions with individuals throughout the Air Force and industry, too numerous to mention by name.

At RAND, we thank Phil Anton, David Galvan, and Mark Lorell for their insightful comments, which further improved the report. We are also thankful to Obaid Younossi, Barbara Bicksler, and Jerry Sollinger, who provided helpful critiques and other support throughout this project. The content and conclusions of the report, however, are the responsibilities of the authors.

Abbreviations

AFSPC	Air Force Space Command
APB	advanced processing builds
A-RCI	Acoustic Rapid COTS Insertion
CDD	Capability Development Document
COTS	commercial off-the-shelf
DevOps	Development Operations
DFARS	Defense Federal Acquisition Regulation Supplement
DoD	Department of Defense
DoDD	DoD Directive
EVM	earned-value management
FAR	Federal Acquisition Regulation
FFF	form, fit, and function
GPS	Global Positioning System
ICD	Initial Capability Document
ISO	International Standards Organization
ITU	International Telecommunications Union
JCIDS	Joint Capability Integration and Development System
JMS	Joint Space Operations Center Mission System
MAJCOM	Major Command
MOSA	modular open systems architecture
NDAA	National Defense Authorization Act
OMIT	operation, maintenance, installation, and training
OSD	Office of the Secretary of Defense
OTA	Other Transaction Authority
PE	program element
PM	program manager

POR	program of record
RDT&E	research, development, test, and evaluation
S&T	science and technology
SEV	Space Enterprise Vision
SMC	Space and Missile Systems Center
STRG	Submarine Tactical Requirements Group
T2BT	technology to be transitioned
TI	technology insertion
TTA	Technology Transition Agreement
USB	Universal Serial Bus
USB-IF	USB Implementers Forum
USC	United States Code

Appendix A. Modular Open Systems Architecture

This appendix provides supplemental information on our analysis of modular open systems architecture (MOSA) as a potential acquisition approach to support the Space Enterprise Vision (SEV) objectives. In particular, we discuss the technical definition of MOSA, high-level benefits of MOSA, standards management models, and data rights issues.

Modular Open Systems Architecture Technical Definition

To understand fully the benefits of a MOSA strategy, one must understand the technical aspects of a MOSA architecture. A MOSA system has three key technical characteristics:[1]

1. It is modular, being decomposed into architectural components that are cohesive, loosely coupled with other components (and external systems), and encapsulate (hide) their implementations behind visible interfaces.
2. Its key interfaces between architectural components conform to open interface standards (that is, consensus based, widely used, and easily available to potential users).
3. Its key interfaces have been verified to conform to the associated open interface standards.

The MOSA terms “modular” and “loosely coupled” mean that the system may be separately developed at the subcomponent level and integrated by a third party. This process differs from the more monolithic model used by many Department of Defense (DoD) programs where a single entity (i.e., the prime contractor) is responsible for both component development and system integration. Regardless of whether that entity procures a component from a subcontractor or develops it in-house, it is responsible for the selection of that component and its integration into the full system. If a subcomponent is not designed modularly, upgrade of that component may require significant system-level modification or replacement if new capability is desired. Employment of a modular design enables capability development and upgrade at a lower functional level rather than requiring periodic system redesigns or system replacement.[2]

The term “cohesive” in the MOSA definition has two aspects. First, a robust integration effort is performed by a designated entity to ensure proper operation of the system. Secondly, despite the fact that the system is developed in pieces, interface requirements are agreed to ahead of time so that each entity involved in system development is working as cooperatively as possible during its part of the design effort. Standardized interfaces are required to enable component compatibility in a MOSA environment.

[1] Donald Firesmith, *Open System Architectures: When and Where to Be Closed*, SEI Insights Blog, Carnegie Mellon University, Software Engineering Institute, Pittsburgh, Pa., October 2015.

[2] U.S. Department of Defense, *Program Manager’s Guide to Open Systems*, Version 2.0, September 2004b.

Interface standards provide the technical information necessary to enable compatibility or interoperability among system elements. This involves the delineation of requirements for a specific interface between system elements. The interface can be hardware,[3] software,[4] or a combination of hardware and software elements. Interface standards define functionality that a component can rely on from other components through the interface when integrated into a full system. Individual component designs must also adhere to applicable requirements in the interface standard to ensure they provide their defined functionality to other parts of the system. Verification of conformance to standards is typically performed using tests specified by a standards governance organization. Standards governance organizations and models are discussed later in this appendix.

The DoD *Program Manager's Guide to Open Systems* uses the terms "open" and "proprietary" to describe a standard's breadth of use. The guide defines an open standard as one that is consensus based, widely used in industry, easily available to potential users, and published or maintained by a recognized industry standards organization. Proprietary standards, then, are defined as being owned by an individual or organization and may require a license or fee to be used.[5]

This definition can be somewhat problematic. A standard may be owned by a recognized organization and involve a relatively minor certification or membership fee but still be widely adopted in industry. Though not "open" according to the above definition, adoption of such a standard would carry many, if not all, of the benefits of adoption of an open standard. An alternative definition of open standard is provided by the International Telecommunications Union (ITU), a prominent international standards organization: "standards made available to the general public and are developed (or approved) and maintained via a collaborative and consensus driven process."[6] The ITU's official definition of open standards goes on to list other key elements of an open standard:[7]

- The standard is developed and managed over time using collaborative, transparent processes that involve voluntary and market-driven development that is open to all interested parties.

---

[3] Hardware-specific interface parameters could include component physical size, types of mechanical connector, power distribution bus voltage/current levels, and thermal interfaces. These examples are taken from Universal Serial Bus USB Technical Standard version 3.1 (Universal Serial Bus Implementer's Forum, *About USB Implementers Forum, Inc*., undated) and U.S. Department of Defense, *MIL-STD-704F, Aircraft Electric Power Characteristics*, Revision F, March 2004a.

[4] Software-specific interface parameters could include allowable data structures, types/structures of data file, message structures, communications protocol, allowable programming language libraries, and appropriate operating system partitioning schemes. These examples are from Future Airborne Capability Environment technical standard, edition 2.1 (Open Group, *Technical Standard for Future Airborne Capability Environment*, Edition 2.1, May 2014).

[5] U.S. Department of Defense, 2004b.

[6] International Telecommunications Union, *Definition of "Open Standards,"* November 11, 2015.

[7] International Telecommunications Union, 2015.

- The technical interests of the standard are "reasonably balanced" and are not dominated by any one interest group.
- The intellectual property rights essential to implement the standards are either available to all applicants for free or on "reasonable terms and conditions (which may include monetary compensation)."
- The technical standards are publicly available at a "reasonable price."
- The standards have ongoing support by some sort of standards organization.

Importantly, the emphasis in the ITU's definition is on the level of adoption of the standard rather than the ownership model or cost associated with use of the standard. Next, we explore a familiar example of a commercial standard to demonstrate the above principles.

Standards Management Models

An example of an open commercial standard that governs hardware and software requirements is the Universal Serial Bus (USB) standard. USB was developed in the early 1990s by a consortium of industry entities with a goal of growing the market for computer peripherals by standardizing the peripheral-to-PC connection point. Before this initiative, computer peripherals each required a specialized interface, which not only created challenges for PC users but also PC manufacturers, who were unable to keep up with consumer demand for connection points for PC peripheral devices. Adopting a low-cost, common PC peripheral interface would reduce the barrier to entry for PC peripheral designers, which could in turn enable new uses for PCs in the commercial market and benefit all members of the consortium.[8]

Early versions of the USB standard focused on such issues as appropriate data transmission bandwidth, PC-to-peripheral power transmission specifications, plug-and-play capability, enabling backward compatibility between versions of the USB standard, and standardization of the hardware interface.[9] With these standards established, newer versions of the standard have grown to support newer capabilities, such as battery charging and much higher data rate communication.[10]

Notably, the USB standard does not specify what type of connector is required at the peripheral side of the peripheral-to-USB connection. This side of the system interface is not "key" for the USB standard, and peripheral developers are free to create their own proprietary connections for this interface. An example of a proprietary device connector standard is Apple's Lightning connector for many of their devices. Apple manufactures USB-to-Lightning cables to enable connection of their Lightning-compatible devices to a variety of USB-capable devices, but the Lightning connector is not used for non-Apple applications. Additionally, Lightning

---

[8] Intel Corporation, *Two Decades of "Plug and Play": How USB Became the Most Successful Interface in the History of Computing*, 2014.

[9] Connector size, pin location, wire gauge, and so on.

[10] Intel Corporation, 2014.

connector cables are equipped with a special authentication device to prevent the manufacture of unlicensed third-party cables.[11] This makes Lightning an example of a proprietary standard as previously defined.

The voluntary development and adoption of a common PC-to-peripheral standard is an example of industry recognition of shared benefit associated with commonality and development of an open standard to achieve these benefits. Voluntary standardization such as this does not always occur. Additionally, the development and management of a technical standard requires united long-term effort. For example, standard tests and certifications must be created to demonstrate a component's compliance with a given interface standard. Additionally, standards must be periodically reviewed and revised to ensure continued relevance in the commercial market. The creation and operation of a common governance body provides a mechanism to handle these issues and share costs across the set of organizations interested in using the standards. For the USB standard, the USB Implementers Forum (USB-IF) is the entity responsible for issues associated with USB standard maintenance. The USB-IF is a nonprofit organization comprising a board of directors, corporate officers, and multiple working committees with representation from industry organizations that are members of the forum. Though the USB-IF has made access to the USB standard free of charge, USB-IF working committee membership and access to USB compliance certification require membership in the USB-IF. Membership in the USB-IF involves payment of a minor fee.[12]

When voluntary industry standardization is unlikely or infeasible, government or other third-party involvement in standards development and management may be beneficial to driving consensus toward common standards. Nationally focused organizations like the U.S. Government's National Institute of Standards and Technology and DoD Standardization Program Office participate in the development and maintenance of standards applicable to interfaces of interest to the U.S. government. The DoD Standardization Program Office specifically is involved in the development of various specification documents, such as federal and/or military specifications, qualified product lists, and DoD international standardization agreements.[13] Other international bodies, such as the International Standards Organization (ISO) and ITU, perform similar functions at a broader scale.[14] Active, long-term involvement of a standards management organization is necessary to ensure a standard remain current and

---

[11] Eric Limer, *Here's the Chip Apple Is Using to Stop you from Buying Cheap Cables*, Gizmodo, September 24, 2012.

[12] Universal Serial Bus Implementer's Forum, undated.

[13] U.S. Department of Defense, Defense Standardization Program, *About the DSP,* undated.

[14] ISO is made up of a body of international participants who may individually identify a technical need for a new standard, collaboratively develop the standard to meet consensus needs via technically focused committees, provide guidance on compliance certification, and manage standard revisions over time. ISO does not itself determine a need for a standard but rather provides a forum for international collaboration and consensus to develop new standards and develop consensus for adopting a given standard. See International Standards Organization, *How We Develop Standards*, undated.

certification processes are carefully managed and enforced. The absence of such united management would likely result in the proliferation of competing, nonstandardized designs or obsolescence and replacement of the existing standard. This is what happened to the proprietary PC peripheral connections that were superseded by USB.

High-Level Benefits to Implementing Modular Open Systems Architecture

Once the MOSA's modular framework is in place, key interfaces have been identified, and appropriate widely adopted standards and associated conformance tests have been selected, system owners have the start of a flexible, upgradeable environment to operate within. Flexibility and upgradeability are not the only benefits of a MOSA architecture, and achieving any MOSA-related benefits is not without challenges. Benefits associated with MOSA implementation will be explored in more detail below.

The U.S. Deputy Assistant Secretary of Defense for Systems Engineering identifies five primary benefits associated with MOSA systems:[15]

- enhanced competition
- incorporation of innovation
- enabling of cost savings/cost avoidance
- improvement of interoperability
- facilitation of technology refresh.

Enhanced competition can result from a MOSA because a modular framework based on standardized interfaces enables system designers to consider a broad supplier base for each module. When a widely adopted standard or set of standards is adopted for an interface, any supplier familiar with a product that is compatible with the selected standard may contribute a technical solution. This includes suppliers beyond a traditional supplier base, if the standards that are selected are adopted widely enough. Having the ability to engage suppliers regularly outside a typical supplier base can help prevent *vendor lock*. Vendor lock is a phenomenon present in acquisition environments like DoD where design agents are limited in their choices of component supplier. It is the result of an acquisition environment where suppliers provide highly specialized products or services, and few other suppliers can provide the same outputs. The ability to engage a broader supplier base provides an increased credible threat of competition, which in turn helps reduce vendor lock. Being able to engage a broader than usual supplier base also provides opportunities for innovative solutions and cost savings, discussed further below.

Improved innovation can stem from a MOSA for a few reasons. For one, as mentioned above, the ability to engage atypical suppliers resulting from a MOSA provides an opportunity to pursue solutions outside of the traditional supplier ecosystem. It can also be said that competition

---

[15] U.S. Department of Defense, Office of the Deputy Assistant Secretary of Defense for Systems Engineering, *MOSA Initiative*, undated.

promotes innovation in and of itself; any supplier interested in a competitive contract is incentivized to provide superior, innovative technical solutions.[16] Finally, the ability to quickly and flexibly reconfigure a modular system to respond to emergent needs provides an opportunity for operational innovation. This capability is particularly important when the threat is rapidly evolving.[17]

Cost savings/avoidance are also a side effect of increased competition and innovation. Vendors involved in acquisition environments with credible competitive threats have incentives to reduce cost when possible to increase the likelihood of winning contracts. Increased competition is not the only way that MOSA enables cost savings. When common modular architectures and widely adopted standards are agreed to at a larger scale than a single system—the *enterprise* level—systems can share investment risk and reuse technologies or full modules across applications. Technology and module reuse across an enterprise reduces the need for unique development efforts, eliminating significant development costs. In a DoD setting, enabling technology reuse includes the potential for easier integration of commercial off-the-shelf (COTS) technology into a system, provided selected standards are widely used in commercial markets.[18]

Implementation of MOSA principles at an enterprise level, rather than just a single system level, can enable improvements in interoperability across the enterprise. Interoperability is defined by DoD Directive (DoDD) 5000.01 as "the ability of systems, units, or forces to provide data, information, materiel, and services to and accept the same from other systems, units or forces, and to use the data, information, materiel, and services so exchanged to enable them to operate effectively together."[19] The use of common architecture and communications standards for systems in an enterprise, such as the Open Mission Systems and Unmanned Aerospace Systems Command and Control Initiative standards, is one way to ensure that systems across an enterprise can effectively communicate and interoperate. Communications standards govern message structure and data-sharing mechanisms to ensure multiple systems can communicate with one another.

Finally, MOSA facilitates improved technology refresh implementation by providing system agents the opportunity to replace modules or subsystems without changing an entire system. This improves a system's responsiveness to rapidly changing capability needs and also its obsolescence posture as technology within its modules ages.[20] Widely adopted interface standards that are designed to enable backward compatibility, as is true for USB, enable MOSA

---

[16] U.S. Department of Defense, *Open Systems Architecture Contract Guidebook for Program Managers*, Version 1.1, June 2013b.

[17] U.S. Department of Defense, 2004b.

[18] U.S. Department of Defense, 2013b.

[19] U.S. Department of Defense, *Directive 5000.01: The Defense Acquisition System*. November 20, 2007.

[20] U.S. Department of Defense, 2013b.

system modular components to be upgraded or refreshed on module-appropriate time lines—provided they adhere to the appropriate standards documentation. Importantly, MOSA architectures enable upgrades on both subsequently produced and deployed units.

Notably, all five of the above benefits directly translate to primary goals of the SEV: improving capability acquisition responsiveness and broadening the view of space assets to an enterprise scale.

Data Rights Issues for Modular Open Systems Architecture

The main report contains several recommendations that directly pertain to data rights within an acquisition program. An effective data rights strategy can enable competition throughout a program's life cycle, allow for future program iterations and sustainment, and encourage innovative suppliers to participate in competitive DoD environments. As reference to the main report, this section briefly introduces the general terminology associated with data rights, highlights specific implications for data rights in MOSA, and identifies observations of data rights use in space acquisition.

Data Rights Terminology

The DoD data rights are codified in 10 U.S. Code §§ 2320, 2321, and are a vital part of any program. While defined in several ways, the term *data rights* is shorthand for the license rights that DoD acquires for technical data and computer software.[21] Technical data are best described as recorded information of scientific or technical nature and excludes computer software,[22] data related to contract administration, and financial information. Examples of technical data include design review data packages, engineering drawings, technical specifications, interface control documents, and test plans.[23] A program's strategy for acquiring license rights to technical data and computer software can vary if the acquisition is developing a hardware-heavy, software-heavy, or combined hardware/software system.

The Defense Federal Acquisition Regulation Supplement (DFARS) has defined several data rights categories based on whether the product contains noncommercial or commercial data. There are four categories of noncommercial technical data that DoD can purchase: *unlimited rights*, *government purpose rights*, *limited rights*, and *specifically negotiated license rights*. The same categories apply for noncommercial computer software with the exception of limited

---

[21] U.S. Department of Defense, *Intellectual Property Strategy*, Department of Defense Open Systems Architecture Data Rights Team, August 2014.

[22] Computer software is defined by Defense Federal Acquisition Regulation Supplement (DFARS) as computer programs, source code, source code listings, object code listings, design details, algorithms, processes, flowcharts, formulas, and related material that would enable the software to be reproduced, recreated, or recompiled. See DFARS §§ 252.227-7013(a)(3) and 252.227-7014(a)(4).

[23] DFARS § 252.227-7013(a)(15).

rights, which is replaced by *restricted rights*.[24] In addition to regulations surrounding these terms, specific regulations apply to Small Business Innovation Research (SBIR) program–supplied systems and their associated data rights. Figure A.1 lists the aforementioned data rights categories and the permitted uses of data associated with noncommercial products. With respect to commercial systems, DoD rights to technical data are very similar to limited rights, unless the data were previously provided without restrictions.[25] Notably, the government's ability to claim specific types of data rights is dictated by those who financed the development of the system.[26]

While the claim to specific types of data rights, notably design information (e.g., technical data package),[27] is dictated by those who financed the development of the system (see 10 United States Code [USC] 2320), there is a set of standard data rights to which DoD is always entitled. These data types include form, fit, and function (FFF) data; operation, maintenance, installation, and training (OMIT) data; and computer software documentation.[28] DoD has unlimited rights to these data for noncommercial items, as well as similar rights for commercial items.[29] In MOSA-designed systems, the availability of FFF and OMIT data can be a key enabler in implementing an open systems design approach, which will be explicated below.

During contract negotiations between the DoD customer and defense contractor for data rights, it is also vital to address the data deliverables required for the program. The U.S. Army Product Data and Engineering Working Group developed a framework to separate product data types into three bins (e.g., product definition information, product operational information, and associated information) to prioritize what deliverables are necessary to define, manufacture, and

---

[24] Space and Missile Systems Center (SMC), *Acquiring and Enforcing the Government's Rights in Technical Data and Computer Software Under Department of Defense Contracts: A Practical Handbook for Acquisition Professionals*, 7th ed., Office of the Staff Judge Advocate, Space and Missile Systems Center, August 2015, p. 9.

[25] SMC, *Acquiring and Enforcing the Government's Rights in Technical Data and Computer Software Under Department of Defense Contracts: A Practical Handbook for Acquisition Professionals*, 3rd ed., Office of the Staff Judge Advocate, Space and Missile Systems Center, January 2011, p. 11.

[26] DoD is entitled to *unlimited rights* if the system is developed exclusively with U.S. government funding. If the system is developed with a mix of U.S. government and private funds, DoD is entitled to *government purpose rights*. DoD is entitled *limited/restricted rights* if the product is developed exclusively with private funds. See DFARS §§ 252.227-7013 and 252.227-7014.

[27] The *technical data package* is defined by MIL-STD-31000A as "a technical description of an item adequate for supporting an acquisition, production, engineering, and logistics support." It often contains the proprietary solution of the contractor and is considered the critical, and controversial, component in licensing data rights, U.S. Department of Defense, 2013a.

[28] U.S. Army, *Army Data & Data Rights (D&DR) Guide, A Reference for Planning and Performing Data Acquisition and Data Management Activities Throughout the DoD Life Cycle*, 1st ed., U.S. Army Product Data and Engineering Working Group, August 2015, p. 20.

[29] DFARS clause 252.227-7015 defines the standard rights for commercial items as the "unrestricted right to use, modify, reproduce, release, perform, display, or disclose technical data," as opposed to *unlimited rights*. It is DoD policy to acquire the minimum amount of necessary data rights when utilizing commercial technology.

Figure A.1. Noncommercial Data Rights Categories and Permitted Uses

Rights Category	Criteria for Applying Rights Category	Permitted Uses Within Government	Permitted Uses Outside Government
Unlimited	Developed exclusively at Government expense	No restrictions. Government can use, modify, reproduce, perform, display, release, or disclose data in any manner, and for any purpose whatsoever, and to have or authorize others to do so.	
Government Purpose	Developed using mixed funds from both the Government and the contractor	No restrictions.	May release for "Government purposes" only. No commercial use.
Specifically Negotiated License	Mutual agreement of both parties	Specific conditions as negotiated by the parties. Similar to Limited/Restricted but with additional rights. Cannot be more restrictive than Limited or Restricted rights.	
Small Business Innovative Research	Developed under SBIR contract	Any Government purpose.	Same as Limited Rights or Restricted Rights during SBIR data protection period.
Limited *(Technical Data)*	Developed exclusively at private expense	May not be used for manufacture; otherwise, no restrictions.	May not be released without contractor permission except for evaluation, emergency repair or overhaul. May share with CGSCs.
Restricted *(Computer Software)*	Developed exclusively at private expense	May not reverse engineer or decompile.	May not be released without contractor permission except as listed in DFARS 252.227-7014(a)(15). May share with CGSCs.

SOURCE: U.S. Army, 2015.
NOTE: CGSC = Covered Government Support Contractor. Data rights categories highlighted in green indicate that the government has flexibility to use that data for future product contract competitions. Those highlighted in red indicate that the government's ability to recompete a contract using those data rights is significantly limited, including potentially limiting the government to a sole source provider for that product.

support a system over its life cycle.[30] It is critical to manage data rights and data deliverables together because without requirements for the contractors to supply deliverables written into the contract, DoD's ability to take advantage of their data rights will be limited.

Implications for Data Rights in Modular Open Systems Architecture

Although there are exceptions, most DoD systems contain at least some proprietary technology to which limited and/or restricted data rights apply, which can potentially inhibit full and open competition during acquisition.[31] However, through defining a subsystem *manufacturing and sustainment model* and a *design documentation approach*, technology with such rights can be integrated into a competitive environment.

[30] U.S. Army, 2015, p. 9.

[31] U.S. Department of Defense, 2014.

There are generally two types of subsystem manufacturing and sustainment models. The first is the restricted-proprietary model, when either the data rights or deliverables are insufficient to allow for competitive manufacturing and sustainment of the specific subsystem. In such a model, a small amount of proprietary technology can possibly restrict the competition of a much larger subsystem. Conversely, the open-competitive model is applicable when both the data rights and deliverables are adequate to allow for competitive manufacturing and sustainment of the subsystem.[32] When selecting a sustainment model, it is important to verify the technical segregability of proprietary elements from DoD-funded, open-competitive elements. Programs can reap tremendous benefits from transitioning system modules to an open-competitive model. While complex systems can often be disaggregated into subsystems, this is most easily accomplished in a MOSA-designed system.[33]

Once the sustainment model is selected, the next level in granularity is to define the design documentation approach, which determines the specific data rights requirements for each module. Potential options include a *detailed design approach* (government controlled or source controlled), *incomplete design approach* (source controlled), and the *open systems design approach*. While the first two approaches are valid under appropriate circumstances, the open systems approach is very promising for managing limited and restricted data rights in space acquisition. In the open systems design approach, DoD owns and maintains the performance and interface requirements and consequently acquires subsystem solutions from multiple contractors. While they own the technical baseline for the architecture, DoD does not have visibility or configuration control of the contractor's detailed design of the modules. However, a key advantage of the open systems design approach is that the performance and interface requirements are included in FFF data, to which DoD has unrestricted access from a data rights standpoint. If a system is MOSA-designed, the integrator can plug proprietary modules into the greater system using FFF data and sustain accordingly through unrestricted OMIT data. The open systems design approach is valuable when multiple proprietary solutions are available for frequent competition and is particularly relevant for technology with a short obsolescence time line. If these conditions do not exist, this approach may not be economically feasible for DoD.[34]

As explained above, a primary goal of MOSA is the ability to plug and unplug various modules developed by different contractors, with the assumption that FFF data will suffice for integration. But there are instances, based on anecdotal evidence, where DoD will need information more detailed than FFF data for integration, which can potentially reside in subsystems for which the government owns only limited and restricted rights.[35] As a result of

[32] U.S. Army, 2015, p. 34.

[33] U.S. Department of Defense, 2014.

[34] U.S. Army, 2015, p. 35.

[35] Council of Defense and Space Industry Associations (CODSIA), Response to DFARS Case 2012-D022, September 30, 2016, p. 5.

this issue, Section 815 of the FY 2012 National Defense Authorization Act (NDAA) introduced the concept of “segregation and reintegration,” which would expand DoD access to limited and restricted rights beyond FFF data for the purposes of “segregation and reintegration.” The law did not explicitly define segregation and reintegration data, which became the responsibility of DoD to define in the DFARS.[36] While a codified definition of “separation and reintegration” has yet to be agreed upon,[37] the most current update to this clause is in Section 809 of the FY 2017 NDAA, which limited the scope of segregation and reintegration data to information “pertaining to an interface.” At present, DoD is in the process of implementing the provisions of this act into DFARS. When integrated into DFARS, separation and reintegration data will hopefully enable further use of MOSA throughout DoD.

Data Rights Practice Within Space Acquisition

While all DoD procurements share some similarities with respect to data rights, there are specific elements unique to space acquisition programs. The nature of acquiring spacecraft requires a very development-focused program due to the ground testing necessary before launching into orbit. It is difficult to conduct maintenance and repair activities in low earth orbit and nearly impossible with current capability in medium earth orbit and geostationary earth orbit. So, given the high cost of the procurement and launch of space vehicles, the value of independent verification and validation prior to launch is crucial for avoiding hardware and software failure. However, the release of information to contractors for independent verification and validation activities is restricted under limited and restricted data rights. For this reason, space acquisition programs typically need to acquire greater than limited or restricted rights.[38] Some program offices have mitigated this issue through implementing *enabling* clauses, which require the contractors with proprietary data to provide support contractors with access to said data.[39]

In MOSA-designed space systems, industry has recently expressed concerns about how deep into a system an open architecture will be implemented.[40] The deeper into a contractor’s proprietary “black box” solution the government specifies interface requirements, the more nervous they will feel about disclosing their intellectual property. If a prime contractor needs to access the source code during integration, for instance, and consequently requires access to

---

[36] DFARS would define “segregation and reintegration” data as the physical, logical, or operational interface or similar functional interrelationship between the items or processes and may include, but would not typically require, detailed manufacturing or process data or computer software source code.

[37] DoD published a proposed rule in the Federal Register in June 2016, which accommodated industry’s concerns about Section 815 of the FY 2012 NDAA.

[38] SMC, 2011, p. 14.

[39] SMC, 2015, p. 21.

[40] Boeing staff, discussion on MOSA, agile, and rapid prototyping, El Segundo, Calif., June 29, 2017.

proprietary data, subtier suppliers can become threatened. Therefore, it is vital for the prime integrator and subtier supplier to have a transparent relationship involving the sharing of the maximum practical amount of technical data, which can allow the framework provider to finish the product successfully. A consortium approach is a practical way to find consensus among all suppliers in the industrial base so such situations are alleviated in the future.

Appendix B. Acoustic-Rapid Commercial Off-the-Shelf Insertion Program Case Study

The Navy's Acoustic-Rapid COTS Insertion (A-RCI) program is an example of a program that employs incremental fielding to support evolving operational needs. A-RCI is an upgradeable COTS hardware-based sonar processing system developed to replace Cold War–era sonar processing hardware. It is currently installed on all in-service U.S. Navy submarines and therefore requires an enterprise entity to develop its operational requirements with all ship classes in mind. It is managed in an incremental program with scheduled, periodic Joint Capability Integration and Development System (JCIDS) reviews and periodic integrated developmental and operational tests. While individual software algorithms are developed competitively to develop innovative solutions to operational issues, the Navy has a long-term relationship with a single integration vendor. Similar principles are used in the Navy's Aegis program, and these lessons are being extended to a broader suite of Navy combat systems through the Submarine Warfare Federated Tactical Systems effort.[1]

Program History

In the 1990s, a reduction in U.S. submarine acoustic advantages over various adversaries combined with budget pressures led to the Navy's drive to use COTS technology and MOSA to revolutionize their sonar processing capability. A-RCI system architecture was designed to consist of COTS hardware and separately developed, custom software applications and algorithms that performed sonar processing and data presentation to operators. A-RCI also extensively uses middleware to separate individual software applications and algorithms from the hardware and software infrastructure to enable interchangeability.[2]

From a fielding standpoint, A-RCI's hardware and software interchangeability is manifest via scheduled, parallel software and hardware updates delivered to ships on their own time lines. Software updates, known as advanced processing builds (APB), were initially developed every year. Hardware updates, called technology insertions (TI), were developed every two years.[3] However, due to the strictness of submarine maintenance availability time lines, each submarine

---

[1] Paul DeLuca et al., *Assessing Aegis Program Transition to an Open-Architecture Model*, Santa Monica, Calif.: RAND Corporation, RR-161, 2013; Jessie Riposo et al., *CANES Contracting Strategies for Full Deployment*, Santa Monica, Calif.: RAND Corporation, TR-993, 2013; U.S. Department of Defense, "AN/BQQ-10 Acoustic Rapid Commercial Off-the-Shelf Insertion (A-RCI) Sonar," *Director, Operational Test and Evaluation FY2016 Annual Report,* December 2016.

[2] Riposo et al., 2013, p. 53.

[3] DeLuca et al., 2013, p. 80.

cannot receive each APB and TI as soon as it is released. Additionally, some changes to sonar processing hardware or software were significant enough to require additional crew training or have other operational effects. This resulted in a reconsideration of the initial APB/TI time line. The A-RCI program has settled on two separate two-year cycles, one for hardware and one for software. Hardware is delivered to submarines in even years, and software is delivered in odd years.[4] This schedule of out-of-phase hardware and software cycles emphasizes the importance of middleware and modularity of software and hardware: A-RCI software must be compatible with multiple versions of A-RCI hardware. As of 2006, 23 software/hardware configurations of the A-RCI system existed across the fleet of over 70 submarines.[5]

At the onset of the program, desired full capability of the A-RCI system was unknown, because the program's goal was to create an evolving system with capability improvements delivered as fast as commercial technology became available. That said, the first increment of A-RCI was limited to hardware and software necessary to operate a towed array sonar system carried by Los Angeles Class submarines. Incrementally, A-RCI's scope increased to encompass all sonar processing hardware and software on board all submarine classes.[6] By 2004, it was estimated to have improved towed array sonar performance by over seven times and reduced cost per processing operation by over 60 times.[7] This incremental system development and fielding approach has grown popular within the Navy for shipboard combat system applications and is used for the Aegis combat system and other submarine combat systems.

Program Management Approach

Separate hardware and software development and procurement divides A-RCI program management into two efforts. As is true with other MOSA systems, an entity not responsible for development of every subcomponent of the A-RCI system performs system integration.

Software Development and Procurement

Software development and procurement is the backbone of the A-RCI program. Software development follows a four-step process: identification of algorithms, independent algorithm testing by a government agent, laboratory algorithm testing on the current TI processor baseline,[8] and at-sea testing. Algorithm identification involves evaluation of software applications available

---

[4] Rather than each one as soon as it is available.

[5] Michael Boudreau, "Acoustic Rapid COTS Insertion: A Case Study in Spiral Development," NPS-PM-06-041, October 30, 2006.

[6] Riposo et al., 2013, p. 53.

[7] William Johnson, "The A-RCI Process—Leadership and Management Principles," *Naval Engineers Journal*, Vol. 116, No. 4, 2004, pp. 99–105.

[8] The software is also tested on prior TIs but as part of the later testing cycles of the program and not in this step of this process.

within the broad research industrial base with sources ranging from defense research entities to small businesses to academic institutions. This survey is carried out by the program's technical community but is most often driven by operational requirements identified by the Submarine Tactical Requirements Group (STRG). The STRG is a group of senior submarine community leaders who represent the user community and identify capability that must be developed in a given APB. That said, the process of scoping a given APB involves give and take. If the technical community identifies a software algorithm or application that is mature and could provide operational benefit, it may be considered for inclusion within an APB, even if it is not part of the STRG-identified need.[9]

The second step tests software applications in simulated data environments to determine which ones are mature enough for further testing in more operational environments. Once an application reaches the third step of testing, an algorithm is planned for inclusion in the next APB, and formal documentation for its inclusion in the APB is drafted. Following completion of the final step of at-sea testing, an algorithm is qualified for shipboard use and may be included in a future APB, if it is selected for final inclusion.[10] Software is procured on government contract and is provided as government-furnished equipment to the system integration vendor for system integration.[11]

Hardware Procurement and System Integration

Since A-RCI utilizes COTS hardware, the hardware procurement aspect of the A-RCI program model is relatively straightforward. The government is responsible for developing specifications for what hardware will go into a given APB, and provides these specifications to the system integration contractor[12] for procurement. The integration contractor is responsible for assembling hardware that it procures to government specifications and integrates hardware with software delivered to it.

Interactions with Joint Capability Integration and Development System and Developmental and Operational Test

A-RCI is formally established as an incremental program in JCIDS, which affords it documentation and testing flexibility. When first reviewed in JCIDS, each of A-RCI's increments was required to generate its own set of mandatory JCIDS documents. However, over time, A-RCI was afforded the flexibility to cover one hardware increment and two software increments within a single Capability Development Document (CDD) and Capability Production

---

[9] Boudreau, 2006.

[10] Boudreau, 2006.

[11] Riposo et al., 2013, p. 53.

[12] Currently, Lockheed Martin.

Document.[13] This results in a scheduled, yearly JCIDS review, which streamlines A-RCI's progression through the JCIDS process.[14]

A-RCI also takes advantage of its unique structure to streamline developmental and operational testing requirements. The Navy produces mandatory test plan documents in parallel with the four-step software application development and testing process. Operational testing completion is considered in Director, Operational Test & Evaluation assessments of a system's operational suitability.[15]

Insights from the A-RCI Case Study

The A-RCI program provides some insights on the successful implementation of an incremental, MOSA system in a DoD setting:

- Implementation of an incremental program is achievable within JCIDS, provided appropriate planning. This includes how to implement operational, developmental, and testing processes in an integrated, repeatable fashion. The establishment of calendar-driven, formalized processes to complete these activities in set time lines can be beneficial.
- A program's incremental development and fielding time line *must* have an operational focus to be truly effective. A-RCI's need to reevaluate APB/TI fielding time lines due to submarine availability schedules and crew training concerns demonstrates that developing a technology fielding plan in a vacuum is not appropriate for a DoD weapon system.
- In addition to dividing a program into increments, splitting a program's development processes into smaller parts can have significant benefits.
    - The broad use of COTS hardware can provide sufficient capability at a reduced cost, but caution is warranted. Use of off-the-shelf hardware reduces program development complexity but does not relieve the need for careful integration engineering and testing to determine whether the performance and vulnerabilities are satisfactory.
    - The competitive development of individual software applications, even in a niche market like submarine sonar processing algorithms, can drive revolutionary developments and innovations.
    - In addition to using broadly adopted commercial interface standards, the widespread use of middleware to separate hardware and software infrastructure from individual software applications can improve module interchangeability.
    - While competition can be a successful contracting technique, a long-term relationship with a single integration vendor can also be a successful approach.

---

[13] Currently, Lockheed Martin.

[14] Boudreau, 2006.

[15] U.S. Department of Defense, 2016.Additionally, if a system's developmental testing is performed in an operational environment, it may be considered for combined developmental and operational testing per 2008 Director Operational Test & Evaluation guidance. See U.S. Department of Defense, *Director, Operational Test and Evaluation Test and Evaluation Master Plan Guidebook*, Version 3.1, January 19, 2017, p. 101.

Appendix C. Agile Acquisition

This appendix discusses the characteristic attributes of agile acquisition and describes a high-level overview of how an agile approach works.[1]

Characteristic Attributes of Agile Acquisition

Agile acquisition arose (most recently) from the application of Lean Six Sigma methods to software development. In recent years, it is increasingly being applied to hardware development as well. While a wide range of agile practices exist, they tend to share a number of characteristics. We describe those characteristics below.

Incremental Approach

Project members break development tasks down into pieces and develop them one piece at a time. Each increment involves the negotiation of acceptance criteria for the product of the next round of development; a steady, sustained development effort; testing during development, including, where possible, automated testing to allow software programmers to focus on development itself; a planned, short capstone period of testing and integration to verify readiness for the user; and then immediate delivery of usable software to the user. This approach contrasts with a waterfall approach, which tends to complete larger chunks of system development before development and user testing. But we should not overstate the distinction. In practice, a great deal of interim testing occurs under a waterfall development approach. Agile simply places greater emphasis on breaking a development task down and adjusting direction as development proceeds.

Flexible Approach

Adjusting direction is much more important to an agile approach than to the traditional approach. The traditional approach invests substantial effort in developing detailed requirements before serious development effort begins. It then seeks to hold the initial requirements as stable as possible. An agile approach follows a different philosophy. It believes that a user cannot know what a system should actually do until the user has experience with it. Growing experience generates information that the user can apply to refine requirements for the system under development.

Even if a user could fully express requirements in advance, in any dynamic environment, the user's needs change. For Air Force Space Command (AFSPC), the threats to its systems in

[1] In this appendix, we often substitute the word "agile" for "agile acquisition" when the context allows.

space change; recognition of this fact is a central argument behind the SEV. Technologies change; an incremental approach designs in capacity to insert new capabilities easily when they become available. Moreover, the development process reveals information about the feasibility of achieving certain goals with acceptable costs and schedules.[2]

In practice, of course, software developers applying traditional methods face external demands for engineering changes throughout any complex development and discover in the course of development that trades need to be made among goals as difficulties reveal themselves. The difference lies in how agile and traditional developers (and their supervisors) think about change in priorities and plan for it. Which approach is more useful is bound to differ from one circumstance to another.

User Input

Agile acquisition depends heavily on user input during development to guide the course of a development. That input is most likely to be useful if developers give users prototype versions of their product that the users can test in as realistic a setting as possible. As a result, agile acquisitions seek to structure increments that each yield a product that can give a user some tangible, useful new capability, even if the capability falls far short of what the user ultimately needs.

Ideally, just as ongoing testing during development gives developers useful immediate feedback, having as much access to the user as possible during development does the same thing. Traditional software developers want the user or a surrogate present when requirements are determined at the beginning and when something nearing a final product is ready for a final operational test. But traditional bureaucracies deliberately separate developers from users so each can spend their time doing what they do best. The most aggressive forms of agile acquisition simply reject this logic. They demand active participation of real users in the assessment of the product of each increment over the course of a development.

In some cases, they demand the continuous presence of real users onsite with the developers over the course of the entire development. Development Operations (DevOps), which seeks to integrate the cultures of developers and operations throughout the course of frequent product upgrades, currently displays perhaps the most extreme version of this perspective. These demands pose challenges for traditional bureaucracies. As a practical matter, agile acquisition is less likely to succeed in bureaucracies that are not willing or able to break down such walls between operational users and developers. Advocates of DevOps report that such walls pose their greatest challenges.[3]

---

[2] The Global Positioning System (GPS) is trying to do this with the *tech inject* points on its GPS III Space Vehicle 11-32 acquisition strategy.

[3] Valerie Silverthorne, "Change the Culture to Fix the Tragedy That DevOps Can Be," TechTarget.com, 2017.

Tailoring

Part of agile acquisition's legacy as a descendant of lean production is a persistent emphasis on eliminating waste: what Japanese practitioners of lean methods call *muda*.[4] *Muda* is any expenditure of resources, including calendar time, that does not add value to the final user of a product. Agile developers want to deliver useful capability to users as rapidly as possible. And they do not want to spend time or effort on any capabilities that users do not want. They do not want to gold-plate a product with capabilities users will not use. And in particular, they do not want to create documentation that users do not need to take advantage of a new capability.

This emphasis on what current users need to be successful draws a sharp distinction between the agile and traditional development of software. When they address acquisition logistics, traditional developers put considerable attention into what future users might need. They also emphasize the need for a record of how developers made decisions today to help a different set of future developers upgrade the software developed today. Agile advocates counter that skillful developers can (and should) write code or design hardware that approaches self-documentation and that, if a product is user-friendly, detailed user manuals are not necessary or even desired.

This distinction in point of view comes to a head in a federal setting over the definition of deliverables on a Contract Data Requirement List. In practice, there is usually an opportunity to learn from both perspectives. How to define what a user needs will differ from one context to another. We can probably improve on the extreme approach offered by purists of either persuasion.

Delegation

Agile seeks to clarify something like a "commander's intent" to convey the central goal of any particular development task and then delegate the responsibility for executing that intent to highly skilled developers who can use their local knowledge of development activities to shape the development effort as it proceeds. In its most extreme version, agile development teams self-organize and manage their daily activities without formal project management. Even less extreme versions give individual developers more authority. This is feasible only when a team has more highly skilled participants than those in a traditional waterfall setting.

Periodically during the execution of an incremental development, developers revisit code to improve its ability to execute the task at hand. When authority to do this is delegated, it must be delegated to developers who understand which changes can occur without negatively affecting the code that other team members are writing in parallel.

---

[4] Joel Bradbury, "Lean: Muda, Mura, Muri," *Quality Digest*, March 8, 2017.

How a Typical Agile Acquisition Proceeds

When *agile software development* emerged as a formal term of art in 2001, an almost religious divide arose between (1) traditional software engineers who thought this new approach threatened the discipline created by plan-driven software development and (2) young progressives impatient with the traditional view that it was possible to create a complete plan for a software program before coding even started.[5] Effective caricatures of the alternatives appeared in textbook-like descriptions of "pure" versions of the two approaches as polar opposites. But experienced practitioners knew that the so-called linear approach that traditional engineering emphasized included a great deal of iteration and learning as a development moved steadily from start to finish. And the advocates of the new approach knew very well that close discipline was required to use iteration and learning to yield a useful product. Practical experience with the new approach has demonstrated that, when properly applied, it is every bit as disciplined as the traditional approach but with a different set of dynamics, risks, and requirements alignments. Agile methods appear to be getting increasing acceptance as additional tools that any software engineer should consider using in any software development.

That said, agile acquisition is quite different from the plan-driven approach that DoD has grown accustomed to when developing requirements, adapting requirements as development continues, monitoring progress during development, and testing a final product. Many variations exist, and the agile approach continues to evolve in DoD as experience with it accumulates. The description below offers a high-level view of what DoD users of this new approach should expect to see if the approach is working well.[6] Table C.1 summarizes key terms used in the description.

As an agile software development starts, a team of relevant stakeholders and developers gathers to identify, at a high level, the operational capabilities that users want from the new software. This team effort describes these capabilities in users' language, in the agile lexicon, to define the *theme* for the software development. This theme in effect describes the first version of the requirements for this development. Developers estimate at a high level how much effort will

---

[5] The clarion call of the "Agile Manifesto" that introduced the term to a broader public in 2001 contributed to that sense of a revolution. See Kent Beck et al., "Manifesto for Agile Software Development," 2001.

[6] This description draws heavily on the especially clear description offered in Peter Modigliani and Su Chang, *Defense Agile Acquisition Guide: Tailoring DoD IT Acquisition Program Structures and Processes to Rapidly Deliver Capabilities*, MITRE Corporation, McLean, Va., March 2014, pp. 2–5. It also benefits from descriptions in Barry Boehm and Richard Turner, *Balancing Agility and Discipline: A Guide for the Perplexed,* Boston: Addison-Wesley, 2004; Mary Ann Lapham et al., *Agile Methods: Selected DoD Management and Acquisition Concerns*, Technical Note CMU/SEI-2011-TN-002, Carnegie Mellon University, Software Engineering Institute, Pittsburgh, Pa., October 2011; and Bertrand Meyer, *Agile! The Good, the Hype, and the Ugly*, Zürich, Switzerland: Springer, 2014.

Table C.1. Summary of Terms Used in the Description of Agile Software Development

	Element of Development			
	Program	Release	Sprint	Scrum
Place in development	Spans the development, end to end	Subelement of the program	Subelement of a release	Subelement of a sprint
Definition of user capability requirement	Theme	Epic	User stories, one for each capability	User stories in play
Product developed	Final release to user	Incremental release to user	Demonstration to users, stakeholders	Report at scrum meeting
Term of development	As long as it takes	Four to six months	About a month	One day

be required to achieve each capability. Considering the relative importance of these capabilities to users and the relative level of effort required to achieve them, the team rank orders the capabilities so that, with a given amount of effort, developers can deliver the most important capability first, then the next most important, and so on until stakeholders and developers agree that no further development is worth the effort.

Based on this initial planning, the team breaks the total program into a set of releases that the development team thinks it can deliver as operational software to users at fixed intervals of, say, four to six months each. The team then focuses on the first release and agrees on acceptance criteria: what is commonly called the *definition of done* in the agile lexicon. This definition focuses on specific capabilities that users can verify that they have when they receive the initial release. The definition can go beyond this to define details about code completion, the level and types of testing, and (just enough) documentation expected before the release to operational users occurs. In the agile lexicon, the capabilities and acceptance criteria for the capabilities, stated in users' language, together define the *epic* for this release. This epic refines the requirement relevant to the first release that was initially embodied in the theme for the development.

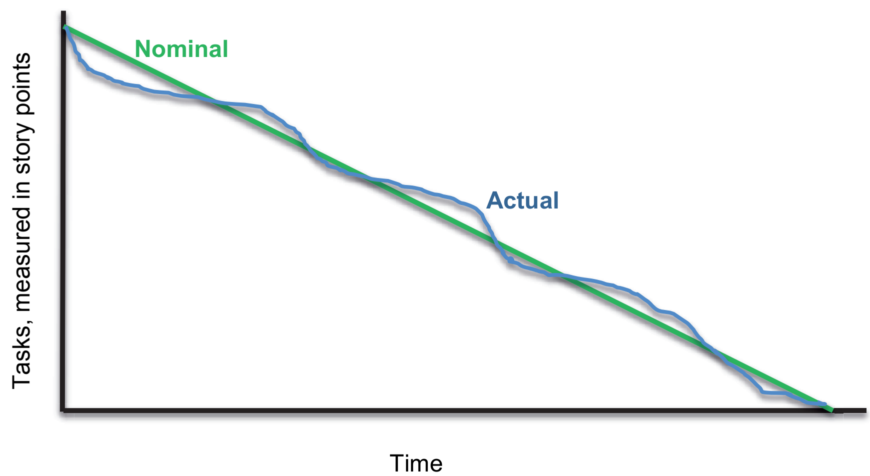
When this first release has been defined, the development team then turns its attention to what it plans to do during the first sprint, which might last for a month. It now describes the user capabilities it plans to develop in this sprint in terms of one or more *user stories*, which effectively refine the requirement relevant to the first sprint that was initially embodied in the epic for the first release. During this sprint, for each user story addressed in the sprint, developers complete a well-defined development cycle that includes the following stages:

- initial planning
- a design for how to get from the user story to code
- development of code, to include tracking coding tasks
- continuous integration, at least daily
- automated and integrated testing
- demonstration of functionality to users and stakeholders.

Following the demonstration(s), the sprint does not actually release any incremental operational capability to users. As in the epic, the user story includes a definition of done, which must address capabilities that users will receive and can include additional detail on code completion, the level and types of testing, and (just enough) documentation expected at the completion of the sprint. This definition of done is locked for the duration of the sprint.

During the sprint, the development team uses a burn-down chart to track progress. Figure C.1 displays time on the horizontal axis and cumulative completion of tasks on the vertical access. Tasks are entered in the vertical dimension in terms of the *story points* expected to be needed to complete each task. With potential support from third-party experts, each team uses its professional judgment at the opening of each sprint to assign story points to tasks. They assign more story points to more complex tasks. The burn-down chart displays a planned, nominal path, from 100 percent of the *backlog* of work for the sprint to (ideally) 0 percent of the backlog of work for the sprint, over the fixed period established for the sprint. The chart is updated daily to display the actual burndown of this backlog.

Figure C.1. Notional Burn-Down Chart



As the development continues, the team creates and tracks progress against analogous burn-down charts for each sprint, each release, and ultimately the program as a whole.

As the first sprint approaches completion, the development team begins planning for the second sprint. If the team achieves more or less than expected during the first sprint, it then uses an updated version of the backlog for the release to plan the second sprint. Based on what it learned from the first sprint, it may add or subtract user capabilities to address in the program, adjust the story points associated with them, or adjust the order in which it will address additional capabilities. Analogous sprints continue in series, potentially followed by a period at the end of the release devoted to additional integration and testing activities. When these are complete, the development team delivers a functional release of software to users for application in an operational setting. Users provide feedback that the development team can use to move forward.

Analogous releases, each comprising a set of sprints and final integration and testing, proceed in series until the software envisioned in the program is entirely in users' hands. Two points are worth noting: (1) This software could look quite different from what the initial theme envisioned. Agile development by design expects learning, by users and developers, through the course of a development. With this in mind, it approaches the refinement of requirements incrementally, reaching complete details on requirements only for the next set of tasks at hand. (2) This incremental approach continues until the program is complete. Users and developers, informed by resources available during any period, decide together when to stop. In some cases, development may simply stop, leaving the developers to move on to other tasks. In others, the environment in which development occurs is dynamic enough so that it is worth pursuing repeated new releases to address surprises in user priorities, technologies available, or the progress of development itself.

As long as development continues, active communication occurs among users, developers, and other stakeholders. Developers conduct a brief daily scrum meeting where all members of a team can report what they have just done, what they plan to do next, and what obstacles they anticipate as they go forward. At the completion of a sprint, members of the development team hold a *hotwash* to assess how well the sprint went and adjust their approach in the future to address problems identified. Users and stakeholders receive demonstrations at the end of each sprint. Users receive operational releases at the end of each release. During sprints, the development team may consult with relevant third-party subject-matter experts and representatives of users and stakeholders. An agile coach may be present to monitor and mentor team members' activities during a sprint and adjust the approach being used. An agile advocate may monitor activities to help users, developers, and other stakeholders place the activities in a broader context.

An office overseeing such an effort needs a mechanism for tracking progress and asking when the approach to date should be adjusted. A DoD system program office would typically use an earned-value management (EVM) system to track the interim goals achieved and resources expended against a schedule of each defined at the beginning of the development. A burn-down chart offers an analog to such an approach during a sprint. But in the beginning, an agile approach only outlines any plan for upcoming releases or the program as a whole. As development proceeds, the requirements for the program could shift dramatically, invalidating any EVM system schedules projected at the beginning of the development. Agile acquisition prefers less formal oversight that monitors how well developers meet the user goals stated at the beginning of each sprint, how well they deploy the resources available to them, and how well they respond through the development as user goals shift. Each decision to move forward with a developer essentially uses an informal assessment of past performance. If the developers meet the expectations of the government oversight office, development continues. If not and resolution cannot be achieved, it may be appropriate to take an unscheduled off ramp and change developers. Any contract should define this option to keep its cost as low as possible. Easy exit can be an integral part of effective oversight.

Appendix D. Observations on the Use of Agile Methods for Joint Space Operations Center Mission System Increment 2 Program

Various assessments of the Joint Space Operations Center Mission System (JMS) Increment 2 program, such as the one conducted by SMC in 2015,[1] concluded that it was not ready to employ agile. We concur with such assessments, but a judgment that the Air Force was not ready to use agile acquisition in the JMS Increment 2 program should not be read as a determination that JMS Increment 2 was *inherently* a poor candidate for the application of agile methods. Rather, the Air Force had not prepared to use agile methods effectively in this acquisition and so should not have been surprised by many of the difficulties that the acquisition encountered. Indeed, the number one finding of the current state of knowledge is that if an organization has not prepared properly to use agile methods, it should not attempt to use them.

This appendix uses criteria related to system applicability and government level of commitment that are relevant to the successful application of agile methods to observe what the Air Force could have done to use agile methods more effectively.[2] As we go forward, we make no judgment about whether JMS Increment 2 was inherently suited to use agile methods.

System's Applicability

System Criticality

Plan-driven methods can be more suitable for mission critical tasks, but agile methods have been successfully used in mission-critical programs. The current definition of JMS Increment 2 apparently allows little flexibility in what the increment would deliver or when operational elements of the increment would be delivered. But, in principle, a hybrid approach could be defined to achieve, at a minimum, the capabilities of the legacy systems it will replace in the first increment, which might best be pursued in a plan-driven approach. Once that is achieved, a more agile approach might be used to build on that capability as technology and user priorities change over time (more on that below). Once the minimum level of performance is achieved, the mission criticality of additional increments drops. Such a hybrid strategy could have made JMS Increment 2 more amenable to agile methods.

---

[1] The assessment appears in Vale Sather, Michelle Dobard-Anderson, and David Naiditch, "JMS Inc. 2 Agile Fit Check," Version 1.2, SMC/EN, Los Angeles Air Force Base, Los Angeles, Calif., November 16, 2015.

[2] We drew these criteria from Sather, Dobard-Anderson, and Naiditch, 2015. We omit criteria on the "contractor's level of commitment" to avoid including contractor proprietary information. A review of contractors would yield analogous findings.

Dynamism

JMS Increment 2 requirements were conceived in a plan-driven setting. Commercial users of agile methods are discovering that these methods open up new opportunities to respond rapidly to changes in their external environments and so are especially useful in dynamic environments. The SEV is a response to the recognition that U.S. space capabilities are now contested. Put another way, the space environment is more dynamic than the environment envisioned when the JMS requirements were developed. Requirements designed to allow rapid responses to surprises in threats, technological capabilities, or the performance of fielded software might necessitate more than one requirement change per month. The degree of dynamism relevant to JMS Increment 2 probably depends on the level of dynamics in the environment in which it will operate. The SEV has raised that level of dynamics. That change could well have made JMS Increment 2 more amenable to agile methods.

Requirements Clarity

The discussion of dynamism above applies here as well. The requirements determined in the past are apparently clearly written and easily understandable. Any new requirements that emerge in a dynamic environment, by definition, are not. The relative desirability of an agile approach would not derive from its ability to correct past misunderstandings but from its ability to support the pursuit of new requirements in close to real time and allow users to identify what they are. That is, agile does not depend on clearly written and easily understandable requirements and so does not have to wait for them to be developed. The savings in calendar time that this allows could have made JMS Increment 2 more amenable to agile methods.

User Time Lines

The discussion of dynamism above applies here as well, almost exactly. In the contested environment that SEV envisions, user time lines should reflect a likely demand for more than annual upgrades, thereby making JMS Increment 2 more amenable to agile methods.

Test Environment

We observe that the operational system, testing processes, and information assurance certification processes are currently not designed to allow a more agile approach to testing, but systems designed with agile approaches in mind have been able to address issues of these kinds. So it is possible that although the current test environment is not ready for agile method, appropriate preparation for an agile development could have made JMS Increment 2 more amenable to agile methods.

Program Scope

Agile methods have proven easier to apply in federal programs that are smaller and simpler. But even where a plan-driven approach integrates all parts of a complex development, with an appropriate design architecture and well-defined interfaces, agile methods are being applied to the development of individual components within such an architecture. And more and more capable tools are allowing the application of an approach like a "scrum of scrums" that can build a hierarchy of team efforts to integrate more complex programs. The application of such new approaches could have made JMS Increment 2 more amenable to agile methods.

Government's Level of Commitment

Leadership Support

The essence of an agile approach is to tailor its application to each program. That requires more than the toleration of agile methods. Rather, it requires a commitment to determine the right set of development methods for the program.[3] Best practice often places an agile advocate—a third-party expert on agile methods—in each headquarters where leaders need to pave the way for an agile approach. Such preparation could have made JMS Increment 2 more amenable to agile methods.

Contracting Strategy

Choosing the right contract strategy for a program is an integral part of tailoring the approach to development. Building a flexible contract that rewards speed and attention to user priorities could have made JMS Increment 2 more amenable to agile methods.

Government Expertise

Agile methods are first and foremost about human capital: the skills of the people who write requirements, the developers, the representatives of users who work with them, the contract personnel who certify contract changes, the testers, and the staff that oversees and integrates all of this effort. Agile methods work best when all of these people have prior experience with agile methods and, ideally, prior experience working together to implement agile methods. Agile training can substitute for some of this, but prior experience is preferred. Agile advocates and agile coaches—hands-on mentors for the specific individuals involved in the process—are especially helpful when the people implementing agile methods are still learning how to do it. Building skills and hands-on support for the people building these skills could have made JMS Increment 2 more amenable to agile methods.

[3] One could argue that this is true for all acquisition programs; however, in the case of JMS, it was particularly important because this was the first time agile acquisition was being applied at SMC, and agile invoked very different practices than the traditional acquisition approach.

Level of Oversight

Other assessments indicate that the program was not designed to use agile methods. An agile approach allows users or a surrogate for users (often called a product owner) to work hand in hand with a warranted contracting officer to choose and validate program changes in near real time. The goal is to prevent calendar time from passing between the time when the information needed to make a decision becomes available and the decision is made and then executed. Agile methods seek to do this by working across functions rather than relying on vertical stovepipes to move information and support decisions. Empowering users and contracting officers to implement contracting changes as soon as the users identify the need for changes could have made JMS Increment 2 more amenable to agile methods.

Collaboration

Agile methods do not require a continuous government presence at a development site. Experience indicates that a continuous presence can be desirable, and the growing success of the DevOps version of agile acquisition highlights this desirability. But it is not necessary. Hybrids can accommodate a smaller government presence. Ultimately, this limit on collaboration results from specific Air Force decisions not to prepare as fully for an agile approach as other successful efforts have in the past. A greater willingness to prepare could have made JMS Increment 2 more amenable to agile methods.

User Involvement

Agile methods do not work without effective participation by users or a suitable surrogate, such as a product owner. Without users, there is no one to drive successive increments of agile development. Getting effective user participation is hard. Ideally, users should be trained for this duty. They should be relieved of other responsibilities so that they can participate effectively. Turnover in user participants should be limited. All of these things are probably easier for a program like the JMS Increment 2 to achieve, because all users lie within AFSPC, which also has substantial influence in the acquisition process. Appropriate preparation could have made JMS Increment 2 more amenable to agile methods.

Summary

Agile acquisition is always implemented in a hybrid form in DoD. Agile acquisition is precisely about tailoring a development effort to local circumstances. On the basis of what we know, we cannot say what the ideal hybrid would be for JMS Increment 2 or whether, in the end, agile acquisition should have played any role in the program. It might well be that the criticality of JMS Increment 2 and the inherent stability and clarity of its requirements made it a poor candidate for agile acquisition. However, had the program properly prepared to use agile methods, they may well have added considerable value to the program in terms of quicker identification and resolution of failures, better and lower cost solutions to realize stated requirements, and maybe even an opportunity to learn from users that they needed capabilities different from those stated with such clarity in the program requirements. Agile acquisition has provided such benefits in other DoD settings where it has been effectively implemented. The challenge for SMC and AFSPC is whether they are prepared to accept the short-term pain and cost required to be ready enough in the future to benefit from agile methods in other programs. If they are not, they will face a situation where they cannot in fact get there from here.

Appendix E. Formal Change Management

The chapters in the main body of the report provide information about three enablers that AFSPC and SMC could use to adjust acquisition policy to support the new SEV. Implementation of some of these changes has begun. Other changes still need more time to be defined well enough to begin formal implementation. If the Air Force ultimately decides to accept all of these kinds of changes, they will not occur overnight. Initial experience will lead to potentially significant adjustments. And successful implementation of the basics across the entire Air Force is likely to take a long time.

This appendix summarizes what has been learned from efforts to achieve similarly significant changes in large, complex American public- and private-sector organizations over the last three decades.[1]

Recent Formal Change-Management Perspectives in the United States

The quality management movement that came to the United States from Japan in the 1980s and has grown and prospered ever since brought with it an approach to formal change management that large, complex public- and private-sector organizations have used successfully to implement a wide range of changes.[2] The approach has been studied and documented in publications aimed at academic, technical business, and broader popular audiences.[3] John Kotter became especially well known in the 1990s for the conclusions he drew from an extensive practice helping large corporations implement change.[4] This appendix draws on Kotter (1996) and others writing in the tradition of the quality management movement to summarize the elements of formal change management that they have emphasized. It argues that there are few substantive differences among these writers. Rather, they tend to emphasize different aspects of

[1] It is adapted from Camm et al., *Charting the Course for a New Air Force Inspection System*, Santa Monica, Calif.: RAND Corporation, TR-1291-AF, 2013, Chapter 7.

[2] For more on the link between the quality management movement and current American thinking about formal change management, see Frank Camm, "Adapting Best Commercial Practices to Defense," in Stuart E. Johnson, Martin C. Libicki, and Gregory F. Treverton, eds., *New Challenges, New Tools for Defense Decisionmaking*, Santa Monica, Calif.: RAND Corporation, MR-1576-RC, 2003.

[3] For literature reviews of academic and technical business publications likely to be useful in an Air Force setting, see Gail Zellman et al., "Implementing Policy Change in Large Organizations," in Bernard D. Rostker et al., *Sexual Orientation and U.S. Military Personnel Policy: Options and Assessment*, Santa Monica, Calif.: RAND Corporation, 1993, MR-323-OSD, pp. 368–394; and Cynthia R. Cook, Laura Werber Castaneda, and Abigail Haddad, "Implementation," in Bernard D. Rostker et al., *Sexual Orientation and U.S. Military Personnel Policy: An Update of RAND's 1993 Study*, Santa Monica, Calif.: RAND Corporation, MG-1056, 2010, pp. 371–388.

[4] John P. Kotter, *Leading Change*, Boston: Harvard Business School Press, 1996.

formal change management. Readers with an interest in a particular element of the change process should consult the authors who give this element the greatest emphasis.

We break the change process into three elements: planning, execution, and sustainment. Some authors refer to these as steps or phases. But it is important to remember that continuous improvement is the order of the day for the quality movement. In such a setting, it is best to think of change as a series of increments, each building on the last by monitoring how well the previous increment worked and adjusting the next to benefit from what can be learned from the empirical performance of the last. In such a setting, at any point in time, a large organization will be planning some incremental changes, executing others, and sustaining or institutionalizing still others. And information will be flowing among all these efforts all the time. So even though we will describe these pieces as though they occur sequentially, in fact, the pieces often have a great deal of overlap, with feedback being shared across them all the time.

This dynamic view of change seems well suited to the Air Force context. The senior leadership at the Air Force, Major Command (MAJCOM), and wing level turns over on a fairly regular schedule. And yet, the Air Force can look out decades in the future and introduce new programs and weapon systems that no one leadership team could complete on its own watch. The Air Force is used to structuring large changes so that leaders fall in on a change effort in progress and hand it off to the next team at the end of their tour.

The management of an aircraft program offers a simple way to see this process. At any point in time, a complex aircraft program has multiple blocks at various points in their life cycles. Future blocks are still assembling technologies that can make new capabilities feasible. Other blocks not so far in the future are in development and testing. Still others approaching operational release are in production. And still others are in the fleet, where the Air Force monitors their performance and adjusts support plans as they age. That is how the formal change-management approach that we describe here works as well. In fact, the advocates of that approach could have learned a great deal from how the Air Force (and the rest of DoD) has managed weapon system programs for decades.[5]

Table E.1 summarizes the elements of the formal change management process that we will discuss. Each row contains one element. The table shows how we have grouped them into planning, execution, and sustainment activities. The columns contain representative references from the literature. We show Kotter (1996) first to reflect its dominance in this literature. We list the others in order of their publication. Camm et al. (2001), Moore et al. (2002), and Cook et al. (2010) are recent publications that explain how to apply the approach explained here in three very different defense settings. Fernandez and Rainey (2006) presents a broader view of the concerns of formal change management in public-sector organizations in general. Judson (1991)

---

[5] The Air Force Planning, Programming, Budgeting, and Execution System has a similar and completely analogous structure. The budget generated by each cycle is a discrete, incremental product of the longer-term process. At any point in time, elements of planning, programming, budgeting, and execution activities in different cycles proceed simultaneously, constantly informing one another.

and Galpin (1996) are more general practical manuals on formal change management that complement the other publications shown here. The contents of the table show which activities in our list each publication emphasizes.

Table E.1. Elements of Formal Change Management Emphasized in Recent Publications

Activity	Kotter (1996)	Judson (1991)	Galpin (1996)	Camm et al. (2001)	Moore et al. (2002)	Fernandez and Rainey (2006)	Cook et al. (2010)
Prepare							
Assess the current as-is situation		X	X				
Establish a compelling need for change	X		X	X	X	X	
Create a coalition of relevant stakeholders	X		X	X	X	X	X
Design a vision and strategy	X		X		X		X
Create a specific plan			X	X	X	X	X
Test and measure			X	X	X		
Communicate	X	X	X		X		
Execute							
Issue new policy							X
Resource					X	X	
Train				X	X		X
Measure				X	X		
Motivate	X			X	X		
Build on incremental successes	X	X		X		X	X
Communicate				X	X		X
Consolidate successes	X	X				X	X
Sustain							
Resource					X		X
Train				X	X		X
Measure			X	X	X		X
Motivate			X	X	X		X
Adjust			X	X	X		X
Communicate				X	X		
Consolidate and anchor	X	X				X	

Element 1: Prepare

Planning raises the chicken-and-egg problem that is present in much discussion of organizational change. Can significant change begin before the senior leadership supports it? If not, how does any idea for change ever come to the senior leadership's attention? Our analysis has suggested that change is the product of an iterative process in which small teams design changes, test them on a small scale, bring the empirical results to some leadership group's attention, and garner support for further design and testing efforts that yield growing interest that ultimately creates the foundation for a significant change agenda backed by the leadership. This iterative process involves considerable cycling through the activities outlined below until enough support exists to move a significant change from planning to execution activities.

Assess the Current As-Is Situation

The standard quality management approach to change documents the current as-is state of the world based on how an organization works today; posits a future *to-be* state of the world based on the goals of change; and performs a gap analysis of the difference and what is required to get from the as-is to the to-be state. Documenting the as-is state is the first step in such analysis. Experience has shown that this activity is necessary. Without it, we cannot know what the current state of an organization is or, perhaps more important, why it is in its current state. Without knowing why, we cannot effectively anticipate where resistance to change will occur and how to overcome it. That said, a common lesson of change management is that large organizations typically spend too much time on this step. An organization's as-is state is something we can know with some confidence, even if we do not know it at the beginning of a change-management process. And getting to know it better and better gives some an excuse (perhaps unconscious) not to start the much harder process of mapping what change might actually look like.

Establish a Compelling Need for Change

Change managers have learned through hard experience that significant change typically does not succeed if it is harder to go forward than to go back. Because organizations are typically deliberately structured to preserve the status quo—that is why they exist—serious change raises basic existential issues about processes and procedures that current employees have often created and committed themselves to emotionally (again, often unconsciously). Successful change requires a compelling explanation for why it is easier to move forward. That explanation should be grounded in reason, but to be effective, it should also have an emotional impact. A metaphor often used is one of a "burning platform." Change managers seek to convince those who must change their behavior for organizational change to succeed that they are on a burning platform and will go down with it if they do not find another place to stand.

Create a Coalition of Stakeholders

Individual people change organizations. Individuals must change their behavior in specific ways for organizational change to succeed. An early task of any major change effort is to determine who must change their behavior for organizational change to succeed and who represents their interests or has authority to direct their behavior in the organization. These stakeholders must sign on to the change if it is to overcome resistance to change. Again, organizations are mainly designed to preserve the status quo, and each stakeholder has a current role to play in doing that. The primary stakeholders have been around for a long time and know how to exploit the current system to their own advantage. Change often fails when stakeholders that oppose change hunker down and wait out the advocates of change.

Large change efforts benefit from formalizing coalitions at two levels. The executive level (in the Air Force, the general officer and Senior Executive Service [SES] level) provides top cover for the change effort and monitors it regularly to ensure that it continues to advance the broader goals of the organization. If the change is not achieving the gains promised, this executive group has the authority to change its direction or shut it down. The management level (in the Air Force, the O-6/GS-15 level) works the details. It ensures effective day-to-day coordination among all MAJCOM command and functional communities within the coalition. It designs the details of the change, oversees its execution, and reports on progress to the executive level. Both levels must remain intact and functional for the duration of any successful major change.

Design a Vision and Strategy

The vision and strategy summarize the need for change, the shape of the change, and the road map to achieve it in the simplest terms feasible. This should take the form of an "elevator speech" that identifies the organizational goals in play, a short list of metrics for success explicitly linked to these organizational goals, and who will be held accountable to achieve these goals. It should be designed so that its basic elements will endure over the whole life of the change. Its content is the currency of the executive coalition overseeing the change. It should appeal to both the reason and the emotions of the stakeholders who will oversee the execution of the change.

Create a Specific Plan

This is a detailed campaign plan that implements the vision and strategy. Its content is the currency of the management coalition overseeing the change. It assigns specific roles and responsibilities, sets schedules, identifies and distributes resources, and details all the actions described below that will support change as it proceeds. It is designed to be malleable. Within the strictures of the vision and strategy, it regularly adjusts in response to accumulating experience with the change. No campaign plan survives the first contact with the enemy; so too here.

Test and Measure

Empirical measurement and assessment are an integral part of the broader quality management approach. Pilot projects offer early opportunities to do this in large change efforts. They enable change managers to test hypotheses about alternative designs. They present early evidence on the validity of a change effort that managers can use to sustain higher-level support. They identify weaknesses in a setting where the negative effects are relatively easy to mitigate and learn from. Politically, pilots also offer one way to help secure ongoing support within a coalition, especially if the members of the coalition can propose pilots that address their concerns.

Such pilots should occur early: as soon as enough support exists to define a design worth testing. They should avoid the temptation simply to *demonstrate* a proposed change without giving it a rigorous test. But they should occur quickly, trading some rigor for the value of generating feedback early enough to inform ongoing decisions. Pilots can be viewed as small, incremental changes in their own right, at least for the duration of a pilot. Effective pilots follow guidelines very much like those described here for larger and longer-term incremental changes. The degree of oversight and support of the pilots is scaled to their size.

Communicate

Formal change management in large, complex organizations has many moving parts, each one serving somewhat different priorities and goals. Constant communication among all these parts—top to bottom, bottom to top, and side to side, in multiple media—ensures that participants share as common a picture of the change effort as possible as it progresses forward. By its very nature, significant change ensures that information becomes obsolete as change proceeds. Effective communication mitigates this degradation, making it easier to keep all the moving parts aligned to a common, synchronized purpose.

Element 2: Execute

Execution in effect elevates an incremental change from the status of an idea to that of a concrete change in an operational setting. The oversight efforts and resources required to facilitate such an operational change ratchet up from those involved in the planning stage. Although all the issues discussed here under execution are relevant to planning, their scale now becomes large enough to give them more explicit and detailed attention.

Issue New Policy

Those who must change their behavior for an organizational change to succeed must know what it is and what it means to them. For example, how does the change affect their roles and responsibilities? How does it affect their status within the organization? How is their behavior supposed to change? What should happen for the organization if they change their behavior? What consequences will they face personally if they resist?

Resource

One of the most common sources of failure in change efforts is a failure to support change efforts appropriately. Such efforts fail to appreciate the commonplace expression that "you have to invest money to make money." Successful change efforts tend to program explicit resources to support the change. Individuals changing their behavior are not expected to use their own funds to support organizational change unless they can expect to get offsetting funds if it succeeds. They are not expected to take on the change initiative without reducing other workloads unless they can anticipate future rewards from success that make it worthwhile to invest their own time in its success. In particular, they get relief from their normal workload to take time in training programs.

Train

Three kinds of training are important. One explains why the change is good for their organization and, by implication, for them. The second explains the concrete details of the change and their role in it and conveys any new substantive skills they will need to execute their new role successfully. The third provides broader training on how to succeed in an environment where continuous improvement is the norm. This third form of training helps employees understand how individual incremental changes fit together and build on one another. It trains them to participate more effectively in the ongoing change process by observing the effects of change in their own localities within the organization and sharing this information to support ongoing efforts to build future increments of change. Training typically focuses within an organization, but organizations that perceive themselves as active links in integrated supply chains may also train individuals in the organizations they deliver outputs to and in the organizations they receive input from.

Measure

Successful change identifies metrics and tracks performance against these metrics. Appreciating that what gets measured gets done, successful change managers seek metrics that align the decisions and behavior of individuals in the organization to its higher-level goals. They also choose metrics that can inform them about how well the organization as a whole is performing, even if it is impossible to link such metrics to individuals in the organization. The campaign plan for a change sets targets for these goals on a schedule and tracks the performance of the change effort itself against such targets. Feedback from such tracking can inform the executives in the coalition overseeing the change. If the metrics are not helpful to them, they can ask for changes in metrics or dates at which they expect target goals to be achieved.

Motivate

Effective motivation includes two steps. The first simply gives individuals room to change their behavior to promote organizational change. If the organizational change calls for them to use risk-assessment information to sample, for example, it includes provisions that make sure that such risk-assessment information is available. If it asks them to interview airmen to assess the quality of discipline at a wing, it gives them tools they can use to structure those interviews and the responses they get. Some writers refer to this first step as *empowerment* of those who must change their behavior.

Second, within the effective freedom they have to act, every organization has its own approach to motivating its personnel to promote its goals. Some use moral suasion, professional standards, and peer pressure. Aspects of transformational leadership fall into this set of tactics as well. Others adapt a style more consistent with transactional leadership, using tangible rewards tied to measured performance, which can range from direct cash payments to larger operational budgets to promises of promotion or better assignments to opportunities for career-enhancing training. The Air Force uses variations on most of these other than options that directly involve cash payments and budgets. The Air Force could align the success or failure of any incremental change to the constellation of instruments that it already uses to motivate its personnel to promote its other goals. Successful change leads to good outcomes for those held accountable; failure to change leads to bad outcomes.[6]

Build on Incremental Successes

Significant change takes time. Large organizations are typically more impatient than the formal hurdle rates they are supposed to use to make decisions would indicate. As a practical matter, in private industry, leaders are often judged against quarterly goals or other relatively short-term metrics. In the Air Force, leaders turn over often, making it difficult for them to take credit for anything that takes more than a year or so to achieve. Bite-size increments can scale a large change down into pieces small enough so that information about progress can be generated every few months. Successful change efforts track their performance at short intervals and trumpet news of success to those they rely on for continuing leadership attention and resource support. Such efforts start with modest resources on relatively easy tasks and, as a change team builds experience, confidence, and success, it takes on increasingly challenging tasks made possible by increasingly large commitments of resources.

---

[6] The quality management movement tends to favor rewarding teams over rewarding individuals. But this presumes that work is organized to be executed by teams, sometimes without formal leaders, and that the teams have formal authority to act without the concurrence of the organizations that its members come from. This makes it feasible to hold the members of teams accountable for their measured mutual performance. That is not how authority is assigned or how performance measurement affects performance assessment in the Air Force today.

Communicate

What was important during preparation for change is even more important during execution, because the resources—and so the stakes—are higher. And many more and more diverse players are involved. Tasks are becoming more and more challenging. The need for active coordination is growing. Rich communication, top to bottom, bottom to top, and side to side, in many media, is critical to effective coordination. Without effective communication, as change happens, people lose a common understanding of how the Air Force works.

Consolidate Successes

As sets of increments meet enough success for the organization to rely on them, it can dissolve legacy processes that are no longer needed. When legacy processes are gone, change is essentially complete. Formal change managers have differing views on how to manage this transition from the old to the new. Some advocate maintaining old and new side by side (so-called scaffolding) until the organization is absolutely sure the new systems will work. They promote this even if it creates duplication, confusion about which is the official system at any point in time, or ambiguity about the depth of the leadership's commitment to real change. Others see the rapid dissolution of legacy processes as a way to heighten individuals' sense of being on a burning platform or to convey the leadership's unbridled commitment to the future.

Element 3: Sustain

Two kinds of sustainment activities are important. The first concerns the institutionalization of the changes completed in any set of execution activities like those described above. Institutionalization changes the status of the changes completed from innovation activities to routine activities, that is, a new status quo. Below, we call the activities that result from such changes "now routine products of a formal change process." This is likely to involve transferring responsibility for these completed changes from the coalition of executives and managers coordinating the changes to the standard MAJCOM command or functional channels that coordinate most activities in the Air Force. The second kind of sustainment activity keeps in mind that, even following such a transfer, new increments of change are likely to continue, building on those that have been completed. The successful design of the new increments is likely to depend on continuing information flow between them and the changes that came before.

Resource

Resourcing now occurs through standard channels and must compete for support without as much special leadership focus. This can place significant demands on organizations responsible for managing the now routine products of a formal change process. These activities continue to generate information that flows into ongoing change management efforts. These activities may retain personnel with expertise in successful change who could be called upon as subject-matter experts or trainers of the personnel still active in ongoing change efforts. Ongoing change-management efforts are likely to have better access to inputs from now routine products of the change process if the ongoing efforts program for their participation. Alternatively, the Air Force can encourage these now routine activities to make their information and personnel available to ongoing change efforts.

Measure

Organizational changes that result from quality management change efforts often build in a different kind of performance measurement than that used elsewhere in the Air Force. The measurement emphasizes ongoing benchmarking against analogous activities elsewhere and emphasis on continuous improvement rather than simple achievement of long-standing Air Force standards. Information from now routine activities will be more valuable to ongoing change efforts the more it flows from this new approach to measurement. Such changes in measurement culture can also potentially seed broader application of this approach across the Air Force.

Motivate

Now routine products of the change process operate within the standard systems the Air Force uses to motivate performance. If the Air Force wants these products of change to continue to inform ongoing change efforts, it must ensure that it gives personnel in these activities space to participate in ongoing change activities and rewards them appropriately (in the context of their current primary duties) for effective participation.

Adjust

Effective continuous monitoring of any activity is likely to reveal ongoing opportunities for improvement. The desirability of organizational change need not end simply because the Air Force now uses its standard oversight mechanisms to align the activity to the Air Force as a whole. As new opportunities for improvement come to light, issues discussed above concerning planning and executing change return. If these changes are large enough, it may be worthwhile to pursue them under the umbrella of the formal change-management coalition described above. The effort to manage additional change should be scaled to the size of the additional change considered.

Communicate

As long as change continues, effective communication can keep all the players up to date on the current status of the now routine product of formal change. In general, the quality change approach encourages greater communication across an organization than the Air Force as a whole does today. Preserving the tradition of such communication in now routine products of formal change management could help seed this approach across the Air Force.

Consolidate and Anchor

Over time, new participants in an activity that has used a new approach successfully for a long time come to the activity with no knowledge of what the activity looked like before the formal change-management exercise. The new approach is fully internalized and taken for granted. People forget that significant change ever happened. As some have said about the gradual absorption of quality management methods into day-to-day management activities in successful organizations, people forget that the language they speak every day is called "prose." As noted repeatedly above, that does not mean that change ends in the activity. It means only that when personnel in the activity today happen to stumble across the language spoken in the activity before the change-management exercise, it sounds archaic and nonsensical to them. Change is truly complete.

Summary

Formal change-management methods have been developed, applied, and refined in the United States over the last three decades. A consensus has emerged, in the academic and technical business literatures that study change management in the private and public sectors, on the factors that have helped ensure the successful implementation of significant changes in large, complex organizations like the U.S. Air Force.

In general, it is helpful to conceive of a change as having three elements devoted to planning, execution, and sustainment. Planning sets the stage for a change by gathering senior leadership support for it, assembling a coalition of all parties with a stake in the change, articulating what the change is and how it helps the organization, and developing initial evidence that the change will have the effects intended. Execution transforms the concept behind the change into concrete actions that roll the change out through the organization, monitoring its effects as evidence about its performance accumulates, and adjusting the change to reflect experience to date. Sustainment moves the change from special oversight to ensure its execution to full integration with the standing policies, practices, and processes of the organization. Large changes typically occur in increments that allow the leadership to monitor performance in a timely way and sustain support for ongoing change. As a result, planning, execution, and sustainment activities often occur simultaneously, side by side, as each incremental change works through its full, life-cycle action plan.

Throughout any change effort, a number of factors can affect the likelihood of success. Training ensures that the individuals who must change their own behavior for organizational change to succeed know how to fulfill their roles in the change. The organization seeks to motivate individual change by giving individuals enough freedom—that is, enabling them—to change their behaviors as required to allow organizational change. It then uses all the instruments that the organization normally uses to motivate its personnel to reward individual behaviors that support organizational change and sanction those that do not. Testing and measurement monitor the status of the change as it proceeds. When such monitoring detects weaknesses or failures, the organization reacts quickly to mitigate the problems involved or to terminate a change that is not working as planned. When such monitoring confirms success, the organization cites this information as evidence throughout the change to sustain the support of the senior leadership and the coalition of interested parties. Appropriate resource planning ensures that the organization makes resources available to support training, motivation, testing, and adjustment throughout the change effort.

Appendix F. U.S. Navy Technology Transition Agreement Template

The following template[1] provides an outline of a typical Navy Technology Transition Agreement (TTA). We recommend AFSPC and SMC implement a similar product to codify rapid prototyping activities and the planned disposition of rapid prototypes.

Prototype Transition Agreement
Between
(Customer for the Prototype)
(Science & Technology [S&T] Manager/Prototype Developer)
and
(Gaining SMC Program Manager)

1. Introduction

1.1. **Purpose/Scope.** Provide a brief statement. [Example: The *Program Manager* and *S&T Organization* mutually agree to enter into this Technology Transition Agreement (TTA) for the purpose of defining technology deliverables from the *appropriate name* technologies development program, to *appropriate name* program. This TTA defines the functional responsibilities and support relationships between the parties signing this agreement. It ensures a clear understanding of the responsibilities of all parties to ensure a successful transition of technology from *S&T organization* to *the program of record name*.]

1.2. **Summary.** Provide a brief overview of two to three paragraphs' length summarizing what this project will provide to the program of record, an explanation of the current situation (usually expressed as a problem or shortcoming), and what funding (list funding amount) will be used in bringing a corrective solution to fruition. Include the causes or reasons for the current problem/predicament. Describe the impact of the problem in terms of reduced operational effectiveness (e.g., warfighting capability or mission accomplishment) or efficiency (e.g., total ownership cost). Justify the reason(s) for a technology insertion outside the normal Program Objective Memorandum (POM) cycle. Describe the seminal transition event and when it will happen. Also provide the number of units (or some other quantitative metric) that will be procured with the new technology transitioned. Indicate the consequence or alternative action to be taken if the transition funding is not implemented (what is your Plan B to get the solution transitioned).

[1] U.S. Navy, Office of Naval Research, "Elements of a Technology Transition Agreement (TTA)," undated.

2. Basic Transition Agreement.

   2.1. **Technology Name.** Name or names of the technology to be transitioned (T2BT). Description of the technology to be delivered. Include all aliases, prior names, or other identifying numbers or acronyms. Indicate if the T2BT is only a subset of the underlying S&T program.

   2.2. **Description of Technology or Capability to Be Delivered.** Specific, technical description of what the S&T program or source-program manager intends to develop for transition to the primary acquisition program, to include numbers of prototypes or test items. The description should include delivery dates, delivery mechanism (purchase, loan, given to program, etc.) and specific exit criteria concerning the capability to be available at each delivery date. Identify the project objectives associated with each fiscal year's funding (corresponding to the funding in Tables F.1 and F.2 in the Annex) as well as the outcomes to be achieved with that funding.

   2.3. **Target Acquisition Program.** Brief description of the acquisition program intended to receive the T2BT. Include major program objectives, acquisition category level, current phase of acquisition life cycle, next milestone decision review (and anticipated date), and projected initial operational capability date.

   2.4. **Acquisition Program Technology Need.** Brief description of the benefit that this T2BT will bring to the acquisition program, or need satisfied. Identify the technology needs of the acquisition program that S&T is expected to provide.

      2.4.1. Relate the benefit to the Initial Capabilities Document (ICD), CDD, Key Performance Parameters, etc.

      2.4.2. Include need dates for specific capabilities.

      2.4.3. Provide an estimate of the Technology Readiness Level (TRL) for each technology/product need identified utilizing a systems approach for hardware and software as the measure of technical maturity and indication of transition readiness. Coordinate the TRL with the S&T activity.

   2.5. **Integration Strategy.** Describe the process for integrating the T2BT into the acquisition program. Include the following elements of the acquisition strategy:

      2.5.1. Evolutionary acquisition, block upgrade, etc.

      2.5.2. Required contractor-to-contractor agreements

      2.5.3. Acquisition appropriation and Program Element (PE) numbers funding the transition

      2.5.4. Annual PE funding levels committed to the transition program (see Table F.1 in the Annex)

2.5.5. Transition FY

2.5.6. Statement conveying the level of commitment. For example:

2.5.6.1. Commitment: [Example: Upon successful demonstration of key performance requirements (exit criteria), *appropriate name acquisition program office* will integrate *XXX* (product S&T organization will deliver) into *appropriate name* (acquisition program that will integrate the deliverable) commencing in FYXX (transition year). This integration effort will be funded under PE XXXXXXX, Project XXXX (Future Years Defense Program budget profile for this acquisition line should be included).]

2.5.6.2. Intent: [Example: Upon successful demonstration of key performance requirements (exit criteria), *appropriate name* (acquisition program office) intends to integrate *XXX* (product S&T organization is delivering) into *appropriate name* (acquisition program that will integrate the deliverable) commencing in FYXX (transition year) under PE XXXXXXX Project XXXX (Future Years Defense Program budget profile).]

2.6. **Program Manager/Project Officer.** Identify the program manager and the individual in the program office responsible for day-to-day management, with contact information, concerning the T2BT.

2.7. **Technology Manager.** Identify the individual designated by either the S&T activity or the source technology program office, Program Manager (PM) or Program Executive Officer to be the coordinator and day-to-day manager of the development of the T2BT.

2.8. **Capability Requirement Basis.** Identify the governing source of the capability requirement: the ICD, CDD, or other official reference documenting the capability need.

2.9. **Resource Sponsor/Requirements Officer.** Identify the resource sponsor and requirements officer responsible for resourcing and establishing requirements for the capability. Include contact information.

3. Technical Details and Programmatics.

3.1. **Current Status of Technology.**

3.1.1. **Status Summary.** Summarize the current state of the development of the T2BT. Identify primary areas where additional development is required. Provide estimate of current TRL ratings. Indicate if an independent analysis of the TRL has been performed, and, if so, provide a summary of that analysis.

3.1.2. **Risk Analysis.** Major areas of risk, prioritized, with planned mitigation activities. Include technical, producibility, affordability, sustainability, cost, and schedule risks.

3.2. **Technology Development Strategy.** Outline planned approach. Describe current efforts and efforts required beyond those currently underway. Detail integration plans if multiple projects are planned. Include planned advanced technology demonstration or advanced concept technology demonstration, if applicable.

3.3. **Key Measures of Transition Readiness.** Identify the key parameters or attributes that will be used as exit criteria to measure whether or not the T2BT effort is proceeding as scheduled. Include parameters to be tracked, current state, interim progress estimates, and final objective. TRLs are a measure of technical maturity and can be used to assess readiness to transition. Provide dates when each higher TRL rating is expected to be achieved.

3.4. **Program Plan.** Show major activities/efforts of the T2BT technology development activity, with milestones. Provide a schedule with all pertinent information. Use Table F.2 in the Annex as a sample to relate the planned tasks with estimated funds required for each.

3.5. **Funding Adequacy.** State and agree that the combined sources of all funding are adequate to achieve the maturity and quantity of the T2BT required by the receiving PM in the time frame(s) required by the PM and as specified in this document.

4. Reporting Requirements.

4.1. The program manager will provide a semiannual technical status no later than June 30 and December 31, a transition report within 60 days of the transition event, and a final letter report within 30 days of fielding the technology.

4.2. The technology manager will provide monthly funds execution status reports no later than the 28th day of each month.

4.3. Address applicable NDAA and Appropriations Act language and reporting requirements as required.

5. Signatures.

Annex

Table F.1. Funding Sources (in millions of dollars)

	FY16	FY17	FY18	FY19	FY20	FY21	FY22	FY23	FY24	FY25	Total
Source	**Transition Funding ($M)**										
(RDT&E) PE: 06XXXXXXX	$0.000	$0.000	$0.000	$0.000	$0.000	$0.000	$0.000	$0.000	$0.000	$0.000	$0.000
Source 2 and PE #	$0.000	$0.000	$0.000	$0.000	$0.000	$0.000	$0.000	$0.000	$0.000	$0.000	$0.000
Sub-total	$0.000	$0.000	$0.000	$0.000	$0.000	$0.000	$0.000	$0.000	$0.000	$0.000	$0.000
Source	**Integration Funding ($M)**										
POR name and PE #	$0.000	$0.000	$0.000	$0.000	$0.000	$0.000	$0.000	$0.000	$0.000	$0.000	$0.000
Source 2 and PE #	$0.000	$0.000	$0.000	$0.000	$0.000	$0.000	$0.000	$0.000	$0.000	$0.000	$0.000
Sub-total	$0.000	$0.000	$0.000	$0.000	$0.000	$0.000	$0.000	$0.000	$0.000	$0.000	$0.000
Source	**Procurement Funding ($M) and Quantity to be Procured**										
POR name and PE #	$0.000	$0.000	$0.000	$0.000	$0.000	$0.000	$0.000	$0.000	$0.000	$0.000	$0.000
Quantity											
Sub-total	$0.000	$0.000	$0.000	$0.000	$0.000	$0.000	$0.000	$0.000	$0.000	$0.000	$0.000
TOTAL	$0.000	$0.000	$0.000	$0.000	$0.000	$0.000	$0.000	$0.000	$0.000	$0.000	$0.000

Table F.2. Major Task Schedule and Funding (in millions of dollars)

		Required Funding ($M)							
Organization	**Milestone Task**	**FY16**	**FY17**	**FY18**	**FY19**	**FY20**	**FY21**	**FY22**	**Total**
SMC/XXX	Contract award	$X.000							$X.000
S&T agency	Development	$X.000							$X.000
Test agency	Test and evaluation		$Y.000						$Y.000
Support agency	Support task		$Y.000						$Y.000
SMC/XXX (PMO)	PMO costs		$Y.000						$Y.000
AFPEO/SP	Procurement			$Z.000	$Z.000	$Z.000	$Z.000	$Z.000	$Z.000
AFSPC unit	Deployment				$Z.000				$X.000
Total		$0.000	$0.000	$0.000	$0.000	$0.000	$0.000	$0.000	$0.000

NOTE: AFPEO/SP = Air Force Program Executive Office for Space. PMO = program management office. POR = program of record. RDT&E = research, development, test, and evaluation.

Appendix G. FY 2016–2017 National Defense Authorization Act Provisions Related to Prototyping

This appendix provides a more detailed summary of prototyping-related provisions enacted in the FY 2016 and FY 2017 NDAAs.

Other Transaction Authority

Codified in Title 10, United States Code, Section 2371 (10 USC § 2371), Other Transaction Authority (OTA) offers a streamlined pathway to execute rapid prototyping projects and transition them into follow-on production and operational service. OTA is a collection of legal authorities that allow the DoD to conduct research-and-development activities largely outside the boundaries of traditional Federal Acquisition Regulation (FAR)–based competitive procurement, contracting, reporting, and oversight requirements. OTA was first enacted in 1989 as a tool for the Defense Advanced Research Projects Agency to conduct basic, applied, and advanced research projects. In 1994, Congress extended OTA provisions for conducting prototyping projects. In the FY 2016 NDAA, Congress substantially expanded the scope and applicability of OTA for prototyping purposes.[1]

10 USC § 2371b, *Authority of the Department of Defense to carry out certain prototype projects*, authorizes the secretary of a military department or any other official designated by the Secretary of Defense "to carry out prototype projects that are directly relevant to enhancing the mission effectiveness of military personnel and the supporting platforms, systems, components, or materials proposed to be acquired or developed by the Department of Defense, or to improvement of platforms, systems, components, or materials in use by the armed forces."[2] In general terms, this authority allows DoD to enter into legally binding arrangements with industry partners, small businesses, and nontraditional defense contractors without bearing the burdens normally associated with traditional FAR-based contracting rules. Though this statute includes several conditions and restrictions regarding when and how OTA may be used, it potentially opens a door for AFSPC and SMC to access innovative space technologies and to develop,

[1] U.S. Air Force Office of Transformational Innovation, "Everything You Always Wanted to Know About Other Transaction Authority, but Were Afraid to Ask," February 2017.

[2] 10 USC § 2371b.

evaluate, and transition prototype systems or capabilities into its evolving space architectures.[3] OTA offers several potential benefits in this regard. It can[4]

- lower barriers for small and nontraditional defense businesses to enter the space S&T arena by reducing the red tape associated with traditional DoD contracting
- allow AFSPC/SMC to tailor business models and OTA arrangements to specific operational circumstances and technology needs
- reduce the solicitation-to-award time line compared to traditional FAR-based contracting
- provide an opportunity for prototype development cost sharing between the government and industry, effectively reducing AFSPC RDT&E investment requirements
- facilitate the transition of prototype systems/capabilities from the development process to initial production and fielding if "competitive procedures" were used to award the prototyping arrangement and the industry participant(s) "successfully completes" the prototype.

FY 2016 NDAA Section 804

In the FY 2016 NDAA (Section 804), Congress directed new guidance for an acquisition process lying between the traditional DoD acquisition model, which typically involves many years from program inception to initial operational capability, and the accelerated acquisition model, which is designed to field a solution for an urgent operational need in two years or less. In doing so, Congress directed the department to establish guidance for "middle tier" acquisition programs that would be completed between two and five years from inception. The act specifically requires the department to establish guidance for two acquisition pathways: rapid prototyping and rapid fielding.[5]

Regarding rapid prototyping, Congress's intent is to provide a rapid means to develop fieldable prototypes that demonstrate innovative technologies and new capabilities in response to emerging military needs. The objective of this process would be to demonstrate such prototypes in operational environments and lay a foundation to field an operational capability within five years from the time an operational requirement is approved. The NDAA (as amended) specifies several aspects the guidance should entail:[6]

> (A) *a merit-based process for the consideration of innovative technologies and new capabilities to meet needs communicated by the Joint Chiefs of Staff and the combatant commanders;*

---

[3] AFSPC is, in fact, currently using OTA to develop its Space Enterprise Consortium construct, where companies that take part in the consortium will be eligible to compete for rapid prototyping projects and, if successful, insert them into PORs.

[4] To understand more about OTA, see U.S. Air Force Office of Transformational Innovation, 2017.

[5] Public Law 114-92, Section 804, November 25, 2015.

[6] See 10 USC 2302 note for the latest version, which only includes four of the five original elements.

(B) a process for developing and implementing acquisition and funding strategies for the program;
(C) a process for demonstrating and evaluating the performance of fieldable prototypes developed pursuant to the program in an operational environment;
(D) a process for transitioning successful prototypes to new or existing acquisition programs for production and fielding under the rapid fielding pathway or the traditional acquisition system.

Regarding rapid fielding, Congress's intent is to provide a quick way to produce and field new or upgraded systems with minimal development requirements. The limits on this process are to begin production within six months and complete the fielding process within five years from the time a requirement for such systems is approved. The law includes several specific provisions this process should entail, including a process for considering lifecycle costs and addressing issues of logistics support and system interoperability for transitioning systems/products.[7]

Recognizing that rapid prototyping also requires flexible funding to initiate and conduct such activities, Congress included a provision in Section 804 that directed the Secretary of Defense to establish a "Department of Defense Rapid Prototyping Fund" to facilitate the new rapid prototyping authorities established in this section of the law. This fund is managed by a senior official of DoD, and the statute authorizes the Office of the Secretary of Defense (OSD) to transfer funding from this account to the military departments to conduct rapid prototyping activities. Other funds could be established by the secretary of each military department.

A potentially beneficial aspect of this legislation is the relief the secretary's guidance may provide from the bureaucratic hurdles often associated with the JCIDS and regular acquisition processes in Joint Staff and OSD guidance (Chairman of the Joint Chiefs of Staff Instruction 3170.01, DoDD 5000.01). In Section 804, Congress provided for streamlining the requirements generation, budget preparation, and acquisition processes to support rapid prototyping and rapid fielding. The legislation specifically exempts rapid prototyping and fielding programs from having to comply with JCIDS Manual and DoDD 5000.01 requirements (except to the extent specifically provided in the Secretary of Defense's guidance) and instead calls for new processes to be put in place that enable a rapid prototyping or fielding requirement to be approved within six months from the time such processes are initiated. Interim guidance was provided by the Under Secretary of Defense for Acquisition and Sustainment on April 16, 2018.[8]

---

[7] Public Law 114-92, November 25, 2015.

[8] Under Secretary of Defense for Acquisition and Sustainment, "Middle Tier of Acquisition (Rapid Prototyping/ Rapid Fielding) Interim Authority and Guidance," Memo, Washington, D.C.: Department of Defense, April 16, 2018.

FY 2017 NDAA Section 806

In the FY 2017 NDAA, Congress further strengthened the rapid prototyping and rapid fielding provisions enacted the previous year by adding new sections 2447a–e to 10 USC. In Section 806 of the act, Congress comprehensively restructured numerous legislative requirements related to weapon system component or technology prototype development and deployment. Among the many provisions contained in this section of the law, Congress[9]

- directs the secretary of each military department to establish an oversight body of senior advisors to manage prototyping projects for weapon system components and related technologies
- requires this oversight body "to issue a strategic plan every three years that prioritizes the capability and weapon system component portfolio areas for conducting prototype projects, based on assessments of high-priority warfighter needs; capability gaps or readiness issues with major weapon systems; opportunities to incrementally integrate new components into major weapon systems based on commercial technologies or S&T efforts that are expected to be sufficiently mature to prototype within three years; and prioritizes the capability and weapon system component portfolio areas for conducting prototype projects, based on assessments of major weapon systems"
- requires this oversight body to annually recommend appropriate funding levels for these prototype projects and to ensure such projects allow for appropriate experimentation and technology risk
- requires this oversight body to ensure prototype projects have a plan to transition the prototype into a fielded system, POR, or operational use upon successful achievement of project goals
- directs prototype projects be completed within two years of initiation
- authorizes prototype projects to be funded via traditional FAR-based contracts, cooperative agreements, or other transactions (10 USC § 2371)
- authorizes the service acquisition executive of each military department to award a follow-on production contract for prototype technologies without the use of competitive procedures under certain conditions
- authorizes the secretary of each military department to transfer unobligated procurement funding as needed to fund low-rate initial production of systems or components for rapid fielding projects (within certain limitations).

[9] See 10 USC, Sec. 2447a-e. The original statutes came from Public Law 114-328, Section 806, December 23, 2016.

References

AFSPC—*See* Air Force Space Command.

AFSPC Public Affairs, "AFSPC Commander Announces Space Enterprise Vision," April 11, 2016. As of May 17, 2018: http://www.afspc.af.mil/News/Article-Display/Article/730817/afspc-commander-announces-space-enterprise-vision/

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, and Dave Thomas, "Manifesto for Agile Software Development," 2001. As of January 12, 2017: http://agilemanifesto.org/

Boehm, Barry, and Richard Turner, *Balancing Agility and Discipline: A Guide for the Perplexed,* Boston: Addison-Wesley, 2004.

Boeing staff, discussion on MOSA, agile, and rapid prototyping, El Segundo, Calif., June 29, 2017.

Boudreau, Michael, "Acoustic Rapid COTS Insertion: A Case Study in Spiral Development," NPS-PM-06- 041, October 30, 2006.

Bradbury, Joel, "Lean: Muda, Mura, Muri," *Quality Digest*, March 8, 2017. As of March 8, 2017: http://www.qualitydigest.com/inside/lean-article/muda-mura-muri-021317.html?utm_source=MadMimi&utm_medium=email&utm_content=A+short+history+of+lean+++Muda%2C+mura%2C+muri+++Auditing%E2%80%99s+most+important+question&utm_campaign=20170307_m138003778_QD+3-08-17+Sparta&utm_term=full+story

Camm, Frank, "Adapting Best Commercial Practices to Defense," in Stuart E. Johnson, Martin C. Libicki, and Gregory F. Treverton, eds., *New Challenges, New Tools for Defense Decisionmaking*, Santa Monica, Calif.: RAND Corporation, MR-1576-RC, 2003, pp. 211–246. As of August 23, 2018: https://www.rand.org/pubs/monograph_reports/MR1576.html

Camm, Frank, Jeffrey A. Drezner, Beth E. Lachman, and Susan A. Resetar, *Implementing Proactive Environmental Management: Lessons Learned from Best Commercial Practice*, Santa Monica, Calif.: RAND Corporation, MR-1371-OSD, 2001. As of August 23, 2018: https://www.rand.org/pubs/monograph_reports/MR1371.html

Camm, Frank, Laura Werber, Julie Kim, Elizabeth Wilke, and Rena Rudavsky, *Charting the Course for a New Air Force Inspection System*, Santa Monica, Calif.: RAND Corporation, TR-1291-AF, 2013. As of August 23, 2018: https://www.rand.org/pubs/technical_reports/TR1291.html

Cook, Cynthia R., Laura Werber Castaneda, and Abigail Haddad, "Implementation," in Bernard D. Rostker et al., *Sexual Orientation and U.S. Military Personnel Policy: An Update of RAND's 1993 Study*, Santa Monica, Calif.: RAND Corporation, MG-1056, 2010, pp. 371–388. As of August 23, 2018: https://www.rand.org/pubs/monographs/MG1056.html

Council of Defense and Space Industry Associations (CODSIA), Response to DFARS Case 2012-D022, September 30, 2016.

Defense Federal Acquisition Regulation Supplement (DFARS) §§ 252.227-7013, 252.227-7014.

DeLuca, Paul, Joel B. Predd, Michael Nixon, Irv Blickstein, Robert W. Button, James G. Kallimani, and Shane Tierney, *Assessing Aegis Program Transition to an Open-Architecture Model*, Santa Monica, Calif.: RAND Corporation, RR-161, 2013. As of August 23, 2018: https://www.rand.org/pubs/research_reports/RR161.html

DFARS—*See* Defense Federal Acquisition Regulation Supplement.

Fernandez, Sergio, and Hal G. Rainey, "Managing Successful Organizational Change in the Public Sector," *Public Administration Review*, Vol. 66, No. 2, 2006, pp. 168–176.

Firesmith, Donald, *Open System Architectures: When and Where to Be Closed*, SEI Insights Blog, Carnegie Mellon University, Software Engineering Institute, Pittsburgh, Pa., October 2015. As of September 7, 2017: https://insights.sei.cmu.edu/sei_blog/2015/10/open-system-architecture-when-and-where-to-be-closed.html

Galpin, Timothy J., *The Human Side of Change: A Practical Guide to Organization Redesign*, 1st ed., San Francisco: Jossey-Bass Publishers, 1996.

Intel Corporation, *Two Decades of "Plug and Play": How USB Became the Most Successful Interface in the History of Computing*, 2014. As of September 7, 2017 https://www.intel.com/content/dam/www/public/us/en/documents/articles/usb-two-decades-of-plug-and-play-article.pdf

International Standards Organization, *How We Develop Standards*. As of September 7, 2017: https://www.iso.org/developing-standards.html

International Telecommunications Union, *Definition of "Open Standards,"* November 11, 2015. As of September 7, 2017: http://www.itu.int/en/ITU-T/ipr/Pages/open.aspx

Johnson, William, "The A-RCI Process—Leadership and Management Principles," *Naval Engineers Journal*, Vol. 116, No. 4, 2004, pp. 99–105.

Judson, A. S., *Changing Behavior in Organizations: Minimizing Resistance to Change*, Cambridge, Mass.: Basil Blackwell, 1991.

Kotter, John P., *Leading Change*, Boston: Harvard Business School Press, 1996.

Lapham, Mary Ann, Suzanne Garcia-Miller, Lorraine Adams, Nanette Brown, Bart Hackemack, Charles (Bud) Hammons, Linda Levine, and Alfred Schenker, *Agile Methods: Selected DoD Management and Acquisition Concerns*, Technical Note CMU/SEI-2011-TN-002, Carnegie Mellon University, Software Engineering Institute, Pittsburgh, Pa., October 2011.

Limer, Eric, *Here's the Chip Apple Is Using to Stop you from Buying Cheap Cables*, Gizmodo, September 24, 2012. As of September 7, 2017: https://gizmodo.com/5945889/some-third-party-adapters-might-not-work-with-your-new-iphone

Meyer, Bertrand, *Agile! The Good, the Hype, and the Ugly*, Zürich, Switzerland: Springer, 2014.

Modigliani, Peter, and Su Chang, *Defense Agile Acquisition Guide: Tailoring DoD IT Acquisition Program Structures and Processes to Rapidly Deliver Capabilities*, MITRE Corporation, McLean, Va., March 2014.

Moore, Nancy Y., Laura H. Baldwin, Frank Camm, and Cynthia R. Cook, *Implementing Best Purchasing and Supply Management Practices: Lessons from Innovative Commercial Firms*, Santa Monica, Calif: RAND Corporation, DB-334-AF, 2002. As of August 23, 2018: https://www.rand.org/pubs/documented_briefings/DB334.html

Open Group, *Technical Standard for Future Airborne Capability Environment*, Edition 2.1, May 2014. As of September 7, 2017: http://www.opengroup.org/face/tech-standard-2.1

Public Law 114-328, Section 806, December 23, 2016. As of August 17, 2017: https://www.congress.gov/congressional-report/114th-congress/house-report/840/1

Public Law 114-92, Section 804, November 25, 2015. As of August 17, 2017: https://www.congress.gov/bill/114th-congress/senate-bill/1356/text/pl?overview=closed

Riposo, Jessie, John Gordon IV, Robert Murphy, Bradley Wilson, and Isaac R. Porche III, *CANES Contracting Strategies for Full Deployment*, Santa Monica, Calif: RAND Corporation, TR-993, 2013. As of October 3, 2018: https://www.rand.org/pubs/technical_reports/TR993.html

Sather, Vale, Michelle Dobard-Anderson, and David Naiditch, "JMS Inc. 2 Agile Fit Check," Version 1.2, SMC/EN, Los Angeles Air Force Base, Los Angeles, Calif., November 16, 2015.

Silverthorne, Valerie, "Change the Culture to Fix the Tragedy That DevOps Can Be," TechTarget.com, 2017. As of May 24 2017: http://searchsoftwarequality.techtarget.com/feature/Change-the-culture-to-fix-the-tragedy-that-DevOps-at-scale-can-be

SMC—*See* Space and Missile Systems Center.

Space and Missile Systems Center, *Acquiring and Enforcing the Government's Rights in Technical Data and Computer Software Under Department of Defense Contracts: A Practical Handbook for Acquisition Professionals*, 3rd ed., Office of the Staff Judge Advocate, Space and Missile Systems Center, January 2011.

———, *Acquiring and Enforcing the Government's Rights in Technical Data and Computer Software Under Department of Defense Contracts: A Practical Handbook for Acquisition Professionals*, 7th ed., Office of the Staff Judge Advocate, Space and Missile Systems Center, August 2015.

Under Secretary of Defense for Acquisition and Sustainment, "Middle Tier of Acquisition (Rapid Prototyping/Rapid Fielding) Interim Authority and Guidance," Memo, Washington, D.C.: Department of Defense, April 16, 2018. As of March 9, 2018: http://acqnotes.com/wp-content/uploads/2018/05/OSD-Middle-Tier-of-Acquisition-Interim-Authority-and-Guidance-16-Apr-2018..pdf

Universal Serial Bus Implementer's Forum, *About USB Implementers Forum, Inc*. As of September 7, 2017: http://www.usb.org/about

———, *Universal Serial Bus 3.1 Specification*. As of September 7, 2017: http://www.usb.org/developers/docs/

U.S. Air Force Office of Transformational Innovation, "Everything You Always Wanted to Know About Other Transaction Authority, but Were Afraid to Ask," February 2017. As of August 26, 2018: http://www.transform.af.mil/Projects/Other-Transaction-Authority/

U.S. Army, *Army Data & Data Rights (D&DR) Guide, A Reference for Planning and Performing Data Acquisition and Data Management Activities Throughout the DoD Life Cycle*, 1st ed., U.S. Army Product Data and Engineering Working Group, August 2015.

U.S. Code, 10 USC § 2371b. As of August 17, 2017: https://www.gpo.gov/fdsys/pkg/USCODE-2015-title10/html/USCODE-2015-title10-subtitleA-partIV-chap139-sec2371b.htm

U.S. Department of Defense, *MIL-STD-704F, Aircraft Electric Power Characteristics*, Revision F, March 2004a. As of October 4, 2018: https://prod.nais.nasa.gov/eps/eps_data/137899-SOL-001-015.pdf

———, *Program Manager's Guide to Open Systems*, Version 2.0, September 2004b. As of October 4, 2018: http://www.acqnotes.com/Attachments/Program%20Managers%20Guide%20to%20Open%20Systems,%20Sept%202004.pdf

———, *Directive 5000.01: The Defense Acquisition System*. November 20, 2007. As of October 4, 2018: https://www.dau.mil/guidebooks/Shared%20Documents%20HTML/DoDD%205000.01.aspx

———, *MIL-STD-31000A*, *Department of Defense Standard Practice: Technical Data Packages*, Revision A, February 2013a. As of October 4, 2018: http://quicksearch.dla.mil/Transient/D31F7FB903F847A6857D0FAC0B34A375.pdf

———, *Open Systems Architecture Contract Guidebook for Program Managers*, Version 1.1, June 2013b. As of October 4, 2018: http://www.acqnotes.com/Attachments/Open%20System%20Architecture%20(OSA)%20Contract%20Guidebook%20for%20Program%20Managers%20June%2013.pdf

———, *Intellectual Property Strategy*, Department of Defense Open Systems Architecture Data Rights Team, August 2014. As of October 4, 2018: https://www.dau.mil/cop/mosa/DAU%20Sponsored%20Documents/IP%20Strategy%20Brochure_Final%202-10-15.pdf

———, "AN/BQQ-10 Acoustic Rapid Commercial Off-the-Shelf Insertion (A-RCI) Sonar," *Director, Operational Test and Evaluation FY2016 Annual Report,* December 2016. As of September 7, 2017: http://www.dote.osd.mil/pub/reports/FY2016/pdf/navy/2016arci.pdf

———, *Director, Operational Test and Evaluation Test and Evaluation Master Plan Guidebook*, Version 3.1, January 19, 2017. As of October 4, 2018: http://www.dote.osd.mil/docs/TempGuide3/TEMP_Guidebook_3.1.pdf

U.S. Department of Defense, Defense Standardization Program, *About the DSP,* undated. As of September 7, 2017: http://www.dsp.dla.mil

U.S. Department of Defense, Office of the Deputy Assistant Secretary of Defense for Systems Engineering, *MOSA Initiative*, undated. As of September 7, 2017, http://www.acq.osd.mil/se/initiatives/init_mosa.html

U.S. Navy, Office of Naval Research, "Elements of a Technology Transition Agreement (TTA)," undated. As of August 16, 2017: http://www.onr.navy.mil/en/-/media/Files/03T/Elements-of-a-Technology-Transition-Agreement

Zellman, Gail, Joanna Zorn Heilbrunn, Conrad Schmidt, and Carl Builder, "Implementing Policy Change in Large Organizations," in Bernard D. Rostker et al., eds., *Sexual Orientation and U.S. Military Personnel Policy: Options and Assessment*, Santa Monica, Calif.: RAND Corporation, 1993, MR-323-OSD, pp. 368–394. As of August 23, 2018: https://www.rand.org/pubs/monograph_reports/MR323.html