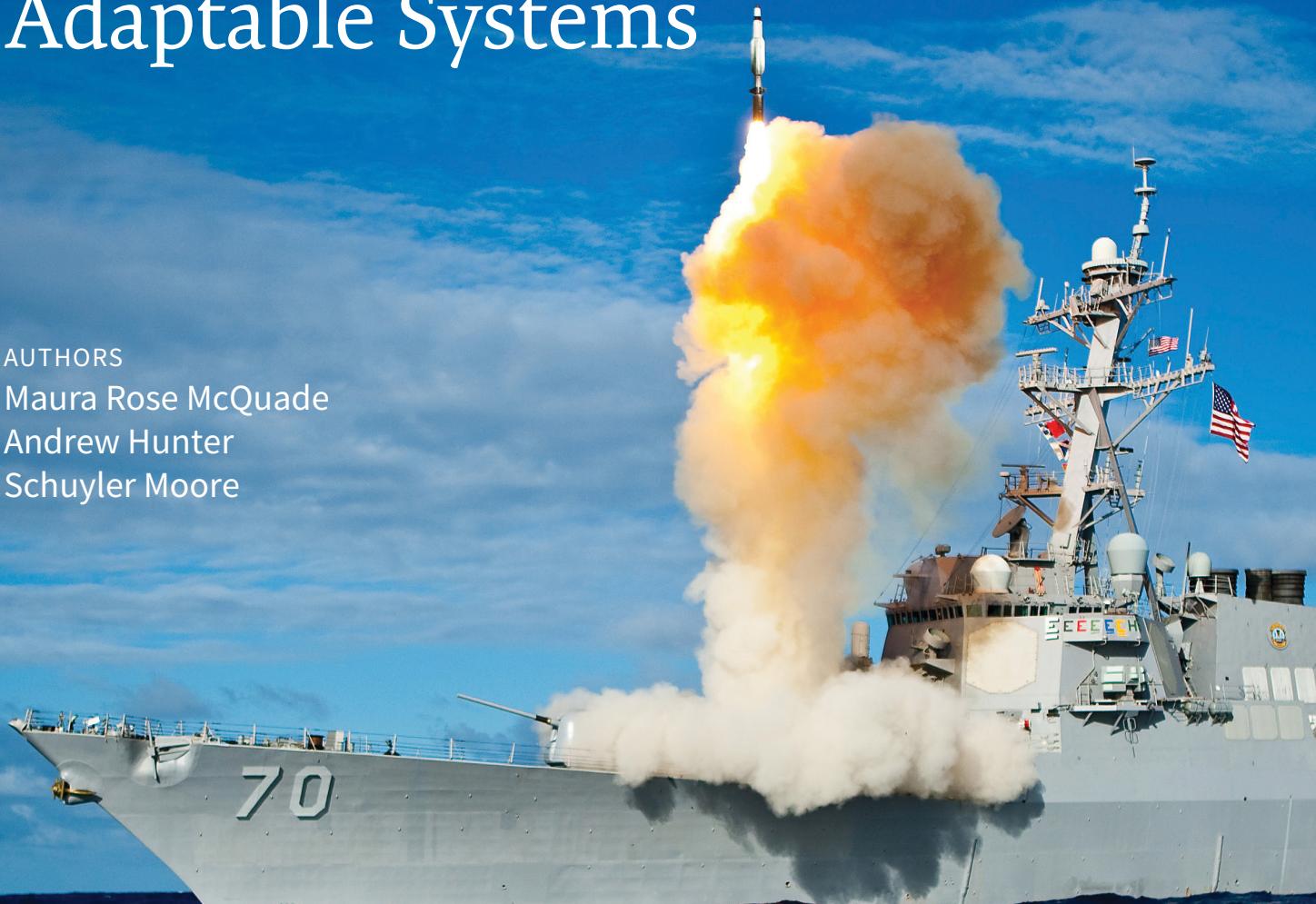


AUGUST 2019

Acquisition of Software-Defined Hardware-Based Adaptable Systems

AUTHORS

Maura Rose McQuade
Andrew Hunter
Schuyler Moore



A Report of the
CSIS DEFENSE-INDUSTRIAL INITIATIVES GROUP

CSIS | CENTER FOR STRATEGIC &
INTERNATIONAL STUDIES

diig
DEFENSE-INDUSTRIAL
INITIATIVES GROUP

AUGUST 2019

Acquisition of Software-Defined Hardware-Based Adaptable Systems

AUTHORS

Maura Rose McQuade
Andrew Hunter
Schuyler Moore

A Report of the CSIS DEFENSE-INDUSTRIAL INITIATIVES GROUP

CSIS | CENTER FOR STRATEGIC &
INTERNATIONAL STUDIES



About CSIS

Established in Washington, D.C., over 50 years ago, the Center for Strategic and International Studies (CSIS) is a bipartisan, nonprofit policy research organization dedicated to providing strategic insights and policy solutions to help decisionmakers chart a course toward a better world.

In late 2015, Thomas J. Pritzker was named chairman of the CSIS Board of Trustees. Mr. Pritzker succeeded former U.S. senator Sam Nunn (D-GA), who chaired the CSIS Board of Trustees from 1999 to 2015. CSIS is led by John J. Hamre, who has served as president and chief executive officer since 2000.

Founded in 1962 by David M. Abshire and Admiral Arleigh Burke, CSIS is one of the world's preeminent international policy institutions focused on defense and security; regional study; and transnational challenges ranging from energy and trade to global development and economic integration. For the past seven years consecutively, CSIS has been named the world's number one think tank for international security by the University of Pennsylvania's "Go To Think Tank Index."

The Center's over 220 full-time staff and large network of affiliated scholars conduct research and analysis and develop policy initiatives that look to the future and anticipate change. CSIS is regularly called upon by Congress, the executive branch, the media, and others to explain the day's events and offer bipartisan recommendations to improve U.S. strategy.

CSIS does not take specific policy positions; accordingly, all views expressed herein should be understood to be solely those of the author(s).

**© 2019 by the Center for Strategic and International Studies.
All rights reserved.**

Center for Strategic & International Studies
1616 Rhode Island Avenue, NW
Washington, DC 20036
202-887-0200 | www.csis.org

Acknowledgments

The authors of this study relied upon feedback received at two workshops held at CSIS, as well as on discussions with other knowledgeable individuals in industry and the defense acquisition community. Particularly important were discussions held with the staff of the Defense Innovation Board who were engaged in the Software Acquisition and Practices Study.

This CSIS study was made possible by the support of Northrop Grumman

Contents

| | |
|---|----|
| Introduction | 1 |
| 1 The Need for Adaptable Systems | 3 |
| <i>What Adaptable Systems Can Bring to Defense</i> | 4 |
| <i>Additional Advantages of Adaptable Systems</i> | 5 |
| <i>The Challenges of Adaptable Systems</i> | 6 |
| 2 Adaptables Systems Usage in Defesne | 7 |
| <i>BACN</i> | 8 |
| <i>SEWIP</i> | 8 |
| <i>Aegis Combat System</i> | 9 |
| <i>JTRS</i> | 11 |
| 3 Barriers to Adaptable Systems | 13 |
| <i>Design of the Traditional Acquisition System</i> | 14 |
| <i>Budgeting</i> | 15 |
| <i>Misaligned Business Incentives</i> | 16 |
| <i>Lack of In-House Technical Expertise</i> | 18 |
| 4 Enablers | 19 |
| <i>MOSA and Adaptable Architecture</i> | 20 |
| <i>Incremental and Iterative Development</i> | 20 |
| <i>Increased User Feedback and Testing</i> | 21 |
| <i>Budgeting for Adaptable Systems and Flexible Funding</i> | 23 |
| <i>Contracting Mechanisms</i> | 24 |
| <i>Dynamic Marketplace</i> | 25 |
| <i>Functionally Aligned Workforce and Increased Training in SW Expertise</i> | 25 |
| 5 Overall Strategy | 27 |
| Conclusion | 29 |
| About the Authors | 30 |
| Appendix A: How Agile Software Development Came to Enable Adptable Systems | 31 |

Introduction

Today's security environment requires the United States to prepare for defense against a wide range of adversaries. The 2018 National Defense Strategy (NDS) emphasizes that the prosperity and the security of our country are both challenged by factors such as the reemergence of long-term strategic competition, a resilient but weakening post-WWII international order, and rogue regimes and non-state actors that destabilize regions critical to international security.¹ Adversaries are adopting and deploying technology in new and innovative ways, challenging the United States to rapidly respond and adapt to a variety of threats. The Department of Defense (DOD) must reexamine almost every facet of its operations in order to assess what changes are required to effectively respond to these new threats. As part of this effort, the acquisition system is rightly considered to be a central element requiring reform.

1. Department of Defense, *Summary of the 2018 National Defense Strategy of the United States of America: Sharpening the Military's Competitive Edge* (DOD, 2018), <https://dod.defense.gov/Portals/1/Documents/pubs/2018-National-Defense-Strategy-Summary.pdf>.

Acquisition reform is a continuous process undertaken in parallel by the Department of Defense and Congress, in pursuit of objectives that are sometimes—but not always—aligned. In light of the NDS, both the DOD and Congress have seen the impetus for acquisition reform shift from a priority on cost control to a new emphasis on speed. This shift, while necessary in many respects, is not sufficient to address the requirements of the NDS. In addressing the need for greater speed, great attention has been given to streamlining, accelerating, and reforming how the acquisition process works. Comparatively less attention, however, has been given to the question of what the process is being optimized to deliver. This problem is critical, because systems capable of responding to the wide range of changing threats identified in the NDS—adaptable systems—face several barriers in the current acquisition system. This paper identifies the need for adaptable systems, their characteristics, the barriers they face in the current acquisition system, the enablers that can allow for their successful development and deployment, and potential changes for the acquisition system resulting from this analysis.

CHAPTER 1

The Need for Adaptable Systems

As the NDS notes, today's security environment is increasingly complex, being defined by rapid technological advancement and the changing character of war, where "the drive to develop new technologies is relentless, expanding to more actors with lower barriers to entry, and moving at accelerating speeds."² This rate of change challenges the U.S. to meet a variety of different threats, which are advancing and changing by the day. The NDS states that "Success no longer goes to the country that develops a new technology first, but rather to the one that better integrates it and adapts its way of fighting."³ The future threat environment suggests that technological superiority or inferiority will not be static; instead, with the rise of peer competitors, defending national security will necessitate the ability to quickly and flexibly leverage areas of strength and mitigate areas of weakness. History demonstrates that technological superiority may not always win wars.; however, refusal to adapt to changing technology will almost always lose wars. Future success is therefore dependent on the nation's ability to adapt and rapidly adjust to threat uncertainties, nimble adversaries, rapidly emerging (and obsolescing) technologies, and new domains.

2. Ibid.

3. Ibid.

Rapid change in commercial technology is a key driver in the strategic environment. Commercial technology development methods have advanced toward more agile processes, which are better suited to meet a rapidly evolving set of user needs and customer demands. This shift is especially true in the area of software, where continuous, iterative software development can harness technology advances, merge previously separate functions, provide continuous upgrades, utilize machine learning, and better leverage user feedback. Potential U.S. adversaries are just as able to use rapidly developing commercial technology to drive adaptability in military operations as are the United States and its allies.

What Adaptable Systems Can Bring to Defense

The U.S. Department of Defense can capitalize on technology trends that have developed to meet rapidly evolving user needs and customer demands through the design of adaptable systems. Adaptable systems are multifunctional, with the inherent ability to deliver a wide variety of capabilities from a single basic design. They are also able to grow, readily adding capability over time at what former defense secretary Jim Mattis would term “the speed of relevance.”

“They [adaptable systems] are also able to grow, readily adding capability over time [growth] at what former defense secretary Jim Mattis would term the “speed of relevance.”

Adaptable systems are not entirely new. In defense, features such as multifunctionality and growth potential were traditionally delivered by very expensive high-end systems, designed to support the addition of sensors and weapons by having excess space, weight, and power built in from the start. One example of this traditional approach to multifunctionality in defense is a Navy ship. Navy ships grew ever larger in the twentieth century to support a wide variety of missions and to address a wide range of threats. In the twentieth century, adaptability was a major cost driver for the most expensive systems in the defense arsenal.

The commercial technology sector has embraced a different approach to adaptable systems, using continuous software development to respond to changing needs. A classic commercial example is the smartphone, which has developed to absorb the functions of many previously separate devices through relatively simple hardware additions, more complex software development, and networking. However, it is becoming increasingly clear

that the characteristics of adaptable systems can be achieved more cheaply and more successfully in the defense sector as well: by writing new software, existing (or slightly modified) hardware can attain multifunctionality and growth.

Today's systems don't require massive scale and expense to achieve adaptability, since the most important elements of functionality are increasingly defined in software and can be modified without substantial changes to the hardware. As a result, a piece of gear that can transmit and receive electrons may be a radio, radar, and electronic warfare asset simultaneously, and it can be upgraded quickly as the technology evolves. These systems are hardware-based, but they are software-defined.

Additional Advantages of Adaptable Systems

While there is a compelling rationale for developing adaptable systems to compete with adversaries who are likely to be attempting to do the same, there are additional, inherent benefits to the use of adaptable systems. Adaptable systems, because they are designed to readily add capability, can speed the deployment of the key new technologies identified in the National Defense Strategy, such as artificial intelligence, autonomy, and robotics. Deploying these technologies in support of military missions requires integrating them in some form into new or existing military platforms, which adaptable systems can enable. Adaptable systems can also reduce risk: their iterative, evolving nature means that individual modifications are continuous and highly incremental, which creates the opportunity to reduce the scope of risk included in any individual upgrade, as well as the ability to fail fast and move on when necessary.

While adaptable systems will present challenges to industry, particularly to prime contractors who will have to manage in a far more dynamic environment, they also bring benefits at multiple tiers of the supply chain. At the level of subsystem suppliers and component developers, adaptable systems can foster enhanced competition, as the frequent modification and upgrade cycles generate new market opportunities on a regular basis. While electronics obsolescence is always a challenge, adaptable systems may be able to effectively avoid or mitigate it in their subsystems and components more effectively, therefore extending their useful service life. Similarly, adaptable systems can ease the process of adapting U.S.-built systems for allied needs and/or incorporating interoperability with U.S. systems into allied equipment. In terms of life cycle costs, individual adaptable systems may not be cheaper to own than systems that hew closer to a static baseline, but in terms of capability delivered per dollar expended, spending efficiency

could be high. Such increased efficiency in the DOD's acquisition spend could translate into savings elsewhere in the overall defense budget.

The Challenges of Adaptable Systems

Adaptable systems are inherently hybrid in nature. Because they are hardware-based—that is, they often have a metal superstructure, such as the array on a radar system—they look like hardware to the acquisition system, and they are generally handled as such. Because they are software-defined, however, it is the 1s and 0s of code that truly generate the bulk of the military capability that they deliver. However, acquisition processes developed solely for software may not address important aspects of what an adaptable system is required to do. Adaptable systems still need to develop sophisticated hardware elements along with their software elements. Consequently, an acquisition system that can successfully leverage the software components of hardware-based systems will be better able to harness continuous development, multifunctionality, and adaptability.

The multifunctionality of adaptable systems also can present challenges due to the interdependent nature of these functionalities. The Defense Science Board notes that “Unexpected complications can arise from unanticipated interdependencies within the software itself, often driven by the underlying architecture. A current DOD acquisition best practice is to reduce project risk by specifying the function of the software in detail at the beginning of a program.”⁴ The more multifunctional and adaptable a system is, the more challenging it is to forecast the scope of its functionality and to predict its interdependencies from the start.

4. Defense Science Board, *Design and Acquisition of Software for Defense Systems* (February 2018), 4.

CHAPTER 2

Adaptable Systems Usage in Defense

Before further discussing the barriers and enablers associated with developing and deploying adaptable systems for military missions, it is useful to examine in greater detail some examples of adaptable systems in defense.

Battlefield Airborne Communications Nodes (BACN)



Source: U.S. Air Force photo by Staff Sgt. Eric Harris

The Air Force's Battlefield Airborne Communications Node (BACN) is an adaptable system which originally leveraged a commercial aircraft base, relatively simple networking nodes, and lots of software to serve as a critical theater network hub, connecting disparate parts of the joint force.⁵ Since its initial development, BACN capability has also been incorporated on unmanned platforms such as the Global Hawk. BACN is the opposite of the elaborate, expensive multifunctional military

platforms of previous decades. It can leverage the inherent ability of software-defined systems to deliver multifunctionality and growth by adding new code, rather than by incorporating additional communications links via adding different hardware, which allows it to connect more systems as needed over time.

BACN provides a communications relay by translating data links and voice systems into a common output. This data sharing contributes to three objectives: it improves interoperability of platforms and systems using disparate communication forms, it allows ground troops to "see" the battlespace beyond the horizon, and it provides improved situational awareness and a common battle picture for all parties in a joint operation. BACN was initially developed as a Quick Reaction Capability (QRC) to address a Joint Urgent Operational Need (JUON), and it was named a Program of Record in 2018. The system was originally meant to be a technology demonstration, but the Air Force was able to accelerate BACN development and field the system, ultimately delivering four integrated BACN systems within 16 months.⁶

Surface Electronic Warfare Improvement Program (SEWIP)

The Surface Electronic Warfare Improvement Program (SEWIP) is an electronic warfare system comprised of radar warning receivers and active jamming systems. It is integrated with a ship's self-defense system,

5. Jennifer Hlad, "More BACN, Please," *Air Force Magazine*, February 17, 2017.

6. "Battlefield Airborne Communications Node (BACN)," Northrop Grumman, <http://www.northrop-grumman.com/Capabilities/BACN/Pages/default.aspx>.

triggering the deployment of decoys and flares in the event of an attack.⁷ SEWIP supports early detection, analysis, threat warning, and protection from anti-ship missiles.



Source: Yasu Osugi (Creative Commons Attribution-Share Alike 4.0 International)

The program uses an “evolutionary acquisition and incremental development” strategy to upgrade each system.⁸ SEWIP is modular, with open architecture, and is upgraded in blocks; SEWIP Block I was focused on obsolescence mitigation and special signal intercept, Block II provided electronic support capability improvement, Block III is in the process of adding electronic attack capabilities, and Block IV will integrate electro-optical and infrared sensing capabilities onto

the existing electronic warfare system.⁹ The most recent upgrade to Block III includes a shift to solid-state digital receivers and transmitters, allowing for more reliability and easier maintenance while making the system more adaptable.¹⁰ SEWIP exemplifies the multifunctionality available in an adaptable system: by primarily using software changes, it is capable of performing electronic warfare, electronic attack, and electronic intelligence functions.

Aegis Combat System

The Aegis Combat System is one of the more high-profile examples of a system shifting from a closed, hardware-dependent structure to an open, software-dependent one. Aegis was first fielded on a commissioned U.S. Navy ship in 1983, and the Navy’s fleet of Ticonderoga-class cruisers and Arleigh Burke-class destroyers have all been outfitted with Aegis. The newest 11 cruisers and the whole fleet of destroyers are undergoing modernization that converts Aegis into an open architecture format, in addition to various

7. Defense Industry Daily staff, “USN Ship Protection: From ‘Slick 32s’ to SEWIP,” *Defense Industry Daily*, May 7, 2018, <https://www.defenseindustrydaily.com/us-navy-from-slick-32s-to-sewip-05365/>.

8. “Surface Electronic Warfare Improvement Program (SEWIP),” U.S. Navy, accessed October 9, 2018, https://www.navy.mil/navydata/fact_display.asp?cid=2100&tid=475&ct=2.

9. Sam LaGrone, “Navy Awards SEWIP Block III Contract to Northrop Grumman,” *USNI News*, February 12, 2015, <https://news.usni.org/2015/02/12/navy-awards-sewip-block-iii-contract-northrop-grumman>.

10. Sydney J. Freedberg Jr., “Navy Forges Ahead With New Surface Ship Electronic Warfare: SEWIP,” *Breaking Defense*, March 20, 2015, <https://breakingdefense.com/2015/03/navy-forges-ahead-with-new-surface-ship-electronic-warfare-sewip/>.



Source: U.S. Navy photo by Mass Communication Specialist 2nd Class Daniel Viramontes/Released

other upgrades.¹¹ Additionally, the USS Arleigh Burke will be the first destroyer to be modernized to merge Aegis open architecture with Aegis Ballistic Missile Defense (BMD) capability, with the goal of ultimately giving the entire destroyer fleet BMD capabilities.¹²

The business model for Aegis' open architecture transition is composed of four parts. First, it requires concurrent development, integration, and testing to upgrade software capabilities. Second, it applies modern software engineering

processes with agile development, rather than the traditional waterfall development. Third, it opens competition up to multiple commercial vendors for individual components of the software. Finally, it leverages points of overlap in capability development across weapons systems.¹³

This process has taken place over multiple decades and ship upgrades. The first step was to implement Commercial Off-The-Shelf (COTS) infrastructure and systems onto cruisers and destroyers to simplify the upgrade process and set a common standard. Next, some systems were broken down into component-based software decoupled from hardware, in order to allow for a layered architecture and spiral development: software upgrades can now occur every two years, while hardware refreshes occur every four. In recent years, more systems within Aegis have been transitioned to this open architecture framework, based on their common set of components and application programming interfaces(referred to as "Baseline 9").¹⁴ As a result of the evolution of Aegis, it now functions as an adaptable system.

The current Aegis modernization program builds on previous upgrades and software developments. The next phases of development will include Aegis Modernization (AMOD) Advanced Capability Build 12 for both destroyers

11. "AEGIS Weapon System: Fact File," U.S. Navy, accessed October 1, 2018, https://www.navy.mil/navydata/fact_display.asp?cid=2100&tid=200&ct=2.

12. Mike Pearn, "Seek and Destroy: The Aegis Combat System," *Naval Technology*, November 12, 2008, <https://www.naval-technology.com/features/feature45460/>.

13. Paul DeLuca et al, "Assessing Aegis Program Transition to an Open-Architecture Model," RAND Corporation, 2013, https://www.rand.org/pubs/research_reports/RR161.html.

14. Jamie Durbin and Richard Scharadin, "The Modernization of the Aegis Fleet with Open Architecture," (Lockheed Martin, May 18, 2011), <http://www.dtic.mil/cgi/tr/fulltext/u2/a557871.pdf>.

and cruisers, with each phase focused on transitioning more components of Aegis to open architecture, therefore allowing increased data sharing and communication between Aegis ships and the rest of the fleet.

Joint Tactical Radio System (JTRS)



Source: Joint Tactical Radio System Rifleman Radio [Image 1 of 3], by Katie Cain/DVIDS (Defense Visual Information Distribution Service)

An example of a program that experienced major challenges, in part because it struggled to develop the characteristics of an adaptable system, is the Joint Tactical Radio System (JTRS). The JTRS program, which was abolished in 2012, sought to develop a set of software-defined radios intended to replace all existing radios in the U.S. military. JTRS sought to enable communication across a range of frequencies and waveforms, allowing increased interoperability both within the U.S. military and with U.S. allies by

converting analogue signals to digital. The JTRS program was built around the Software Communications Architecture (SCA) as an open architecture framework, with the goal of enabling rapid, flexible upgrades. All JTRS components were required to be SCA compliant.¹⁵ The system supported various formats: Network Enterprise Domain (NED), Ground Mobile Radios (GMR, now cancelled), Handheld, Manpack & Small Form Fit (HMS), Multifunctional Information Distribution System (MIDS), and Airborne & Maritime/Fixed Station (AMF). All systems could be upgraded with new software via a wireless information network, allowing for rapid insertion of new technologies across a broad range of systems.

However, the JTRS program faced significant challenges along the way. Although GMR was certified for use in 2012, the Army ultimately cancelled that branch of the program due to cost overruns and technical challenges. When the program first started, software-defined radio technology was in its infancy, but JTRS GMR tried to accomplish too much at one time and the acquisition process was unable to handle its constantly shifting hardware design and software requirements throughout the development

15. "Software-defined radio and JTRS," *Military & Aerospace Electronics*, December 1, 2004, <https://www.militaryaerospace.com/articles/print/volume-15/issue-12/features/special-report/software-defined-radio-and-jtrs.html>.

phase. Furthermore, JTRS failed to adopt an agile approach that would have allowed for user feedback throughout the development process—instead, the program adopted a waterfall methodology that only allowed users to interact with the system after 13 years of development, by which point the problems in GMR were solidified and difficult to reverse.¹⁶ At the same time, developments in commercial software-defined radios technology led industry to develop radios outside the JTRS program that provided capabilities the JTRS program had not been able to deliver. As a result, the JTRS GMR program was terminated in 2011.

Some programmatic descendants of the JTRS program are continuing to move forward. MIDS/JTRS has been successfully integrated onto platforms in the U.S. as well as sold overseas, allowing for increased data interoperability between NATO countries. Both JTRS HMS and AMF have been fielded for low-rate initial production, and their variants continue to be tested.¹⁷

16. Sean Gallagher, "How to blow \$6 billion on a tech project," Ars Technica, June 18, 2012, <https://arstechnica.com/information-technology/2012/06/how-to-blow-6-billion-on-a-tech-project/2/>.

17. Sean Gallagher, "The Army's costly quest for the perfect radio continues," Ars Technica, March 8, 2018, <https://arstechnica.com/information-technology/2018/03/the-armys-costly-quest-for-the-perfect-radio-continues/>.

Barriers to Adaptable Systems

While the case for using adaptable systems in defense is strong and not without precedent, there are reasons why such systems are not widespread. There are substantial barriers to the development and deployment of adaptable systems inherent in the defense acquisition system. It is crucial to understand what these barriers are and how they operate in order to develop an approach to overcoming them.

Design of the Traditional Acquisition System

We must ensure that we adapt the acquisition system to deliver the systems we need, rather than simply optimize it to deliver the wrong systems more quickly or more cheaply.

For the DOD, adaptable systems are essential for fully leveraging the capabilities of existing technologies to meet future warfighting needs. Software-defined adaptable systems will play an increasingly critical role going forward. However, these types of systems test the limits

of the current acquisition system, which is accustomed to working in a much more tightly defined and linear manner. As a result, the DOD has struggled to evolve at the same pace as commercial technology. The defense acquisition system was originally focused on Major Defense Acquisition Programs (MDAPs) with long development cycles, enormous quantities, and tightly defined requirements because the system was designed to provide oversight to high-value hardware systems that were planned to remain in production for decades.

MDAPs almost always begin with highly detailed, highly defined requirements that specify in advance what threats a system is likely to confront and how it is expected to operate in military missions. While useful, this approach introduces the risk of over-specifying systems toward problems, which may morph rapidly over the long development and delivery time scales of defense acquisition.

The DOD 5000.02 acquisition milestone process is designed to progressively reduce technical risk by proceeding through discrete phases of development, test, and evaluation before entering full-rate production.¹⁸ If upgrade increments are planned, they are usually executed serially, not simultaneously. There are high transaction costs for change and high thresholds for justifying a new increment. Communication between the different elements of the acquisition system are organized around acquisition milestones, guided by Milestone Decision Authority (MDA) directives. The stakes for changing program content, such as by adding upgrade increments or revisiting previous decisions, are high and so this happens only rarely.

However, adaptable systems (like other software-oriented development efforts) work best when developed in conjunction with frequent iterative feedback loops throughout the process. Under an adaptable systems approach, acquisition programs would be engaged simultaneously in development, production, and sustainment, which are not easily disentangled for review

18. Department of Defense, *Instruction 5000.02*, January 7, 2015, <https://www.dau.mil/guidebooks/Shared%20Documents/DoDI%205000.02.pdf>.

according to the traditional milestones. Instead, adaptable systems require continuous communication on requirements, budgets, and acquisition benchmarks.

Traditional acquisition metrics can be a major problem for adaptable systems. The Earned Value Management System (EVMS) is a common tool for measuring progress in acquisition programs. It is designed around breaking down a program's master schedule throughout its entire development into discrete work packages, which register as earned value when they are completed at, or below, expected costs. As traditionally implemented, however, EVMS requires an almost entirely static program baseline to function. When the content of work packages is subject to continuous change, the ability of EVMS to meaningfully track progress decays rapidly.

The DOD 5000.02 acquisition system calls for discrete acquisition phases and benchmarks, while adaptable systems utilize fluid development processes. Given this, the traditional approach to acquisition hinders the elements that are critical for the success of an adaptable system.

Budgeting

Current acquisition budgeting also presents roadblocks for adaptable systems, given the defense acquisition system orientation around MDAPs. Budgets for acquisition programs provide prescriptive funding at levels set years in advance, an approach that may be incompatible with the rapidly evolving needs of an adaptable system. Adaptable systems simultaneously consider multiple new and expanded features in each upgrade cycle. The current budget process requires five-year projections for every technology insertion and asks for detailed production and sustainment plans before allocating development resources, a model ill-suited to adaptable systems. For adaptable systems to realize their true promise, the development of their technologies would therefore frequently need to be budgeted outside of an MDAP and then transitioned into a major system, something with precious little evidence of success.

The DOD's budgeting process also includes separate "colors of money" for research and development, production, and operation and maintenance. This is designed to support systems as they move through the acquisition lifecycle; adaptable systems, however, do not progress in a linear way, usually being engaged in development, production, and sustainment simultaneously. While it is entirely possible for programs to budget multiple colors of money at the same time, it is almost inevitable that these budget estimates will not keep pace with program developments in adaptable systems, creating the need for constant reprogramming of funds.



Current acquisition budgeting also presents roadblocks for adaptable system given the defense acquisition system orientation around MDAPs.

The multifunctionality of adaptable systems is also a major challenge for a budget process that organizes around distinct program offices and organizational lines of responsibility. When an acquisition and budget system is accustomed to handling major functions such as communications, battlespace awareness, and electronic warfare as

separate systems—procured by separate offices and using separate budgets—a multifunctional adaptable system is difficult to procure. This aspect of federal budgetary structure means that the DOD is likely to miss front-end opportunities to utilize the inherent multifunctionality of adaptable systems to meet existing and emerging needs with modifications and improvements to adaptable systems, choosing instead to develop new systems or to procure single-function systems. At the same time, there is limited budgetary incentive to explore how adaptable systems can eliminate existing single-function systems, whose capabilities can be merged into existing or emerging adaptable systems.

Misaligned Business Incentives

Business incentives that normally motivate and compensate industry for taking risks are frequently missing when it comes to developing adaptable systems. Traditional government contracts are designed such that prime contractors derive most of their return on investment from delivering hardware. They profit when they build and deliver this hardware themselves, as well as from integrating hardware provided by subcontractors and performing overall systems engineering. There is substantially less profit incentive in developing alternatives to aspects of their existing hardware designs—and even less to developing alternative software approaches for functions that may not even be formal system requirements. Because adaptable systems require contractors to engage in exactly these kinds of activities, they can result in highly misaligned business incentives. Configuration instability and potentially shifting design requirements for adaptable systems can undermine prime contractor profitability and create business uncertainty for first and second tier subcontractors, whose business may be displaced as an adaptable system develops.

Although this incentive problem exists in multiple aspects of adaptable systems, it is a fundamental issue for software development. From a business incentive perspective in government contracts, software development is often an ancillary activity in a profit schedule tied almost entirely to the

Not only are the defense acquisition and budgeting processes misaligned for adaptable systems, but so too are the business incentives within the defense industry.

delivery of hardware. A company that chooses to invest in software development may be able to recover the cost of that work, but it is likely to see no increase in its profit margin as a result of that investment. As a result, industry has little to no incentive to invest in the

development of innovative software that accomplishes previously satisfied requirements better or delivers capability beyond the current requirement. In the commercial sector, industry charges license fees and the potential for ongoing revenue incentivize continuous software development. Commercial firms can also use software development to generate more hardware sales by tying additional software functionality to the purchase of newer models of hardware. These techniques do not translate easily in the defense sector. In defense, hardware sales are low-volume, and there is little opportunity to increase them beyond initial acquisition objectives. Licensing fees are uncommon, because the government is accustomed to paying for software development directly and upfront, rather than indirectly over time. These issues point to the need for an alternative business model for software development in defense.

Additionally, defense prime contractors complain that the adoption of iterative development methods is hampered by DOD contract requirements of documentation, milestone reviews, and incentives, which are based on traditional waterfall models.¹⁹

Rigid contract structures, such as fixed-price development contracts, are a substantial barrier to the development of adaptable systems. Because these contract structures create powerful incentives for the government and the contractor to hew as closely as possible to the original contract terms, the ability to dynamically reshape program content and add capability is effectively precluded.

As RAND's Jonathan Wong has noted, "If the Pentagon wants to reproduce the speedy results of rapid acquisition programs in peacetime, it must find more direct and efficient ways to determine effectiveness that involve the operational user earlier—and not penalize the contractor and the military for going back to the drawing board when something does not work."²⁰

19. Defense Science Board, *Design and Acquisition of Software for Defense Systems* (February 2018), 18.

20. Jonathan Wong, "Don't learn the wrong lessons from rapid acquisition," Defense One, June 23, 2016, <https://www.defenseone.com/ideas/2016/06/dont-learn-wrong-lessons-rapid-acquisition/129332/>.

Lack of In-House Technical Expertise

In-house technical expertise and external partners are equally essential for adaptable systems, both in delivering the necessary technical level of software engineering and in establishing appropriate requirements for software functionality. The DOD has struggled to acquire top software talent, which makes it difficult for all parties to speak in a common language in order to communicate software-based problems. This also makes it challenging to interact effectively with developers and testers to communicate needs, understand opportunities, and test performance, resulting in difficulties at the planning stage as well as additional time for deploying upgrades to operating fleets and training personnel to use them. Software-based systems require not only the necessary software talent, but also an understanding of processes and expectations from both commanders and policymakers. Finally, even as new systems are built to incorporate adaptability, the DOD is faced with the challenges of backward compatibility, cross-system interoperability, and increased variation in existing systems. This complicates both training and sustainment.

Enablers

A variety of enablers exist to overcome or mitigate barriers to adaptable systems adoption. Overall, these enablers encourage earlier and more rapid testing; flexibility in funding; design requirements using base platform/open architecture; and the ability to add on new, interoperable software-based payloads and capabilities, each advancing with iterative and continuous development. They must also incorporate distributed, continual, and agile testing based on shared core architecture, in order to make sure that each update is integrated effectively and that it does not interfere with any other component.

MOSA and Adaptable Architecture

MOSA enables adaptable systems by easing the process of integrating and replacing subsystems and components, as well as by allowing flexibility, competition, and opportunities for distributed development. Architectures that are designed for adaptability from the ground up make flexibility easier, which includes building systems that can easily incorporate new software-defined capabilities. MOSA should be a baseline expectation whenever a system requires adaptability.

Army Major General Bruce T. Crawford has explained that “The industrial base that supports the Department of Defense has been using software to modernize, instead of focusing on just hardware as the mechanism by which they've been able to increase capability.” Software modernization in an open-architecture environment has enabled this approach.²¹

Open standards allow for many different developers to contribute to a system over time, regardless of whether they were involved in the initial system development. This allows for more freedom of innovation and application, due to dispersed development. According to Nick Guertin, senior software systems engineer at the Carnegie Mellon University Software Engineering Institute, MOSA “has helped the Defense Department improve its buying power. It opens up the market opportunities for the greatest possible number of buyers.”²²

In addition, MOSA can help outline possible modernization paths going forward. In the words of Air Force Major General Sarah Zabel, “Open mission systems is a requirement for how every new system is built . . . and we are finding that it's been a great advantage in not only opening us up immediately to a larger part of the industrial base, but also giving us . . . a step by step modernization path.”²³

Incremental and Iterative Development

A variety of tools for incremental and iterative development can be adopted for software-based systems. These include the adoption of commercial software development techniques, such as agile development, DevOps, and development sprints, which enable adaptable systems by providing a

21. Kris Osborn, “Army convenes key parties to discuss software in modern weapons,” Defense Systems, February 27, 2017, <https://defensesystems.com/articles/2017/02/27/solarium.aspx>.

22. Amelia Brust, “Open systems evolving to improve the DOD buying power,” Federal News Network, September 11, 2018, <https://federalnewsnetwork.com/open-systems-2018/2018/09/open-systems-evolving-to-improve-dod-buying-power/>.

23. Katherine Owens, “Air Force pushes faster, more agile IT acquisition,” Defense Systems, November 17, 2017, <https://defensesystems.com/articles/2017/11/17/air-force-it.aspx>.

foundation for iterative change, especially if combined with oversight regimes that eliminate the rigid predictability demands of the current acquisition system. Software-defined systems, if built for flexibility and adaptability, can prolong the effective lifecycle of their base hardware platforms while lowering the cost of technology currency, and they have the potential to simplify hardware sustainment by reducing obsolescence.

The DOD is attempting to realize the benefits of this enabler on the F-35. According to Vice Admiral Mat Winter, Program Executive Officer for the F35, “The current acquisition strategy has us doing a serial [and] sequential design, develop, integrate, test [and] deliver strategy. I’m not convinced that’s the most efficient and effective way, most importantly, to deliver and continuously deliver capability to our war fighters . . . as we go beyond Block 3F.” Winter has worked to develop more of an adaptable systems angle to F-35 upgrades as part of a continuous development and delivery approach. “I am going to be asking the system to do things it’s never done before. I’m asking the system to do true model-based systems engineering simultaneously with capabilities-based testing. The same time. With DT [developmental testing] and OT [operational testing happening at the] same time. Real time. Allowing us to be able to truly change the way we contract and cost estimate.”²⁴



Software-defined systems, if built for flexibility and adaptability, can prolong the effective lifecycle of their base hardware platforms while lowering the cost of technology currency.

of things like prototyping, which can provide real-time testing of systems in warlike environments, and expanded use of virtual twin testing. In virtual twin testing a software “twin” of the system is created and operated in a high-fidelity virtual environment. Deployed systems can then take real-time data and interact in real-time environments. For example, the Navy

Increased User Feedback and Testing

Increased user feedback is necessary for software-based adaptable systems, both in order to improve the functionality of the system and to incorporate the desired changes in real time. Increased feedback loops, a critical part of the agile process, will make sure the product that is delivered is the product the warfighter actually needs. This means increased use

24. Valerie Insinna, “F-35 program office floats new ‘agile acquisition’ strategy,” *Defense News*, September 2, 2017, <https://www.defensenews.com/smr/defense-news-conference/2017/09/06/f-35-program-office-flots-new-agile-acquisition-strategy/>.

currently uses versions of virtual twin testing for its combat systems “so that new technologies can be tested by the crew and commanders before its uploaded into the main combat system, a hedge against reaping unintended consequences by uploading a feature or patch without knowing exactly how it will fit into the ship’s systems.”²⁵ The Army has implemented the use of beta-testing squadrons as well, in order to test out systems in virtual environments before deploying them in Europe.²⁶ The Air Force is using a virtual twin prototyping approach for its program to reengine the B-52 bomber as a primary means of understanding the likely performance of new engines and a key component in the industry competition.²⁷

Increasing user feedback has several benefits. It recognizes that requirements and perceived optimal design may not actually operate as expected or anticipated. Additionally, this process encourages innovation among developers and the user community. As Air Force General Ellen Pawlikowski explains, “Maybe all the requirements aren’t met at the first go, but you have something that you can put in the hands of the operator and they can use it . . . Once you put it in the hands of the operator, maybe some of the requirements you had in the beginning don’t make sense anymore, because [operators] see how they can actually use it and requirements change.”²⁸

This means that the traditional approach, where test and evaluation are treated as a separate phase from development, is incompatible with iterative development. Even as systems are fielded, they will always be in a state of evaluation and upgrade. Air Force Major General Sarah Zabel states, “In order to do that you need to [do] integrated development and test to make sure that what you’re delivering to the field is actually worth delivering to the field.”²⁹

Finally, faster user feedback and real-time testing assists in developing software that can adapt to new environments, revealing problems as they emerge in close to real time. Currently, the time it takes programs to

25. David Larter, “On the new battlefield, Navy must get software updates to the fleet within days, acquisition boss says,” C4ISRNET, September 25, 2018, <https://www.c4isrnet.com/digital-show-dailies/modern-day-marine/2018/09/25/on-the-new-battlefield-the-navy-has-to-get-software-updates-to-the-fleet-within-days-acquisition-boss-says/>.

26. Oriana Pawlyk, “Army Weapons Tests on the Fly in Effort to Win Race Against Russia,” Military.com, October 9, 2017, <https://www.military.com/daily-news/2017/10/09/army-weapons-tests-fly-effort-win-russia.html>.

27. Mandy Mayfield, “Air Force to Release RFP for B-52 Reengineering Program,” *National Defense*, February 28, 2019, <http://www.nationaldefensemagazine.org/articles/2019/2/28/air-force-to-release-rfp-for-b52-reengineering-program>.

28. Katherine Owens, “Agility is the future of software development, says Air Force General,” Defense Systems, July 14, 2017, <https://defensesystems.com/articles/2017/07/14/air-force-cybersecurity.aspx>.

29. Katherine Owens, “Air Force pushes faster, more agile IT acquisition,” Defense Systems, November 17, 2017, <https://defensesystems.com/articles/2017/11/17/air-force-it.aspx>.

assimilate warfighter feedback from the field is too long to incorporate changes into software in a timely manner. The DOD is therefore losing an opportunity to gain advantage.

Budgeting for Adaptable Systems and Flexible Funding

Budgeting for adaptable systems involves multiple aspects. In the first instance, it means recognizing that an adaptable system will not make a linear progression through development to production to sustainment. Rather, the program will be involved simultaneously in all three phases in the budget, with funding to support continuous software development remaining at a fairly constant level throughout most of the system's lifecycle. Although different services have adopted different budgeting strategies for software development, the need for budget mechanisms to support the iterative nature of software is consistent across the DOD.

The Defense Innovation Board has specifically recommended a new category of appropriation for software, covering activities that are currently funded through various appropriations: operation and maintenance; procurement; and research, development, test, and evaluation.³⁰ Compared to the current budgeting system, such a new appropriation would provide substantial flexibility in funding software development and fielding needs, with a minimum of process friction. In addition, existing tools could be modified to reduce the friction currently caused by the need to reprogram funds from one appropriation to another, helping to facilitate agile software development. Helpful measures include clarifying and narrowing the definition of new starts, reducing the rigidity in colors of money so that reprogramming requests are less often necessary, broadening budget justification language to cover broader scopes for research and development, and providing more readily used mechanisms for adjusting color of money.

Budgeting for adaptable systems can also mean creating programmatic space outside of MDAPs for maturing subsystem technologies that may have application across multiple platforms. Congress provided a potential framework for this approach in the National Defense Authorization Act for Fiscal Year 2017 by creating funds in each service for subsystem and component development and prototyping.³¹ This approach would allow military services to budget significant funding for research and development

30. J. Michael McQuade et al, "Software is Never Done: Refactoring Acquisition Code for Competitive Advantage" (Defense Innovation Board, March 21, 2019), 31–32, https://media.defense.gov/2019/Mar/26/2002105909/-1/-1/0/SWAP.REPORT_MAIN.BODY.3.21.19.PDF.

31. National Defense Authorization Act for Fiscal Year 2017, Pub. L. No. 114-328, 114th Cong. (2016), Section 2447d of Title 10 U.S. Code as added by Section 806, <https://www.congress.gov/114/plaws/publ328/PLAW-114publ328.pdf>.

of technologies not directly associated with a program of record (and therefore likely not tied to a program of record requirement). Currently, the Small Business Innovative Research (SBIR) program is one of the only significant sources of R&D funding outside of programs of record, but the SBIR program is not accessible to firms that are not small businesses. Increased use of portfolio-based acquisition management may also enable more technology development outside of MDAPs. The Section 809 panel recommends managing acquisition on a broad portfolio basis rather than focusing on individual programs of record. Such an approach could allow for technology developed in a portfolio to be adopted widely among adaptable systems within the portfolio.³²

Contracting Mechanisms

Contracting mechanisms that best support adaptable systems are likely to be those that foster collaboration between the government and the prime contractor. The more collaboration there is in this relationship, the less effort required to establish tight specifications for every aspect of work. This suggests that it would be challenging, if not impossible, to carry out an adaptable systems program in a fixed price for development contract. Other Transaction Authority agreements (OTAs) and flexible contracting mechanisms, such as multiple award Indefinite Delivery Indefinite Quantity (IDIQ) contracts, can allow for more flexibility in contracting for adaptable systems that can readily add and subtract work scope as needed. Alternatively, fixed price contracts with flexible work scope can allow government and industry to jointly determine what features can be delivered within the time and budget available for the software development effort. In cases where the collaboration may require coordination across large elements of an industrial sector, the use of consortia and alternative competitive constructs may facilitate the coordination and continuous evolution of requirements throughout the acquisition process.

The Section 809 Panel offers recommendations for acquiring technology that is readily available as-is or with modification, which can facilitate contracting for adaptable systems.³³ Similarly, the Defense Innovation Board has proposed streamlined authority-creating software acquisition pathways that can provide a useful mechanism for adaptable systems, particularly for those not originally set up to be adaptable but which are

32. David A. Drabkin et al., *Report of the Advisory Panel on Streamlining and Codifying Acquisition Regulations Volume 3 of 3* (Arlington, VA: Section 809 Panel: 2019), 64–76, https://section809panel.org/wp-content/uploads/2019/01/Sec809Panel_Vol3-Report_JAN19_part-1.pdf.

33. Ibid., 17–48.

transitioning toward an adaptable systems structure.³⁴ More fundamentally, as indicated by Jeff Boleng, the Senior Advisor for Software Acquisition at the Department of Defense, changes to the business model for defense software development will be needed to reward the value delivered by software rather than simply paying for its cost.³⁵

Dynamic Marketplace

A dynamic marketplace approach to working with industry, especially in acquiring technology with strong commercial elements, is recommended by the Section 809 Panel.³⁶ The dynamic marketplace approach involves fostering competition by obtaining proposals from industry prior to establishing discrete performance requirements. The goal of this approach is to leverage commercial innovation and non-traditional partners, placing military missions at the center of government/industry dialogue. Industry consortia can be a good enabler for many of these discussions. The dynamic marketplace approach can support adaptable systems by encouraging commercial practitioners of agile software development to participate in defense acquisition, as well as by reducing the impetus to define highly detailed performance requirements at the front end of acquisition programs.

Functionally Aligned Workforce and Increased Training in SW Expertise

A functionally aligned workforce that is organized to execute iterative software development, along with increased training in software expertise, will enable leadership and allow for greater understanding of the opportunities posed by adaptable systems. With leadership buy-in, the DOD can specify technical career tracks, adjust for competitive talent acquisition, foster cross-service collaboration, develop a broader knowledge of technology across the Department, and offer competitive compensation for potential applicants.

Air Force Chief Technology Officer Frank Konieczny has discussed how the human element is a major factor in the success of agile software development in the Air Force. Turnover in the workforce and challenges in tracking programming skills as part of a career field when making assignments

34. J. Michael McQuade et al., "Software is Never Done: Refactoring Acquisition Code for Competitive Advantage" (Defense Innovation Board, March 21, 2019), Supporting Information Appendix B, https://media.defense.gov/2019/Mar/26/2002105908/-1/-1/0/SWAP.REPORT_SUPPLEMENTARY.DOCX.3.21.19.PDF.

35. Jeff Boleng (remarks presented at The Business of Software Innovation, CSIS, January 29, 2019), <https://www.csis.org/events/business-software-innovation>.

36. *Report of the Advisory Panel on Streamlining and Codifying Acquisition Regulations*, Volume 1 of 3 (January 2018), 7–8.

make it difficult to have personnel continuity and the right mix of skills in pursuing agile software development.³⁷

The DOD must enhance its talent by both leveraging current expertise and attracting and retaining new talent. Specifying technical career tracks and establishing competitive compensation will help significantly. According to the NDS, the DOD plans to “emphasize new skills and complement [their] current workforce with information experts, data scientists, computer programmers, and basic science researchers and engineers—to use information, not simply manage it.”³⁸

Issues with the workforce are not limited to dealing with the development and management of software expertise among those writing and working directly with software. As emphasized in the workforce recommendations of the Defense Science Board study on software acquisition, the DOD also needs to increase software awareness and understanding among program managers and program executive officers, as well as among industry managers.³⁹

Establishing a culture supportive of adaptable systems will take time and will entail taking a different view of risk. According to DIUx Managing Partner Raj Shah: “For us internally, if a team or project team really stretches to try a technology or approach that's really novel but there's technical risk involved . . . technology risk is acceptable and for a certain level we encourage it.”⁴⁰

37. Lauren Williams, “Air Force wants to make ‘Kessel Run’ standard in tech acquisition,” *Federal Computer Week*, October 3, 2018, <https://fcw.com/articles/2018/10/03/usaf-kessel-run-standard.aspx>.

38. Department of Defense, *Summary of the 2018 National Defense Strategy of the United States of America: Sharpening the Military’s Competitive Edge*.

39. Defense Science Board, *Design and Acquisition of Software for Defense Systems* (February 2018), 26.

40. Sean Carberry, “DIUx head wants to drive culture change at the DOD,” *Federal Computer Week*, March 21, 2017, <https://fcw.com/articles/2017/03/21/diu-innovation-carberry.aspx>.

Overall Strategy

While the enablers required for adaptable systems already exist and do not necessarily need new authorities to be implemented, actually combining these tools in an effective and coordinated way remains difficult. It is ultimately essential to understand how these enablers work together and to begin a larger environmental transition toward their use. While elements across the DOD are taking steps to implement a variety of the enablers listed above, the use of many of them is still comparatively rare, and it is even rarer to see several of them used together.



In order to achieve success in the acquisition of adaptable systems, DOD may consider the creation of a clearly defined adaptable systems lane.

In order to achieve success in the acquisition of adaptable systems, the DOD may consider the creation of a clearly defined adaptable systems lane. The DOD currently describes its Adaptive Acquisition Framework⁴¹ as one that includes a variety of approaches, including the Section 804 Middle Tier of Acquisition approach, rapid acquisition,

and traditional acquisition. This framework could be expanded to include an adaptable systems lane as well. Systems in the adaptable systems lane would default to the use of the enablers described above, rather than using them by exception. More traditional approaches could still be used, but they would be the exceptions in the adaptable systems lane. If an adaptable systems lane were created, however, it would be important to ensure that it not monopolize the use of these enablers. The goal of this effort is to enhance the ability of program managers and other acquisition leaders to appropriately use the right tools to acquire adaptable systems, not to impose limitations or straightjackets on them.

41. See the Defense Acquisition University's interactive adaptive acquisition framework tool at <https://www.dau.mil/tools/t/Interactive-Adaptive-Acquisition-Framework-Tool>.

Conclusion

Deploying systems that are adaptable and agile is not just a technology strategy, but a security imperative. Success will ultimately depend on the DOD's ability to adjust rapidly to uncertainty in threats: nimble adversaries, new domains, and unanticipated applications of technology utilizations. Our current acquisition debate is currently failing to directly address the changing nature of what we need to be buying, and as a result, we may be heading toward another round of acquisition reform recriminations in a few years. A successful approach to adaptable system requires using the enablers identified in this report to overcome the barriers to adaptable systems.

About the Authors

Andrew Hunter is a senior fellow in the International Security Program and director of the Defense-Industrial Initiatives Group at CSIS. From 2011 to 2014, he served as a senior executive in the Department of Defense, serving first as chief of staff to undersecretaries of defense (AT&L) Ashton B. Carter and Frank Kendall, before directing the Joint Rapid Acquisition Cell. From 2005 to 2011, Mr. Hunter served as a professional staff member of the House Armed Services Committee. Mr. Hunter holds an M.A. degree in applied economics from the Johns Hopkins University and a B.A. in social studies from Harvard University.

Maura Rose McQuade was a program manager and research associate in the Defense-Industrial Initiatives Group at CSIS from 2017 to 2019.

Schuyler Moore was a research intern in the Defense Industrial Initiatives Group at CSIS in 2018.

Appendix A

How Agile Software Development Came to Enable Adaptable Systems

First executed in 1948, early software was able to complete only the most basic formulas and actions. The range and complexity of software development quickly grew as processing power increased and size of computers decreased, making them accessible to a much wider audience. Growth in personal computers, and, later, mobile devices, further spurred funding for software development as the market for new systems and system applications rapidly expanded.⁴²

Software can be categorized into two types: system software and application software. System software includes all software that manages hardware and connects the user to that hardware, and it ranges from operating systems like Linux to device drivers like those for a mouse or keyboard.⁴³ Application software enables its system to perform a specific function, as in the case of a calculator application, Microsoft Word, or Windows Media Player.⁴⁴ While system software enables the system to operate at a fundamental level, application software can be layered on top of that system software to allow for more specialized tasks.

For both system and application software, the early years of the development process followed a linear, sequential framework—known as “waterfall development”—that set out a strict structure for developers to follow.⁴⁵ As software engineering developed as a field, its methodology was modeled after other engineering fields to help developers carefully trace the stages of development and to fit various projects into a single template. This linear approach hoped to reduce the time required for coding by front-loading effort into the planning process; however, it ignored the fundamental differences

42. Micah Yost, “A Brief History of Software Development,” Medium, January 25, 2018, <https://medium.com/midlandcode/a-brief-history-of-software-development-f67a6e6ddae0>.

43. Alfred Amuno, “The Five Types of Systems Software,” TurboFuture, June 20, 2018, <https://turbofuture.com/computers/The-Five-Types-of-System-Software>.

44. “What is Application Software & Its Types,” Educba, accessed September 25, 2018, <https://www.educba.com/what-is-application-software-its-types/>.

45. Peter Varhol, “To agility and beyond: The history – and legacy – of agile development,” TechBeacon, accessed September 24, 2018, <https://techbeacon.com/agility-beyond-history%E2%80%94-legacy%E2%80%94-agile-development>.

that exist between software engineering and other fields of engineering, as few other fields must account for constantly changing environments and requirements in the way that software does.⁴⁶

Through the 1980s, software engineers began to push back on the idea of waterfall as the most effective method of development. There were concerns that the process was too rigid and that each step was siloed from the rest, creating inefficiencies and lengthening the development cycle. Instead, developers began to advocate an iterative process that allowed interaction and feedback between the various stages of development. In 1986, Barry Boehm of TRW Defense Systems Group published “A Spiral Model of Software Development and Enhancement”, pushing for increased prototyping and software reuse with feedback loops into other stages of development.⁴⁷ His work was widely lauded and, a few years later, James Martin published his Rapid Application Development (RAD) model, which similarly advocated for prototyping based on a looser and more malleable set of requirements.⁴⁸ Both publications laid the groundwork for transitioning software development to an iterative feedback process, and the pressure to change the approach to software continued to grow throughout the 1990s.

Software experienced an “application development crisis” in the mid- to late-1990s. This crisis was born out of the realization that there was an approximate three-year lag between business needs and product application, with some industries (like aerospace and defense) experiencing an even longer gap.⁴⁹ The waterfall development method was put under intense scrutiny as both public and private sector lost patience with its speed and outputs, and in 2001 a group of engineers met in Utah to discuss the future of software development. The result of the meeting was the “Agile Manifesto,” listing twelve principles intended to allow rapid deployment of and rapid feedback on software, and it shifted the standard for software away from the waterfall method toward what is now known as “agile development.”⁵⁰ Since then, the majority of software developers have adopted this methodology and build their products in a more flexible, iterative process.

46. Ibid.

47. Barry Boehm, “A Spiral Model of Software Development and Enhancement” (TRW Defense Systems Group, 1986), <http://csse.usc.edu/TECHRPTS/1988/usccse88-500/usccse88-500.pdf>.

48. Stanley Idesis, “What Is Rapid Application Development?” OutSystems, August 16, 2017, <https://www.outsystems.com/blog/rapid-application-development.html>.

49. Peter Varhol, “To agility and beyond: The history – and legacy – of agile development,” TechBeacon, accessed September 24, 2018, <https://techbeacon.com/agility-beyond-history%E2%80%94-legacy%E2%80%94-agile-development>.

50. Mike Beedle et al, *Agile Manifesto* (2001), <http://agilemanifesto.org/principles.html>.

The “agile” approach is not so much a development method as it is a management strategy. As such, it can be applied to various software development processes, from rapid prototyping, to spiral, to incremental. Agile focuses on iterative development within each stage, as well as real-time communication within a team. The Manifesto of 2001 can be summarized into four main priorities for agile development (emphasis added):⁵¹

1. *Individuals and interactions over processes and tools*
2. *Working software over comprehensive documentation*
3. *Customer collaboration over contract negotiation*
4. *Responding to change over following a plan*

Agile can accommodate flexible specifications through constant design improvement and review with the customer. This enables the software to adapt to the requirements of the product, speeding the fielding process by allowing those requirements to be adjusted as the software is being developed. Various software approaches have evolved to augment agile development, such as DevOps and Scrum: DevOps emphasizes automation as well as continuous integration, testing, delivery, and monitoring to speed up software delivery,⁵² while Scrum attacks a project in a series of pre-planned “sprints” that create a rhythm of coding, testing, integrating, and providing feedback.⁵³

The defense community has been slower to adopt an agile approach and has more frequently stuck to the waterfall method for a number of reasons. The defense acquisition system’s preference for strict procedure and planning does not provide a natural environment for agile development, and the defense industry tends to react more slowly to change than the commercial side, unless there is an imminent national security threat driving funding and mobilizing that change. Agile development promotes proximity and communication between team members working on a project, but the DOD is a massive entity spread across the United States and the rest of the world, traditionally making this sort of environment more difficult to cultivate. Furthermore, the motto of agile development (“fail early and often”) has been perceived as less acceptable to the defense community due to the nature of its work and the consequences of any system failure.⁵⁴

51. Ibid.

52. “What Is DevOps?” New Relic, accessed October 10, 2018, <https://newrelic.com/devops/what-is-devops#Introduction>.

53. “Scrum,” Mountain Goat Software, accessed October 10, 2018, <http://www.mountaingoatsoftware.com/agile/scrum>.

54. Scott Maucione, “DOD doesn’t want to end up like Theranos, so it’s being cautious about ‘fail fast’” Federal News Network, April 26, 2019, <https://federalnewsnetwork.com/de>

There have been efforts to change the DOD approach to software: in 2000, the DOD revised its Instruction, DODI 5000.2 to recommend evolutionary acquisition for software, rather than the traditional structure, to allow for updates and iterative development.⁵⁵ However, while the current DODI 5000.02 generally continues to promote evolutionary acquisition, the details of the document still reflect a hardware-focused mentality that requires rigid program milestones and review cycles.⁵⁶ Some programs have had more success than others in beginning with, or transitioning to, a more agile approach: the Aegis Combat System has successfully transitioned to a more software-dependent, open architecture system over multiple decades, while JTRS GMR attempted to capitalize on nascent software-defined radio technology, although it was ultimately cancelled due to cost overruns and operational challenges.

The push to update DOD software development strategy is largely driven by the fact that current weapon systems, platforms, and IT are unable to keep pace with technology advancement within the existing hardware-oriented acquisition system. The current system promotes slow and methodical movement forward, with emphasis on documentation and siloed development stages, that clearly favors a waterfall development approach.⁵⁷ However, there is a clear need to expedite the fielding of technologies to the warfighter, and with platforms and systems increasingly reliant on software, the agile method provides a potential solution that would enable both faster deployment of systems and more efficient upgrades.

Agile development is not a cure-all and should not necessarily be applied to all programs; the requirements of some systems may require a level of rigidity in the development process, as in the case of initial F-35 development. However, now that the F-35 platform has been proven and tested, future iterations will employ a more agile approach (termed "Continuous Capability Development and Delivery" or C2D2) for future block upgrades.⁵⁸ In recent years, the DOD has increasingly listed "agile development" in its program requirements, and there is evidence that DOD leadership is committed to making agile development the standard

fense-main/2019/04/dod-doesnt-want-to-end-up-like-theranos-so-its-being-cautious-about-fail-fast/.

55. National Research Council, *Achieving Effective Acquisition of Information Technology in the Department of Defense* (National Academies Press, 2010), 48-50, <https://www.nap.edu/read/12823/chapter/5>.

56. Ibid.

57. Lareina Adams, "Agile Software Development in the Department of Defense Environment," Defense Acquisition University, March 31, 2017, <http://www.dtic.mil/cgi-bin/GetTRDoc?Location=U2/1040327.pdf>.

58. Valerie Insinna, "F-35 program office floats new 'agile acquisition' strategy," *Defense News*, September 6, 2017, <https://www.defensenews.com/smrd/defense-news-conference/2017/09/06/f-35-program-office-flots-new-agile-acquisition-strategy/>.

for software, as there have been several reports and directives from the DOD prompting a transition to agile.⁵⁹

The commercial sector has had a faster adoption process, although there are still challenges in shifting to agile development. These include the scalability of software produced through agile development, the geographic distribution of development teams, the need to meet compliance requirements, and the need to establish rigor before the complete set of functional requirements is developed.⁶⁰ Numerous recent reports and studies on how to effectively transition a business from waterfall to agile development suggests that the commercial sector has not fully solved those challenges,⁶¹ but the benefits of agile seem to outweigh any potential growing pains from the transition. The private sector has more flexibility to adjust its teams and team interactions, whereas government structures are slower to adapt to new procedures and methods.

59. Don Johnson, "Enhancing Adoption of Agile Software Development in the DOD [To Improve Acquisition Outcomes]," U.S. Department of Defense, September 23, 2015. <https://www.acq.osd.mil/evm/resources/AgileSep2015/DoD%20AGILE%20DAY%201%20-%20CYBER%20UPDATE.pdf>.

60. Dominic Tavassoli, "Agile software development of military embedded systems," Military Embedded Systems, May 7, 2007, <http://mil-embedded.com/pdfs/Telelogic.May07.pdf>; Jay Thomas, "Can Agile environments accelerate embedded software verification and certification?", Military Embedded Systems, January 27, 2015, <http://mil-embedded.com/guest-blogs/can-agile-environments-accelerate-embedded-software-verification-and-certification/>.

61. Russell Martin, "Changing Your Company's Culture From Waterfall to Agile," Mendix, March 30, 2017, <https://www.mendix.com/blog/changing-your-companys-culture-from-waterfall-to-agile/>.

COVER PHOTO DOD PHOTO COURTESY OF U.S. NAVY/RELEASED



1616 Rhode Island Avenue NW

Washington, DC 20036

202 887 0200 | www.csis.org