United States Government Accountability Office

Report to Congressional Committees

March 2019

DOD SPACE ACQUISITIONS

Including Users Early and Often in Software Development Could Benefit Programs

GAO Highlights

Highlights of GAO-19-136, a report to congressional committees

Why GAO Did This Study

Over the next 5 years, DOD plans to spend over $65 billion on its space system acquisitions portfolio, including many systems that rely on software for key capabilities. However, software-intensive space systems have had a history of significant schedule delays and billions of dollars in cost growth.

Senate and House reports accompanying the National Defense Authorization Act for Fiscal Year 2017 contained provisions for GAO to review challenges in software-intensive DOD space programs. This report addresses, among other things, (1) the extent to which these programs have involved users; and (2) what software-specific management challenges, if any, programs faced.

To do this work, GAO reviewed four major space defense programs with cost growth or schedule delays caused, in part, by software. GAO reviewed applicable statutes and DOD policies and guidance that identified four characteristics of effective user engagement. GAO reviewed program documentation; and interviewed program officials, contractors, and space systems users. GAO also analyzed program metrics, test and evaluation reports, and external program assessments.

What GAO Recommends

GAO is making two recommendations that DOD ensure its guidance that addresses software development provides specific, required direction on the timing, frequency, and documentation of user involvement and feedback. DOD concurred with the recommendations.

View GAO-19-136. For more information, contact Jon Ludwigson at (202) 512-4841 or ludwigsonj@gao.gov.

March 2019

DOD SPACE ACQUISITIONS

Including Users Early and Often in Software Development Could Benefit Programs

What GAO Found

The four major Department of Defense (DOD) software-intensive space programs that GAO reviewed struggled to effectively engage system users. These programs are the Air Force's Joint Space Operations Center Mission System Increment 2 (JMS), Next Generation Operational Control System (OCX), Space-Based Infrared System (SBIRS); and the Navy's Mobile User Objective System (MUOS). These ongoing programs are estimated to cost billions of dollars, have experienced overruns of up to three times originally estimated cost, and have been in development for periods ranging from 5 to over 20 years. Previous GAO reports, as well as DOD and industry studies, have found that user involvement is critical to the success of any software development effort. For example, GAO previously reported that obtaining frequent feedback is linked to reducing risk, improving customer commitment, and improving technical staff motivation. However, the programs GAO reviewed often did not demonstrate characteristics of effective user engagement that are identified in DOD policy and statute:

- **Early engagement.** OCX involved users early; JMS planned to but, in practice, did not; SBIRS and MUOS did not plan to involve users early.
- **Continual engagement.** JMS, OCX, and SBIRS all planned to continually involve users but, in practice, did not fully do so; MUOS did not plan to do so.
- **Feedback based on actual working software.** OCX and SBIRS provided users opportunities to give such feedback but only years into software development; JMS and MUOS did not provide opportunities for feedback.
- **Feedback incorporated into subsequent development.** JMS, OCX, and SBIRS all planned to incorporate user feedback but, in practice, have not done so throughout development; MUOS did not plan to do so.

As reflected above, actual program efforts to involve users and obtain and incorporate feedback were often unsuccessful. This was due, in part, to the lack of specific guidance on user involvement and feedback. Although DOD policies state that users should be involved and provide feedback on software development projects, they do not provide specific guidance on the timing, frequency, and documentation of such efforts. Without obtaining user feedback and acceptance, programs risk delivering systems that do not meet users' needs. In selected instances, the lack of user involvement has contributed to systems that were later found to be operationally unsuitable.

The programs GAO reviewed also faced software-specific challenges in using commercial software, applying outdated software tools, and having limited knowledge and training in newer software development techniques. For example, programs using commercial software often underestimated the effort required to integrate such software into an overall system. Secondly, selected programs relied on obsolete software tools that they were accustomed to using but which industry had since replaced. Finally, GAO found that two of the reviewed programs lacked knowledge of more modern software development approaches. DOD has acknowledged these challenges and has efforts underway to address each of them.

Contents

Letter 1

Background 5
Joint Space Operations Center (JSpOC) Mission System Increment 2 (JMS) 15
Mobile User Objective System (MUOS) 17
Next Generation Operational Control System (OCX) 19
Space-Based Infrared System (SBIRS) 21
Selected DOD Space Programs Have Struggled to Involve Users and Have Infrequently Delivered Software 23
Selected Program Offices Have Had Software-Specific Management Challenges but Are Taking Steps to Address Weaknesses 36
Conclusions 43
Recommendations for Executive Action 43
Agency Comments and Our Evaluation 44

Appendix I Objectives, Scope, and Methodology 46

Appendix II Comments from the Department of Defense 51

Appendix III GAO Contacts and Staff Acknowledgments 53

Related GAO Products 54

Figures

Figure 1: Waterfall Software Development 5
Figure 2: Incremental and Iterative Software Development 7
Figure 3: DevOps Software Development 8
Figure 4: Incremental and Iterative Aspects of Agile Development 9
Figure 5: Critical Software Functions within DOD Space Systems 14
Figure 6: Growing Backlog of OCX Operational Suitability Feedback Points from Users 29
Figure 7: Selected Programs Are Not Delivering Software within Recommended Time Frames 32

Abbreviations

ACAT	Acquisition Category
DevOps	Development and Operations
DDS	Defense Digital Service
DOD	Department of Defense
DODI	Department of Defense Instruction
DSB	Defense Science Board
FAB-T	Family of Advanced Beyond Line-of-Sight Terminals
FITARA	Federal Information Technology Acquisition Reform Act
FFRDC	Federally Funded Research and Development Center
GPS	Global Positioning System
IEC	International Electrotechnical Commission
IEEE	Institute of Electrical and Electronics Engineers
ISO	International Organization for Standardization
IT	Information Technology
JMS Inc. 2	Joint Space Operations Center Mission System Increment 2
JSpOC	Joint Space Operations Center
MAIS	Major Acquisition Information System
MDAP	Major Defense Acquisition Program
MUOS	Mobile User Objective System
NDAA	National Defense Authorization Act
OCX	Next Generation Operational Control System
SBIRS	Space-Based Infrared System
SMDC/ARSTRAT	Space and Missile Defense Command / Army Forces Strategic Command
USSTRATCOM	U.S. Strategic Command

U.S. GOVERNMENT ACCOUNTABILITY OFFICE

441 G St. N.W.
Washington, DC 20548

March 18, 2019

Congressional Committees

Department of Defense (DOD) space systems have grown increasingly dependent on software to enable a wide range of functions, including satellite command and control, early detection and tracking of objects in the earth's orbit, global positioning system (GPS) signals, and radio communication for military forces. Over the next 5 years, DOD plans to spend over $65 billion on its space system acquisitions portfolio, including many systems that rely on software for key capabilities. However, over the last two decades, DOD has had trouble with space acquisition programs where software is a key component, as evidenced by significant schedule delays and billions of dollars of cost growth attributable in part to software problems.

For over 30 years, we have reported on DOD's challenges in acquiring software-intensive weapon systems, including space systems.[1] These challenges include: ineffective management of system requirements, critical software design deficiencies, deferred resolution of problems to later phases of development, inadequate testing of systems, and a lack of meaningful metrics.[2] Congress has mandated DOD to improve its approaches for software development within major defense acquisitions. For example, in 2002, Congress required that each military department establish a program to (1) improve the software acquisition process that includes efforts to develop appropriate metrics for performance measurement and continual process improvement; and (2) ensure that

[1] For the purposes of this report, we use the international standard for software-intensive systems: any system where software contributes essential influences to the design, construction, deployment, and evolution of the system as a whole. International Organization for Standardization / International Electrotechnical Commission / Institute of Electrical and Electronics Engineers (ISO/IEC/IEEE), *International Standard, Systems and software engineering—Architecture description,* 42010 (December 2011).

[2] GAO, *Weapon Systems Annual Assessment: Knowledge Gaps Pose Risks to Sustaining Recent Positive Trends*, GAO-18-360SP (Washington, D.C.: Apr. 25, 2018); *Defense Acquisitions: Stronger Management Practices Are Needed to Improve DOD's Software-Intensive Weapon Acquisitions,* GAO-04-393 (Washington, D.C.: Mar. 1, 2004); *Mission-Critical Systems: Defense Attempting to Address Major Software Challenges,* GAO-93-13 (Washington, D.C.: Dec. 24, 1992); and *Space Defense: Management and Technical Problems Delay Operations Center Acquisition,* GAO/IMTEC-89-18 (Washington, D.C.: Apr. 20, 1989). For more on our previous work in this area, see the related GAO Products page at the end of this report.

key program personnel have an appropriate level of experience or training in software acquisition. In 2010, Congress required that DOD implement processes to include early and continual user involvement, among other things.[3] In 2014, Congress enacted information technology (IT) acquisition reform legislation (referred to as the Federal Information Technology Acquisition Reform Act, or FITARA), which, among other things, requires covered agencies' chief information officers to certify that incremental development is adequately implemented for IT investments.[4]

In response, DOD has made efforts to improve its software development within weapon system acquisitions, such as revising the Department of Defense Instruction (DODI) 5000.02—its instruction for the management of all DOD acquisition programs—in 2015 for programs to use development approaches such as incremental development and to involve users more frequently. In addition, the DODI 5000.02 allows programs to tailor its acquisition procedures to more efficiently achieve program objectives.

Senate and House reports accompanying the National Defense Authorization Act (NDAA) for Fiscal Year 2017 contain provisions for us to review software-intensive DOD space system acquisition programs, among other things. This report addresses, for selected software-

[3] DOD has been directed to improve software acquisition in general and improve user involvement. In particular, the Bob Stump National Defense Authorization Act (NDAA) for Fiscal Year 2003, Pub. L. No. 107-314, § 804 (2002) requires DOD to improve the software-related weapon system acquisition processes with, at a minimum, a documented process for software acquisition planning, requirements development and management, project management and oversight, and risk management; efforts to develop appropriate metrics for performance measurement and continual process improvement; and a process to ensure that key program personnel have an appropriate level of experience or training. In addition, the Fiscal Year 2010 NDAA, Pub. L. No. 111-84, § 804 (2009) requires DOD to implement weapon system acquisition processes that are designed to include early and continual involvement of the user; multiple, rapidly executed increments or releases of capability; early, successive prototyping to support an evolutionary approach; and a modular, open-systems approach.

[4] Carl Levin and Howard P. "Buck" McKeon National Defense Authorization Act for Fiscal Year 2015, Pub. L. No. 113-291, § 831 (codified as amended at 40 U.S.C. § 11319). 40 U.S.C. § 11319(b)(1)(B)(ii) requires that, for covered agencies including DOD, the chief information officers certify that IT investments are adequately implementing incremental development, as defined in capital planning guidance issued by the Office of Management and Budget (OMB). OMB's memorandum M-15-14, Management and Oversight of Federal Information Technology (2015) defines information technology for the purposes of implementing FITARA and sets the standard for adequate incremental development of software or services as delivery of new or modified technical functionality to users at least every 6 months.

intensive space programs, (1) the extent to which these programs have involved users and delivered software using newer development approaches; and (2) what software-specific management challenges, if any, these programs have faced.

We reviewed four software-intensive major defense programs with cost growth or schedule delays attributed, in part, to software development challenges.[5] In selecting these systems from an initial list of 49 DOD space programs, we narrowed our selection to software-intensive Major Defense Acquisition Programs and Major Automated Information Systems as identified by DOD where software development has contributed in some part to cost growth or schedule delays. We further narrowed to those programs that experienced unit cost or schedule breaches or changes and represented different DOD services and acquisition categories.[6] These programs are the Air Force's Joint Space Operations Center Mission System Increment 2 (JMS), Next Generation Operational Control System (OCX), Space-Based Infrared System (SBIRS); and the Navy's Mobile User Objective System (MUOS).[7]

To address the objectives, we interviewed officials from the Undersecretary of Defense for Acquisition and Sustainment, Office of the Deputy Assistant Secretary of Defense for Systems Engineering, Office of

[5] The term ''Major Defense Acquisition Program'' (MDAP) means a Department of Defense acquisition program that is not a highly sensitive classified program (as determined by the Secretary of Defense) and—(1) that is designated by the Secretary of Defense as a major defense acquisition program; or (2) that is estimated by the Secretary of Defense to require an eventual total expenditure for research, development, test, and evaluation of more than $480,000,000 (based on fiscal year 2014 constant dollars) or an eventual total expenditure for procurement of more than $2,790,000,000 (based on fiscal year 2014 constant dollars). 10 U.S.C. § 2430(a); amounts and base fiscal year adjusted on the basis of Department of Defense escalation rates per 10 U.S.C. § 2430(b). Department of Defense Instruction 5000.02, Operation of the Defense Acquisition System (Enclosure 1, Acquisition Program Categories and Compliance Requirements) (Aug. 31, 2018).

[6] The MDAP definitions for significant and critical unit cost breaches are based on unit cost growth as defined in 10 U.S.C. 2433. The Major Automated Information System (MAIS) program definitions for significant and critical changes are based on schedule, cost, or expected performance of the program were defined in 10 U.S.C. 2445c prior to repeal by the National Defense Authorization Act for Fiscal Year 2010, Pub.L. No. 111-84, § 846 (2009).

[7] We initially selected the Family of Advanced Beyond Line-of-Sight Terminals (FAB-T) program but were unable to assess the program's software issues with the same level of detail as the other programs we reviewed because, despite prior software challenges, the program stated it does not have documentation that separately tracks software-related requirements or efforts.

Cost Assessment and Program Evaluation, Office of the Director of Operational Test and Evaluation, Defense Digital Service, Defense Innovation Board, and the Office of the Assistant Secretary of the Air Force for Space Acquisition.[8] We also interviewed officials from the selected program offices and their respective contractors, space systems users, DOD test organizations, and Federally Funded Research and Development Centers.[9]

To determine how effectively selected DOD software-intensive space programs have involved users and adopted newer software development approaches, we reviewed the Fiscal Year 2010 NDAA, in addition to DOD's 2010 report to Congress in response to this statute, and DODI 5000.02, which identified characteristics of user engagement. We then reviewed relevant program plans and documentation—such as human engineering and human systems integration plans, and standard operating procedures—and interviewed program officials and end users to determine the extent to which the program addressed the characteristics. We also examined DOD guidance and applicable leading practices to identify time frames for delivering software under incremental and iterative software development approaches, and we compared these time frames to program performance.

To determine what software-specific management challenges, if any, these selected programs have faced, we reviewed GAO reports and industry reports and studies on software tools and metrics used to manage software programs and also reviewed program management reports, contract documents, and external reports. We also interviewed program and contractor officials and officials from Federally Funded Research and Development Centers. We also reviewed program metrics, test and evaluation reports, and external program assessments. See Appendix I for additional information on our objectives, scope, and methodology.

[8] In 2018, the Office of the Under Secretary of Defense for Acquisition and Sustainment appointed a Special Assistant for Software Acquisition to provide strategic focus and overall policy guidance on all matters of defense software acquisition. In 2015, DOD established the Defense Digital Service (DDS) to influence the way DOD builds and deploys technology and digital services. In 2016, DOD established the Defense Innovation Board to provide independent advice and recommendations on innovative ways to address future challenges in terms of integrated change to technology applications, among other things.

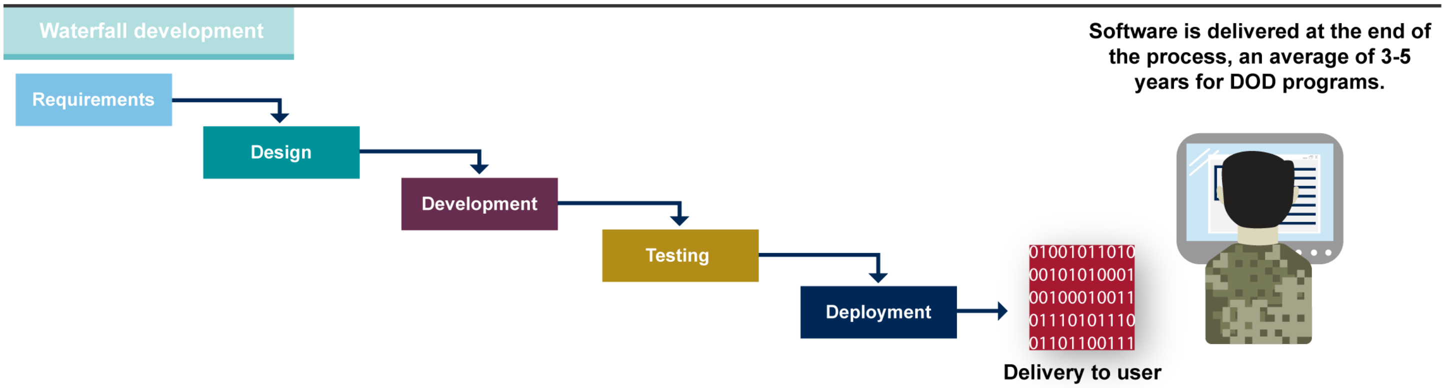
[9] For DOD space systems, users include operators of the system as well as end users of the data produced by the system.

We conducted this performance audit from November 2017 to March 2019 in accordance with generally accepted government auditing standards. Those standards require that we plan and perform the audit to obtain sufficient, appropriate evidence to provide a reasonable basis for our findings and conclusions based on our audit objectives. We believe that the evidence obtained provides a reasonable basis for our findings and conclusions based on our audit objectives.

Background

Software development approaches have evolved over time. DOD weapon system acquisition programs have traditionally developed software using what is known as the waterfall development approach, first conceived in 1970 as linear and sequential phases of development over several years that result in a single delivery of capability.[10] Figure 1 depicts an overview of the waterfall approach.

Figure 1: Waterfall Software Development



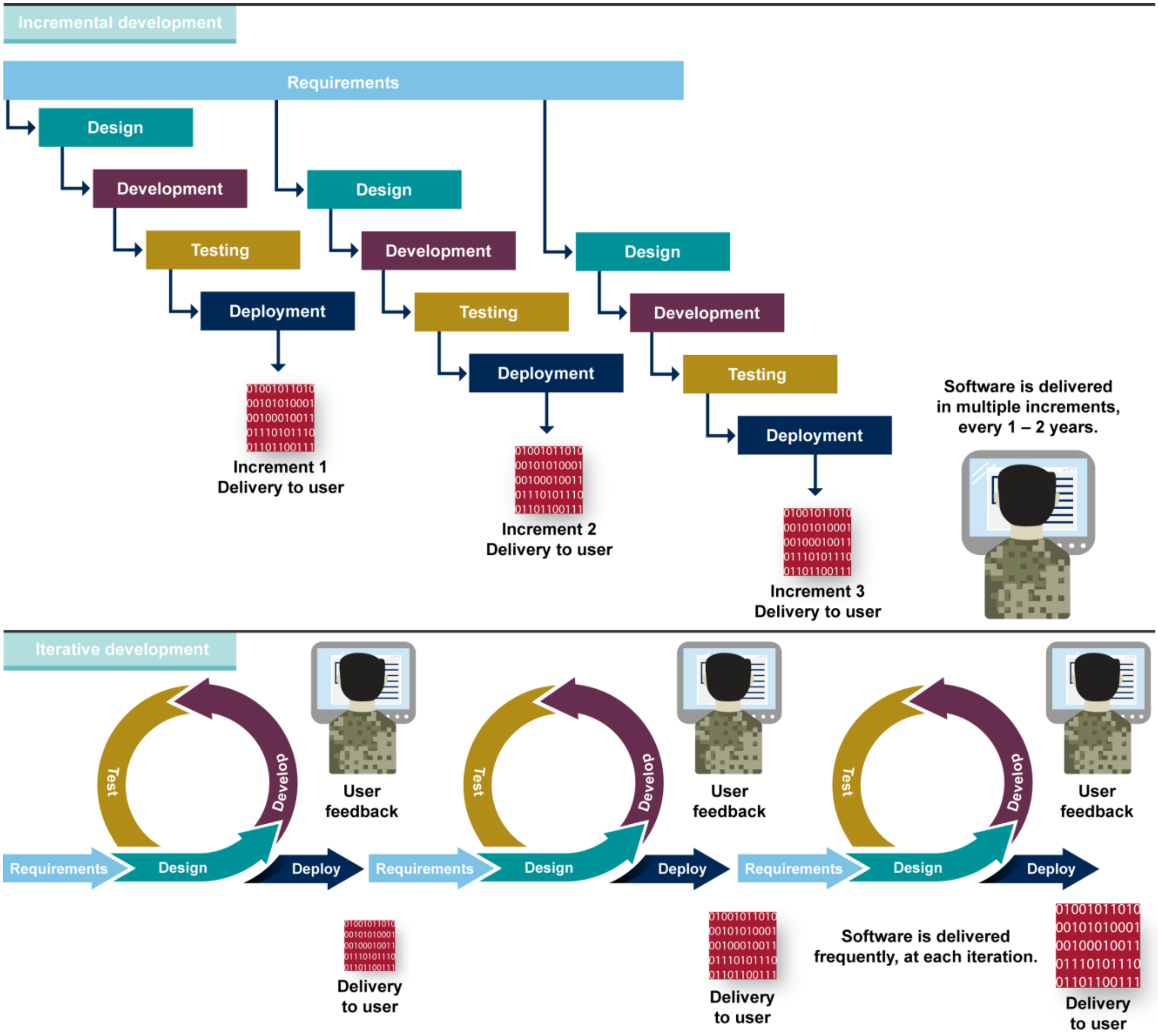
Source: GAO analysis of Department of Defense (DOD) and industry documentation. | GAO-19-136

[10] A 1970 paper entitled *Managing the Development of Large Software Systems,* by Dr. Winston W. Royce, is considered by the Software Engineering Institute and others to be the basis for waterfall methodology. See Royce, Winston. Managing the Development of Large Software Systems. Reprinted from *Proceedings,* IEEE WESCOM (August 1970), pages 1-9. Although the paper never uses the term “waterfall,” the model has sequential phases that flow continuously from one step to the next. While the paper noted that this model is risky because it is unknown how the system will actually work until the testing phase, and recommended iterative interaction between steps, it became the foundation for what is known as the waterfall approach. In 1985, DOD released DOD Standard 2167, which required software development programs to follow a uniform process that mirrored the waterfall methodology.

Within industry, software development has evolved with the adoption of newer approaches and tools. For example, while a traditional waterfall approach usually is often broadly scoped, multiyear, and produces a product at the end of a sequence of phases, an incremental approach delivers software in smaller parts, or increments, in order to deliver capabilities more quickly. This development technique has been preferred for acquiring major federal IT systems, to the maximum extent practicable, and in OMB guidance since at least 2000.[11] In addition, iterative development promotes continual user engagement with more frequent software releases to users. Figure 2 shows an overview of incremental and iterative development.

[11] See Clinger-Cohen Act of 1996, Pub. L. No. 104-106, § 5202 (1996), codified at 41 U.S.C. § 2308 (agencies should use modular contracting, to the maximum extent practicable, to acquire major systems of information technology); see also FAR § 39.103; OMB, Circular No. A-130, Management of Federal Information Resources (Nov. 28, 2000).

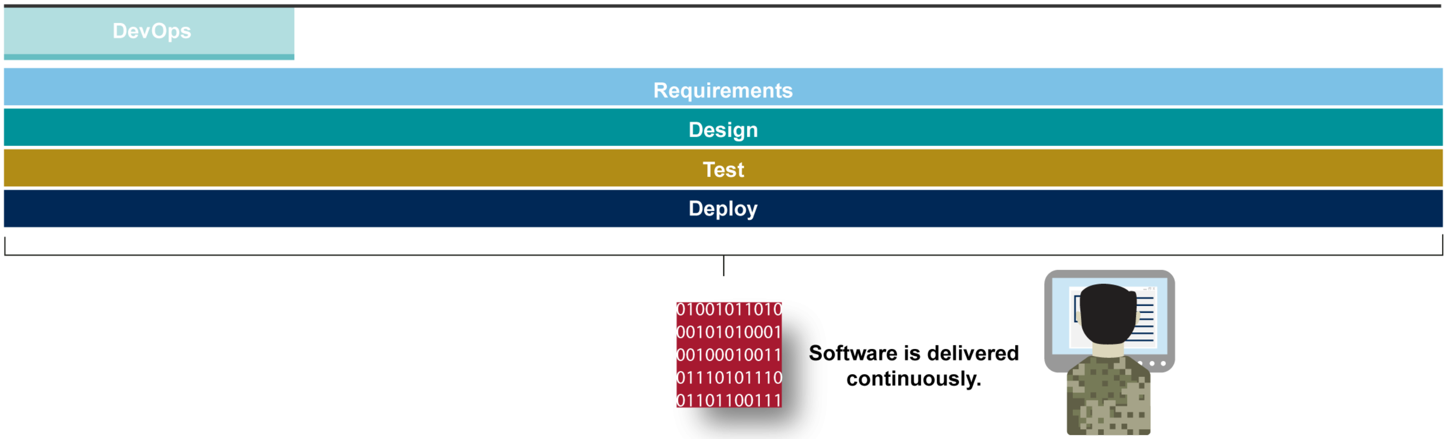
Figure 2: Incremental and Iterative Software Development



Source: GAO analysis of Department of Defense (DOD) and industry documentation. | GAO-19-136

DevOps is a more recent type of software development first used by industry around 2009. According to the Defense Innovation Board, DevOps represents the integration of software development and software operations, along with the tools and culture that support rapid prototyping and deployment, early engagement with the end user, and automation and monitoring of software.[12] Figure 3 shows a notional representation of the DevOps approach based on DOD and industry information. There are also a variety of other software development approaches.[13]

Figure 3: DevOps Software Development



Source: GAO analysis of Department of Defense (DOD) and industry documentation. | GAO-19-136

Incremental, Iterative, and DevOps approaches are further described as follows:

- **Incremental development** sets high level requirements early in the effort, and functionality is delivered in stages. Multiple increments deliver a part of the overall required program capability. Several builds and deployments are typically necessary to satisfy approved requirements. DOD guidance for incremental development for software-intensive programs states that each increment should be delivered within 2 years, and OMB guidance issued pursuant to

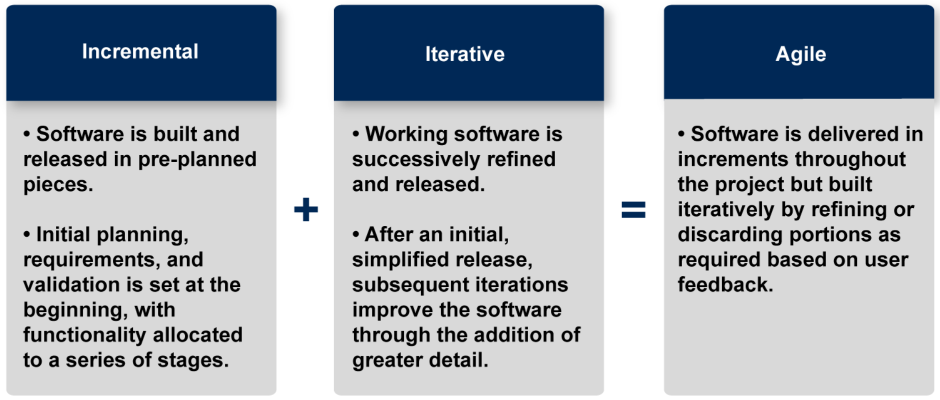
[12] Defense Innovation Board, *Ten Commandments of Software* (Apr. 20, 2018).

[13] Other approaches include Spiral Model, Lean, and others. For the purposes of this report, we focus on the newer approaches that have been adopted by the DOD software-intensive space programs in our review.

FITARA requires delivery of software for information technology investments in 6-month increments.[14]

- **Iterative development** takes a flexible approach to requirements setting. In this approach, requirements are refined in iterations based on user feedback. We include Agile development approaches in this category of development; although most Agile approaches include aspects of both iterative and incremental development, as shown in figure 4. The Agile approach was first articulated in 2001 in what is known as the Agile Manifesto. The Agile Manifesto states the importance of four values: (1) individuals and interactions over processes and tools, (2) working software over comprehensive documentation, (3) customer collaboration over contract negotiation, and (4) responding to change as opposed to following a pre-set plan. Approaches that share common Agile principles include: Scrum, Extreme Programming, and Scaled Agile Framework, among others.[15]

Figure 4: Incremental and Iterative Aspects of Agile Development



Incremental

- Software is built and released in pre-planned pieces.
- Initial planning, requirements, and validation is set at the beginning, with functionality allocated to a series of stages.

+

Iterative

- Working software is successively refined and released.
- After an initial, simplified release, subsequent iterations improve the software through the addition of greater detail.

=

Agile

- Software is delivered in increments throughout the project but built iteratively by refining or discarding portions as required based on user feedback.

Source: GAO analysis of Department of Defense (DOD) and industry documentation. | GAO-19-136

[14] DODI 5000.02 states that Incrementally Deployed Software Intensive Systems software deploys full capability in multiple increments as new capability is developed and delivered, nominally in 1-to 2-year cycles. DODI 5000.02, § 5(c)(3)(d) (Aug. 31, 2018). Pursuant to FITARA, the Office of Management and Budget guidance on information technology investments has described adequate incremental development for software as delivery of new or modified technical functionality to users at least every 6 months. For the purposes of this report, we used the upper end of the DOD range (2 years) in evaluating DOD incremental software deliveries.

[15] Some other key differentiators of Agile software development include: focus on consistent work velocity, focus on high software quality, and focus on maximum defect containment. For more on Agile software development, see http://agilemanifesto.org/.

These approaches stress delivering the most value as early as possible and constantly improving it throughout the project lifecycle based on user feedback.[16] Within industry, Agile development approaches typically complete iterations within 6 weeks, and deliver working software to the user at the end of each iteration.[17] According to DOD and industry, iterative development approaches have led to quicker development at lower costs and have provided strategic benefit through rapid response to changing user needs.[18]

- **DevOps** is a variation of Agile that combines “development” and “operations,” emphasizing communication, collaboration, and continuous integration between both software developers and users.[19] According to the Software Engineering Institute, DevOps is commonly seen as an extension of Agile into the operations side of the process, implementing continuous delivery through automated pipelines. In general, all stakeholders—including operations staff, testers, developers, and users—are embedded on the same team from the project’s inception to its end, ensuring constant communication.[20] Automated deployment and testing is used instead of a manual approach, and the developer’s working copies of software are synchronized with the users. Software code is continuously integrated and delivered into production or a production-like environment. According to industry reports, the use of DevOps may lower costs due to immediate detection of problems as well as result in a greater

[16] This initial delivery is often called a minimum viable product. Within DOD, it may be known as a minimum deployable or minimum releasable product.

[17] The National Defense Industrial Association, International Standards Organization, and other industry studies recommend deliveries of working software within a range of 1 to 6 weeks. The DOD Instruction 5000.02 (Aug. 31, 2018) does not contain guidance for iterative deliveries. For the purposes of this report, we used the upper end of the industry range (6 weeks) in evaluating DOD iterative software deliveries.

[18] Defense Science Board, *Design and Acquisition of Software for Defense Systems* (Washington D.C.: February 2018); Defense Innovation Board, *Ten Commandments of Software* (Apr. 20, 2018); National Defense Industrial Association, *An Industry Practice Guide for Agile on Earned Value Management Programs* (Arlington: Mar. 31, 2017); IBM, *Transitioning from Waterfall to Iterative Development (*Apr. 16, 2004, accessed Dec. 7, 2017), http://www.ibm.com/developerworks/rational/library/4243.html.

[19] There are multiple ways to define DevOps. For the purposes of this report, we used descriptions from DOD, Software Engineering Institute, U.S. General Services Administration, and The MITRE Corporation.

[20] Software Engineering Institute, Software and Cyber Solutions Symposium 2018: DevOps Process & Implementation for Managers and Executives (Mar. 26-28, 2018).

confidence in the software because the users have continuous visibility into development, testing, and deployment.[21]

According to DOD officials from the Undersecretary of Defense, Research and Engineering, adopting Agile and DevOps within DOD weapon system acquisitions—which includes DOD space programs—is challenging and requires programs to adopt comprehensive strategies that cover broad topics. Officials said these strategies should include plans for cultural adoption by the program office and contractor; training and certification for program office and contractor personnel; and tools, metrics, and processes that support continuous integration and delivery, among others.

Collaboration between Developers and Users Is Key to Reducing Program Risk

While there are a variety of approaches to developing software, involving users in early stages and throughout software development helps detect deficiencies early. Industry studies have shown it becomes more expensive to remove conceptual flaws the later they are found.[22] Previous GAO reports as well as other DOD and industry studies have also found that user involvement is critical to the success of any software development effort.[23] For example, we previously reported that obtaining frequent feedback is linked to reducing risk, improving customer

[21] ISO/IEC JTC1/SC7, *DevOps & Agile Study Group Report, Version 1.0,* (May 2017 to April 2018); MITRE, *DevOps for Federal Acquisition,* (2015-2018).

[22] Software Engineering Institute, *Results of SEI Independent Research and Development Projects and Report on Emerging Technologies and Technology Trends*—Technical Report CMU/SEI-2004-TR-018, (Pittsburgh: Oct. 2004); Iosif Alvertis and Sotiris Koussouris, et al, “User Involvement in Software Development Processes,” *Procedia Computer Science,* vol. 97 (2016): 73-83; JC Westland, “The Cost Of Errors In Software Development: Evidence from Industry,” *The Journal of Systems and Software 62,* (2002) p.1-9*;* and Deloitte, *Agile in Government* (2017).

[23] GAO, *Information Technology Reform: Agencies Need to Improve Certification of Incremental Development,* GAO-18-148 (Washington, D.C.: Nov. 7, 2018); *Immigration Benefits System: U.S. Citizenship Immigration Services Can Improve Program Management,* GAO-16-467 (Washington, D.C.: July 7, 2016); *Software Development: Effective Practices and Federal Challenges in Applying Agile Methods,* GAO-12-681 (Washington, D.C.: July 27, 2012); and *Information Technology: Critical Factors Underlying Successful Major Acquisitions*, GAO-12-7 (Washington, D.C.: Oct. 21, 2011). Department of Defense, Office of the Under Secretary of Defense for Research and Engineering; Defense Science Board, *Design and Acquisition of Software for Defense Systems* (Washington D.C.: February 2018); and Software Engineering Institute, *Scaling Agile Methods for Department of Defense Programs*, Technical Note CMU/SEI-2016-TN-005 (December 2016).

commitment, and improving technical staff motivation.[24] We also previously reported that two factors critical to success in incremental development were involving users early in the development of requirements and prior to formal end-user testing.[25]

In the Fiscal Year 2010 NDAA, Congress directed DOD to develop and implement a new acquisition process for information technology systems that, among other things, include early and continuous involvement of the user.[26] This statute, in addition to DOD's 2010 report to Congress in response to the statute, and DODI 5000.02 identify characteristics of effective user engagement for DOD acquisitions, including:

- **Early engagement:** Users are involved early during development to ensure that efforts are aligned with user priorities.
- **Continual engagement:** Users are involved on a regular, recurring basis throughout development to stay informed about the system's technical possibilities, limitations, and development challenges.[27]
- **Feedback based on actual working software:** User feedback during development is based on usable software increments to provide early insight into the actual implementation of the solution and to test whether the design works as intended.

[24] GAO-12-681.

[25] GAO-18-148.

[26] National Defense Authorization Act for Fiscal Year 2010, Pub. L. No. 111-84, § 804 (2009).

[27] Section 804's requirement for early and continual involvement of the user is consistent with the recommendations in Chapter 6 of the March 2009 report of the Defense Science Board Task Force, which notes that enhanced stakeholder engagement and analytical rigor should be throughout the acquisition life cycle. In earlier phases of the acquisition, the program reviews should be quarterly calendar-based events, while later phases should link such reviews with iterations or delivery of multiple, rapidly executed increments/releases of capability, among other things. See Defense Science Board, *Department of Defense Policies and Procedures for the Acquisition of Information Technology* (Washington, D.C.: March 2009).

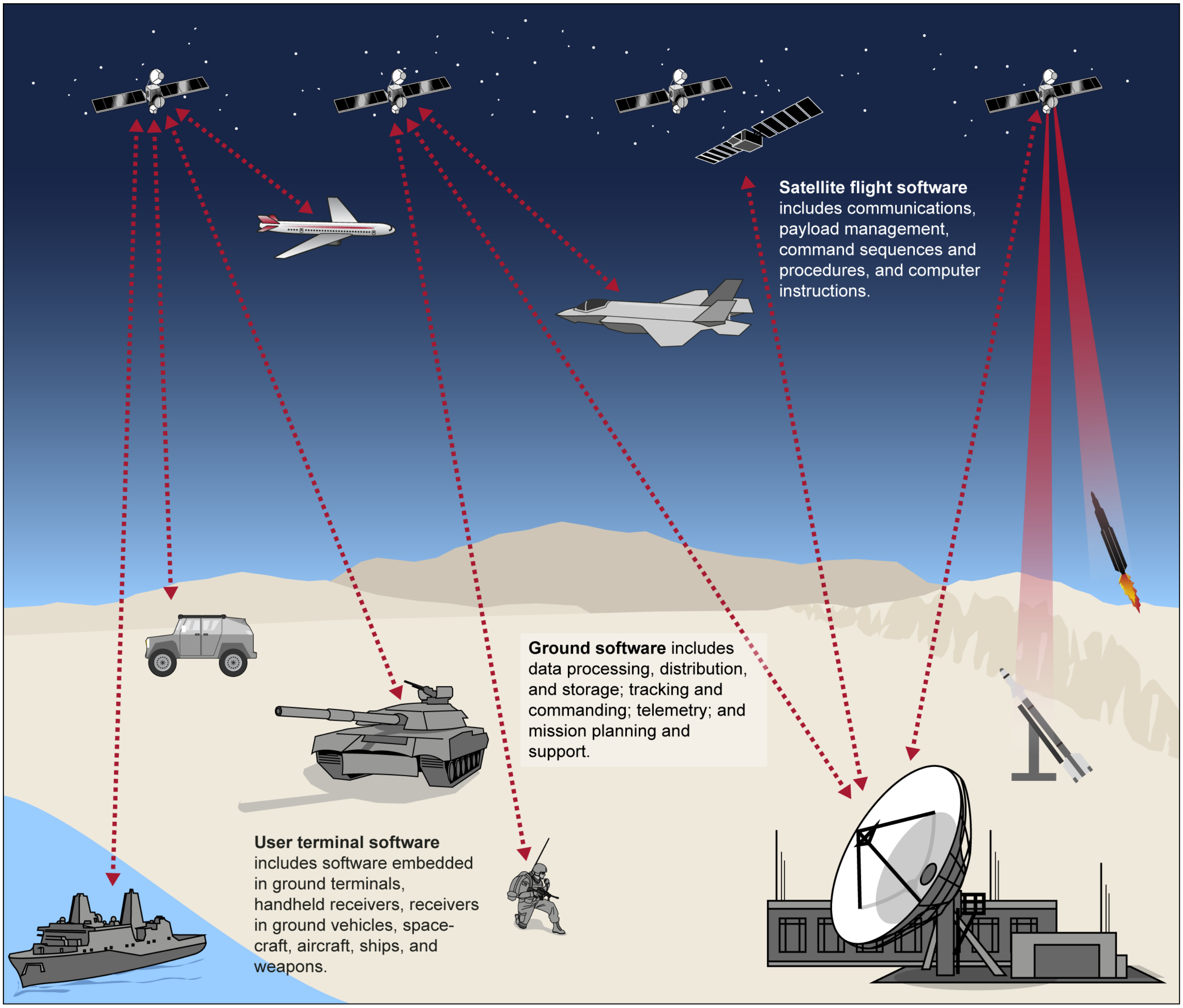
- **Feedback incorporated into subsequent development:** User feedback is incorporated into the next build or increment. [28]

Software Enables Operational Capability in All Segments of Space Systems

Defense space systems typically consist of multiple segments: one or more satellites, ground control systems, and, in some cases, terminals for end-users. Each segment depends on software to enable critical functionality, such as embedded software in satellite vehicles, in applications installed on computer terminals in ground control stations, or embedded signal processing software in user terminals to communicate with satellites, shown in figure 5.

[28] National Defense Authorization Act for Fiscal Year 2010, Pub. L. No. 111-84, § 804 (2009); Office of the Secretary of Defense Report to Congress, *A New Approach for Delivering Information Technology Capabilities in the Department of Defense* (Washington, D.C.: November 2010); DOD Instruction 5000.02, *Operation of the Defense Acquisition System* (Aug. 31, 2018).

Figure 5: Critical Software Functions within DOD Space Systems


Satellite flight software includes communications, payload management, command sequences and procedures, and computer instructions.
Ground software includes data processing, distribution, and storage; tracking and commanding; telemetry; and mission planning and support.
User terminal software includes software embedded in ground terminals, handheld receivers, receivers in ground vehicles, spacecraft, aircraft, ships, and weapons.

Source: GAO analysis of Department of Defense (DOD) documentation. | GAO-19-136

Selected Software-Intensive Space Systems Have a History of Cost Growth and Schedule Delays

We have previously reported on significant cost growth and schedule delays in numerous DOD space systems, with some space program costs rising as much as 300 percent, and delays so lengthy that some satellites spend years in orbit before key capabilities are able to be fully utilized.[29] In particular, the programs described below have experienced significant software challenges, including addressing cybersecurity requirements, which have contributed to cost growth and schedule delays. [30]

Joint Space Operations Center (JSpOC) Mission System Increment 2 (JMS)

The Air Force's JMS program aims to replace an aging space situational awareness and command and control system with improved functionality to better track and catalogue objects in the earth's orbit to support decision making for space forces. Increment 2 is to replace existing systems and deliver additional mission functionality. The Air Force is providing this functionality in three deliveries: the first delivery—Service Pack 7—provided hardware and software updates and was delivered in September 2014; the second delivery—Service Pack 9—aims to improve functions currently being performed, such as determining space object orbits and risks of collision; and the final delivery—Service Pack 11—aims to provide classified functionality.[31] The government is serving as the system integrator directly managing the integration of government and commercially developed software onto commercial, off-the-shelf hardware, so there is no prime contractor.[32]

[29] GAO, *Global Positioning System: Better Planning and Coordination Needed to Improve Prospects for Fielding Modernized Capability*, GAO-18-74 (Washington, D.C.: Dec. 12, 2017); and *Space Acquisitions: DOD Continues to Face Challenges of Delayed Delivery of Critical Space Capabilities and Fragmented Leadership*, GAO-17-619T (Washington, D.C.: May 17, 2017).

[30] In 2018, we reported extensively on DOD’s challenges with addressing cybersecurity for weapon systems. GAO, *Weapon Systems Cybersecurity: DOD Just Beginning to Grapple with Scale of Vulnerabilities,* GAO-19-128 (Washington, D.C.: Oct. 9, 2018).

[31] The Air Force initially planned to deploy JMS as a single increment with five releases and final delivery in 2016. However, in 2011, we recommended, among other things, that DOD ensure that key program risks are fully assessed to help ensure cost, schedule, and performance goals are met. We also noted in the report that implementing this recommendation may require dividing up the program into separate increments, which the Air Force later did in December 2011. See GAO, *Space Acquisitions: Development and Oversight Challenges in Delivering Improved Space Situational Awareness Capabilities*, GAO-11-545 (Washington, D.C.: May 27, 2011).

[32] According to the Air Force, the JMS program is transitioning to provide Space Situational Awareness and Battle Management Command and Control capabilities. For the purposes of this report, we reviewed JMS software development efforts through December 2018.

JMS Inc. 2 Program Essentials

Source: U.S. Air Force. | GAO-19-136

Program office:

Air Force Space and Missile Systems Center, El Segundo, CA

Contractor:

Government is directly managing the integration

Contract Type:

Multiple support contracts

Software Development Approach:

Agile Scrum

Operational Users:

18th Space Control Squadron

614th Air Operations Center

National Space Defense Center

Start of System Development: June 2013

Cost at start of System Development (FY19$M): $1,005.82

Current Cost (FY19$M): $930.39[a]

First Estimated Date of Full Operational Capability: June 2016

Current Estimated Date of Full Operational Capability: TBD

Source: GAO analysis of Department of Defense data.| GAO-19-136

[a]The current estimate reflects de-scoped requirements following a critical change reported to Congress in 2016, due to schedule delays of over 12 months that caused a cost increase of over 25 percent. The critical change reduced the scope and deferred some requirements to later phases in the program.

Historical software development challenges include:

- In 2015, we found that inconsistencies in the program's software development schedule made it unclear whether the program would be able to meet its remaining milestones.[33] The same year, the program declared a schedule breach against its baseline due, in part, to delays in resolving deficiencies identified during software testing.
- In 2016, DOD noted that the revised schedule was still highly aggressive with a high degree of risk because the program was concurrently developing and testing software.
- In 2017, developmental tests found a number of mission critical software deficiencies, which delayed operational testing. The Director of Operational Test and Evaluation also noted that additional work remained to help provide adequate cyber defense for JMS.
- During operational testing in 2018, JMS was found not operationally effective and not operationally suitable due, in part, to missing software requirements, urgent deficiencies that affected system performance, and negative user feedback.

[33] GAO, *Defense Major Automated Information Systems: Cost and Schedule Commitments Need to Be Established Earlier*, GAO-15-282 (Washington, D.C.: Feb. 26, 2015).

Mobile User Objective System (MUOS)

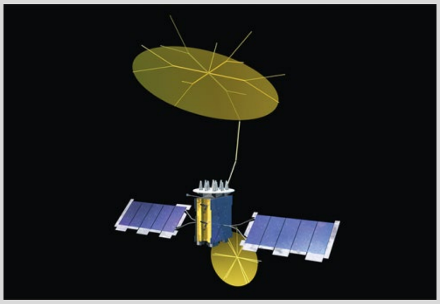
The Navy's MUOS program aims to provide satellite communications to fixed and mobile terminal users with availability worldwide. MUOS includes a satellite constellation, a ground control and network management system, and a new waveform for user terminals.[34] The ground system includes the ground transport, network management, satellite control, and associated infrastructure to both operate the satellites and manage the users' communications. The MUOS constellation is complete, and, according to program officials, software development officially ended in 2012 with the delivery of the waveform software. However, the user community still cannot monitor and manage MUOS. MUOS has two types of users: ground operators responsible for managing the MUOS communications network, and the military users of radios. Space and Missile Defense Command / Army Forces Strategic Command (SMDC/ARSTRAT) was the user representative while MUOS was developed.

While DOD allowed the program to move into sustainment—the phase after development is formally completed—the program continues to resolve challenges with the ground segment, and the contractor continues to deliver software updates to address deficiencies.[35] In 2017, the program transitioned its software sustainment efforts to an Agile development approach in preparation for a follow-on operational test currently scheduled to begin in June 2019. While Lockheed Martin Space Systems is the prime contractor for MUOS, we evaluated software efforts conducted by General Dynamics, the subcontractor performing software development.

[34] Waveforms are software for end-user terminals.

[35] According to program officials, MUOS software has been in sustainment since 2012 but is funding ongoing software efforts as interim contractor support.

MUOS Program Essentials



Source: © 2007 Lockheed Martin. | GAO-19-136

Program office:

Navy Space and Naval Warfare Systems Command, San Diego, CA

Contractor:

Lockheed Martin Space Systems (Prime)

General Dynamics (Software development subcontractor)

Contract Type:

Cost Plus Incentive and Award Fee/Fixed Price Incentive (Firm Target) and Award Fee

Software Development Approach:

Waterfall (Development)

Scaled Agile Framework (Sustainment)

Operational Users:

Naval Computer and Telecommunications Area Master Station Pacific (NCTAMS PAC)

Space and Missile Defense Command / Army Forces Strategic Command (SMDC/ARSTRAT)

Start of System Development: Sept. 2004

Cost at start of System Development (FY19$M): $7,573.58

Current Cost (FY19$M): $7,008.57[a]

First Estimated Date of Full Operational Capability: March 2014

Current Estimated Date of Full Operational Capability: April 2020

Source: GAO analysis of Department of Defense data. | GAO-19-136

[a]The current estimate reflects a reduction of the MUOS satellite quantity from 6 to 5 satellites in 2015.

Historical software development challenges include:

- In 2014, DOD found that 72 percent of the software was obsolete.
- Also in 2014, operational testing was delayed due to software reliability issues in the ground system and waveform.
- In 2015, we found that over 90 percent of MUOS' planned capability was dependent on resolving issues related to integrating the MUOS waveform, terminals, and ground systems.[36]
- Also in 2015, operational tests determined MUOS was not operationally effective, suitable, or survivable due in part to cybersecurity concerns in the ground system.
- As of 2016, there were still existing and emerging cybersecurity vulnerabilities to be addressed.

[36]GAO, *Defense Acquisitions: Assessments of Selected Weapon Programs,* GAO-15-342SP (Washington, D.C.: Mar. 12, 2015).

Next Generation Operational Control System (OCX)

The Air Force's OCX program is designed to replace the current ground control system for legacy and new GPS satellites. OCX software is being developed in a series of blocks: Block 0 is planned to provide the launch and checkout system and support initial testing of GPS III satellites and cybersecurity advancements. Blocks 1 and 2 are planned to provide command and control for previous generations of satellites and GPS III satellites as well as monitoring and control for current and modernized signals.[37] The OCX contractor delivered Block 0 in September 2017. The Air Force took possession of Block 0 in October 2017 by signing a certificate of conformance, and will accept it at a later date after Block 1 is delivered.[38]

[37] GAO has previously reported on OCX challenges. See, for example, GAO, *Weapon Systems Annual Assessment: Knowledge Gaps Pose Risks to Sustaining Recent Positive Trends*, GAO-18-360SP, (Washington, D.C.: Apr. 25, 2018); and *Global Positioning System: Better Planning and Coordination Needed to Improve Prospects for Fielding Modernized Capability,* GAO-18-74, (Washington, D.C.: Dec. 12, 2017).

[38] Per FAR § 46.504, a certificate of conformance may be used in certain instances instead of a source inspection at the discretion of a contracting officer if the following conditions apply: (a) acceptance on the basis of a contractor's certificate of conformance is in the Government's interest; (b)(1) small losses would be incurred in the event of a defect; (b)(2) because of the contractor's reputation or past performance, it is likely that the supplies or services furnished will be acceptable and any defective work would be replaced, corrected, or repaired without contest.

OCX Program Essentials

Source: U.S. Air Force. | GAO-19-136

Program office:

Air Force Space and Missile Systems Center, El Segundo, CA

Contractor:

Raytheon

Contract Type:

Cost Plus Incentive Fee / Cost Plus Award Fee with Cost Plus Fixed Fee line items

Software Development Approach:

Iterative-incremental; DevOps

Operational Users:

2nd Space Operations Squadron

19th Space Operations Squadron

Start of System Development : Nov. 2012[a]

Cost at start of System Development (FY19$M): $3,730.89

Current Cost (FY19$M): $6,247.82

First Estimated Date of Transition to Operations: June 2017

Current Estimated Date of Transition to Operations: April 2022

Source: GAO analysis of Department of Defense data.| GAO-19-136

[a]The Air Force awarded the OCX development contract to Raytheon in February 2010 before completing a milestone B decision formally authorizing the start of development in 2012.

Historical software development challenges include:

- In 2013, DOD paused OCX development due to incomplete systems engineering, which led to continuous rework and deferred requirements.
- In 2015, we reported that, among other things, OCX had significant difficulties related to cybersecurity implementation.[39]
- In 2016, the program declared a Nunn-McCurdy unit cost breach.[40] Also in 2016, the contractor began implementing DevOps at the recommendation of Defense Digital Service but, according to the program office and contractor, only planned to automate development without the operations component of DevOps. The contractor did not achieve initial planned schedule efficiencies.
- In 2017, the Air Force accepted Block 0 despite over 200 open software defects. According to the program, when Block 0 was accepted there was also a plan to resolve the open software defects by the time of the first launch. Since then, according to the program office, all necessary defects related to launch have been addressed.
- In 2018, DOD noted that the schedule was at risk since the program made aggressive assumptions in its plan to develop, integrate, test software, and resolve defects.

[39] GAO, *GPS: Actions Needed to Address Ground System Development Problems and User Equipment Production Readiness*, GAO-15-657, (Washington, D.C.: Sept. 9, 2015).

[40] 10 U.S.C. § 2433, commonly referred to as Nunn-McCurdy, requires the Department of Defense to notify Congress whenever a major defense acquisition program's unit cost experiences cost growth that exceeds certain thresholds.

Space-Based Infrared System (SBIRS)

The Air Force’s SBIRS program is an integrated system of both space and ground elements that aim to detect and track missile launches. SBIRS is designed to replace or incorporate existing defense support ground stations and satellites to improve upon legacy system timeliness, accuracy, and threat detection sensitivity. The Air Force is delivering the SBIRS ground system in one program with two increments: the first increment became operational in 2001 and supports functionality of existing satellites. The second increment, which is still in development, is designed to provide new space segments, mission control software and hardware, and mobile ground capability.[41] The Air Force is delivering these capabilities in multiple blocks: Block 10 was accepted in 2016 and introduced new ground station software and hardware. Block 20 is expected to be complete by late 2019 and is planned to further improve ground station software.[42]

[41] The initial SBIRS architecture included “High” and “Low” orbiting space-based components and ground processing segments. In 2001, the Low component was transferred from the Air Force to the Missile Defense Agency and was renamed the Space Tracking and Surveillance System. The Air Force continues to develop SBIRS High—Geosynchronous satellites (GEO) and Highly elliptical orbit (HEO) payloads—and related ground segment. For the purposes of this report, we evaluated software development in SBIRS ground and applicable flight software development.

[42] The reported costs include the baseline program (GEO 1-4, HEO 1-2 and ground) and the block buy program of GEO 5-6. The Air Force currently reports these programs separately.

SBIRS Program Essentials

Source: U.S. Air Force. | GAO-19-136

Program office:

Air Force Space and Missile Systems Center, El Segundo, CA

Contractor:

Lockheed Martin Space Systems

Contract Type:

Fixed Price Incentive (Firm Target) with Cost Plus Fixed Fee line items

Software Development Approach:

Incremental

Operational Users:

460th Space Wing

Start of System Development: Oct. 1996

Cost at start of System Development (FY19$M): $5,467.97

Current Cost (FY19$M): $19,875.67

First Estimated Date of Full Operational Capability: N/A

Current Estimated Date of Full Operational Capability: N/A[a]

Source: GAO analysis of Department of Defense data. | GAO-19-136

[a] The SBIRS program has not published objective or threshold Full Operational Capability dates.

Historical software development challenges include:

- In 2001, 2002, and 2005, cost increases and schedule delays due, in part, to software complexity problems led to four separate Nunn-McCurdy unit cost breaches.
- In September 2007, we found that the amount of rework resulting from unresolved software discrepancies was contributing to cost growth and schedule delays.[43] In addition, the program had software algorithms that were not yet completed or demonstrated, hundreds of open deficiency reports, and a lack of coordination between space and ground system software databases.
- In 2016, DOD said that software deficiencies were contributing to delays in delivering the ground architecture.
- In 2018, DOD noted that flight software development remained a concern to the overall program schedule. According to SBIRS users and the program office, cybersecurity issues found during Block 10 testing are still being addressed as a part of the Block 20 effort.

[43] GAO, *Space Based Infrared System High Program and its Alternative,* GAO-07-1088R (Washington, D.C.: Sept. 12, 2007).

Selected DOD Space Programs Have Struggled to Involve Users and Have Infrequently Delivered Software

DOD programs we reviewed frequently did not involve users early or continually during development, base user feedback on actual working software, or incorporate user feedback into subsequent software deliveries. Most programs had plans to incorporate these elements of user engagement throughout their software development efforts, but they often did not follow those plans due, in part, to the lack of specific guidance on user involvement and feedback. Regarding frequency of software delivery, while DODI 5000.02 suggests that programs deliver incremental software deliveries every 1 to 2 years, the programs we reviewed often continued to deliver software consistent with the long delivery schedules common to waterfall development. DOD is taking steps to address this issue.

Selected DOD Programs Often Did Not Effectively Engage Users

The four programs we reviewed often did not demonstrate key characteristics of effective user engagement as summarized below:

- **Early engagement.** OCX involved users early and JMS planned to involve users early but, in practice, did not do so; SBIRS and MUOS did not plan to involve users early in software development.
- **Continual engagement.** JMS, OCX, and SBIRS all planned to continually involve users but, in practice, did not fully do so; MUOS did not plan to do so.
- **Feedback based on actual working software.** OCX and SBIRS have provided users opportunities to provide such feedback but only years into software development; JMS and MUOS did not provide opportunities for feedback.
- **Feedback incorporated into subsequent development.** JMS, OCX, and SBIRS all planned to incorporate user feedback but, in practice, have not done so throughout development; MUOS did not plan to do so during software development.

Program efforts to involve users often did not match what their planning documentation described. In addition, when user input was collected, program officials did not capture documentation of how user feedback was addressed. Further, we found that, in practice, none of the programs we reviewed had users providing feedback on actual working software until years after system development began. This was the case even for programs utilizing Agile or iterative-incremental software development approaches, where user involvement and feedback from using functional systems early in the development cycle is foundational.

These shortcomings were due, in part, to the lack of specific guidance on user involvement and feedback. Both DODI 5000.02 and DOD's guiding principles for delivering information technology acquisitions note that software should be developed via usable software deliveries to obtain user acceptance and feedback for the next segment of work, but this guidance lacks specificity. In particular, DOD does not specify when to involve users and request their feedback, how frequently to seek user involvement and feedback on software deliverables, how to report back to users on how that feedback was addressed, and how to document the results of user involvement and feedback.[44]

As a result of programs' shortcomings with user involvement and feedback, programs risk delivering systems that do not meet user needs. In selected cases, delivered software was deemed operationally unsuitable by DOD testers and required substantial rework.

Further details on the extent to which programs implemented the four key characteristics are described below.

JMS: Program documents created at the start of JMS system development contain specific operating procedures for conducting interactions with the user community—Air Force personnel who track and catalogue objects in orbit—during acquisition and fielding. However, the program has not followed these operating procedures during system development.

- **Early Engagement.** The JMS program office planned to involve users early in development but, in practice, did not do so. JMS program documentation states that users were to be involved in user engagement sessions within the first 4 weeks of iterative development.[45] However, the first documented user engagement session was held more than a year after development start.
- **Continual Engagement.** The JMS program office planned to engage users throughout development but, in practice, did not do so. JMS program documentation states that user engagement sessions are to

[44] DOD Instruction 5000.02*, Operation of the Defense Acquisition System* (Aug. 31, 2018); Office of the Secretary of Defense Report to Congress, *A New Approach for Delivering Information Technology Capabilities in the Department of Defense* (Washington, D.C.: November 2010).

[45] According to JMS documentation, user engagement sessions are to be held within and associated with service pack sprints.

be held regularly during development—roughly every 2 to 4 weeks. However, in practice, program officials told us they only involved users as needed during software development. We found that the frequency of user engagement events varied from several weeks to more than 6 months. According to program officials, there were limited users available, and their operational mission duties were prioritized over assisting with system development.

- **Feedback Based on Actual Working Software.** The JMS program office did not provide users an opportunity to give feedback based on actual working software during development. According to program documentation, designs and notional drawings, not working software, were to be used for user engagement sessions. While JMS did provide users opportunities to provide feedback, this feedback was not on actual working software. Program officials said the goal of these events was never intended to include user feedback on actual working software. However, users told us that when they were finally able to use the system for the first time, 4 years after development started, it did not function as needed. The software did not execute what it had been designed to do, and earlier user engagement on actual working software may have identified these issues.
- **Feedback Incorporated Into Subsequent Development.** The JMS program office planned to incorporate user feedback into development but, in practice, did not do so. JMS program documentation states that the program will document user feedback from user engagement events using summary notes communicated back to the user. However, JMS users said it was often unclear if their feedback was incorporated. For example, in March 2016, a user engagement event was held to discuss any questions and concerns relating to the planned system's conjunction assessment—a key feature that predicts orbit intersection and potential collision of space objects—that resulted in 8 user-identified issues. When we met with the users in 2018, they told us that conjunction assessment issues remained unaddressed, and they would still be reliant on the legacy system to fully execute the mission and perform their duties. The legacy system is still needed, they said, because the program deferred critical functions, and the most recent operational test found the system to be operationally unsuitable.

What is Operational Suitability?

Operational Suitability defines the degree to which a system is satisfactorily placed and operated in field use. Consideration for Operational Suitability includes, among other things:

- Reliability
- Availability
- Compatibility
- Transportability
- Interoperability
- Maintainability
- Safety
- Human Factors
- Manpower and Logistics Supportability
- Documentation
- Training Requirements

Source: GAO analysis of DOD and Industry Documentation | GAO-19-136

MUOS: The MUOS program office did not engage users—Army Forces Strategic Command personnel who support the narrowband and wideband communications across the Air Force, Marines, Navy, and Army—during software development but are engaging users while developing software during sustainment, the acquisition phase after development when the program mainly supports and monitors performance.[46] Following the end of development, at an operational test event in 2015, DOD testers deemed the system was operationally unsuitable. The MUOS program office moved to an Agile development approach in 2017 to address software deficiencies in preparation for the next operational test event.

- **Early Engagement**. The MUOS program office did not engage users early in development. Program documentation does not describe any plans for user engagement or involvement during development and, according to program officials, no users evaluated the actual system during development.
- **Continual Engagement.** The MUOS program office did not continually engage with users. Program documentation does not describe any plans for user engagement or involvement during development. Program officials said no users evaluated the system during development because there were no users with real world experience on a system like MUOS. However, as previously noted, SMDC/ARSTRAT represented end users' interests during MUOS development.
- **Feedback Based on Actual Working Software.** The MUOS program office did not provide users an opportunity to give feedback based on actual working software. Program documentation does not describe a process for obtaining user feedback based on actual working software. The first time users had a chance to fully operate the system was after development ended, in preparation for operational testing in 2014, which identified numerous defects. Additionally, MUOS users said that they have since identified 128 functions in 11 critical areas that must be addressed or they will not accept the system.[47] Users

[46] For the purposes of this report, MUOS assessment reflects user engagement characteristics during software development, although the program has since incorporated Agile approaches during software sustainment. Since then, the program engages users on working software and incorporates their feedback into the next iteration of development. Space and Missile Defense Command / Army Forces Strategic Command (SMDC/ARSTRAT) represented military service users across the military components during MUOS system development, but not system operators at the Naval Computer and Telecommunications Area Master Station Pacific. According to program officials, there were no MUOS System Operators (Network Management personnel) with real world experience on a system like MUOS during development.

[47] These items were based on a Joint MUOS Readiness Tracker, which was developed by both users and the program office to track all tasks that need to be completed across the program and operational command prior to full operational acceptance. The tasks are aligned to the system updates needed to return to operational test.

also said that some of the vulnerabilities found during operational testing, including cybersecurity vulnerabilities, have been deferred.

- **Feedback Incorporated Into Subsequent Development.** The MUOS program office did not incorporate user feedback into development. Program documentation did not describe plans to gain user feedback or acceptance into the development of the MUOS system. In addition, users and the contractor told us that program officials did not allow direct interaction during development due to a concern that such interactions could lead to changes in system requirements. The program office said that user involvement to-date has not caused delays to testing or software delivery.

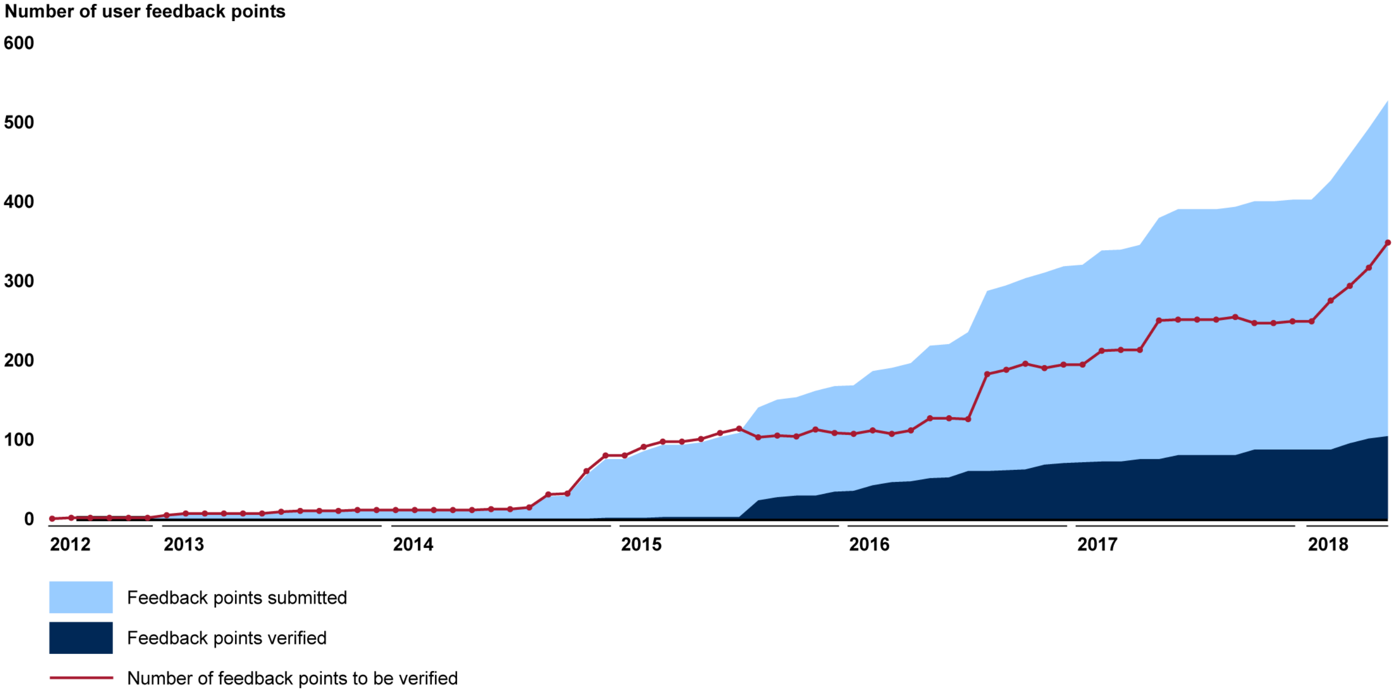
OCX: The OCX program had limited user engagement, but has recently held user engagement events based on releases of actual working software. The program has made efforts to obtain feedback from users, but users have noted there is no time in the schedule to address much of their feedback prior to delivering the system.

- **Early Engagement.** The OCX program office involved users early in development in accordance with its plans. From 2011, OCX users were involved in technical meetings where they provided feedback on the concept of operations and the design of the system.
- **Continual Engagement.** The OCX program office planned to engage users throughout development but, in practice, did not fully do so. OCX planning documentation includes multiple opportunities for user engagement at various stages of system development, including operational suitability and "hands-on" interaction with an integrated system.[48] According to the program office, numerous events were held for users to give feedback on the system. However, since 2012, the program has only held one of its planned events to address operational suitability. In addition, other opportunities for users to operate the system have been removed to accommodate the program's schedule, such as "day in the life" events that allowed users to validate the system as they would actually operate it. Users said that removing events like these created fewer opportunities to identify and resolve new deficiencies.

[48] According to the OCX Human Engineering Program Plan, comments are to be received from three primary event types or triggers: (1) active ("hands on") operator interaction on functional portions of OCX, or the integrated OCX system; (2) operator (non "hands on") participation in scheduled OCX operator engagement events; and stakeholder unsolicited recommendations.

- **Feedback Based on Actual Working Software**. OCX did not plan to provide users an opportunity to give feedback based on actual working software but, in practice, did so years into development. OCX planning documents rely on simulations and mock-ups for evaluating system usability. However, users told us that mock-ups do not allow them to test functionality and may not be representative of the final delivered product. Starting in 2014—2 years after development started—users had opportunities to review the limited functionality available at the time. Since 2017, users said they were able to test working software.
- **Feedback Incorporated Into Subsequent Development.** The OCX program office planned to incorporate user feedback into development but, in practice, did not do so throughout development. OCX planning documentation includes a user comment response process that would collect and validate user comments and communicate results back to the users. According to the program office, for OCX Block 0, users provided feedback that was incorporated prior to the first launch. While OCX users said that they have the opportunity to provide feedback, there is a growing list of unaddressed Block 1 issues to be resolved. Some of these feedback points, if left unresolved, may result in operational suitability concerns and a delayed delivery to operations. According to the program office, critiques from the users have either been closed, incorporated into the OCX design, or are still under assessment between the contractor and users. A majority of user feedback points for the OCX iteration currently in development remain unresolved, as depicted in figure 6. In 2016, DOD told the Air Force and the contractor to utilize DevOps. As previously noted, DevOps is intended to release automated software builds to users in order to unify development and operations and increase efficiency. The contractor stated it implemented DevOps in 2016. However, both the Air Force and the contractor admitted in 2018 they never had plans to implement the “Ops” side of DevOps, meaning they didn’t plan to automatically deliver software builds to the users. Without incorporating the users and experts in maintainability and deployment, the program is not benefiting from continuous user feedback.

Figure 6: Growing Backlog of OCX Operational Suitability Feedback Points from Users



Source: GAO analysis of OCX data. | GAO-19-136

SBIRS: SBIRS users—Air Force personnel who operate, command, and control SBIRS satellites to detect and track missile launches—were not involved during early system development and the program only recently increased the frequency of user events. SBIRS users have been able to provide feedback on working software but are unaware how this feedback is incorporated into software development.

- **Early Engagement**. The SBIRS program office did not engage users early in development because users were not in place and user groups were not defined. The program planning documentation that instituted the framework for user involvement was not in place until 2004.[49] According to SBIRS users and test officials, this resulted in a poor interface design and users being unable to respond adequately to critical system alerts when using the system. Though the program

[49] The SBIRS Human Factors Engineering Program Plan was updated in 2004 to define the use of Operational Design Teams, which are used to apply user-centered design principles from the design effort through test, evaluation, and verification.

contractor told us that user involvement is critical for ensuring the developers deliver a system that users need and will accept, DOD officials said that users were not integrated with the development approach until the software was ready to be integrated into a final product.

- **Continual Engagement.** The SBIRS program office planned to engage users throughout development but, in practice, did not do so. SBIRS planning documentation includes users involved in regular working groups throughout development. SBIRS users began to be involved with system development in 2013 on a weekly basis. Users were not involved during the 17 years of system development prior to this time.
- **Feedback Based on Actual Working Software.** The SBIRS program did not plan to provide users an opportunity to give feedback based on actual working software during development but, in practice, did so years into development. SBIRS documentation only outlines user engagement as reviewing and commenting on design plans. While users were able to provide feedback on working software in 2017, these events did not occur until 21 years after the start of development when the software was ready to be integrated. When users were able to provide feedback, they identified issues with the training system and cybersecurity.
- **Feedback Incorporated Into Subsequent Development.** The SBIRS program planned to incorporate user feedback into development but, in practice, did not do so. SBIRS planning documentation includes methods for users to provide feedback, but users said there is no feedback loop between them and the developers; therefore, users are unaware if their comments and concerns are addressed or ignored.

Selected Programs Have Generally Not Delivered Software Frequently, but DOD Is Taking Steps to Improve Efforts

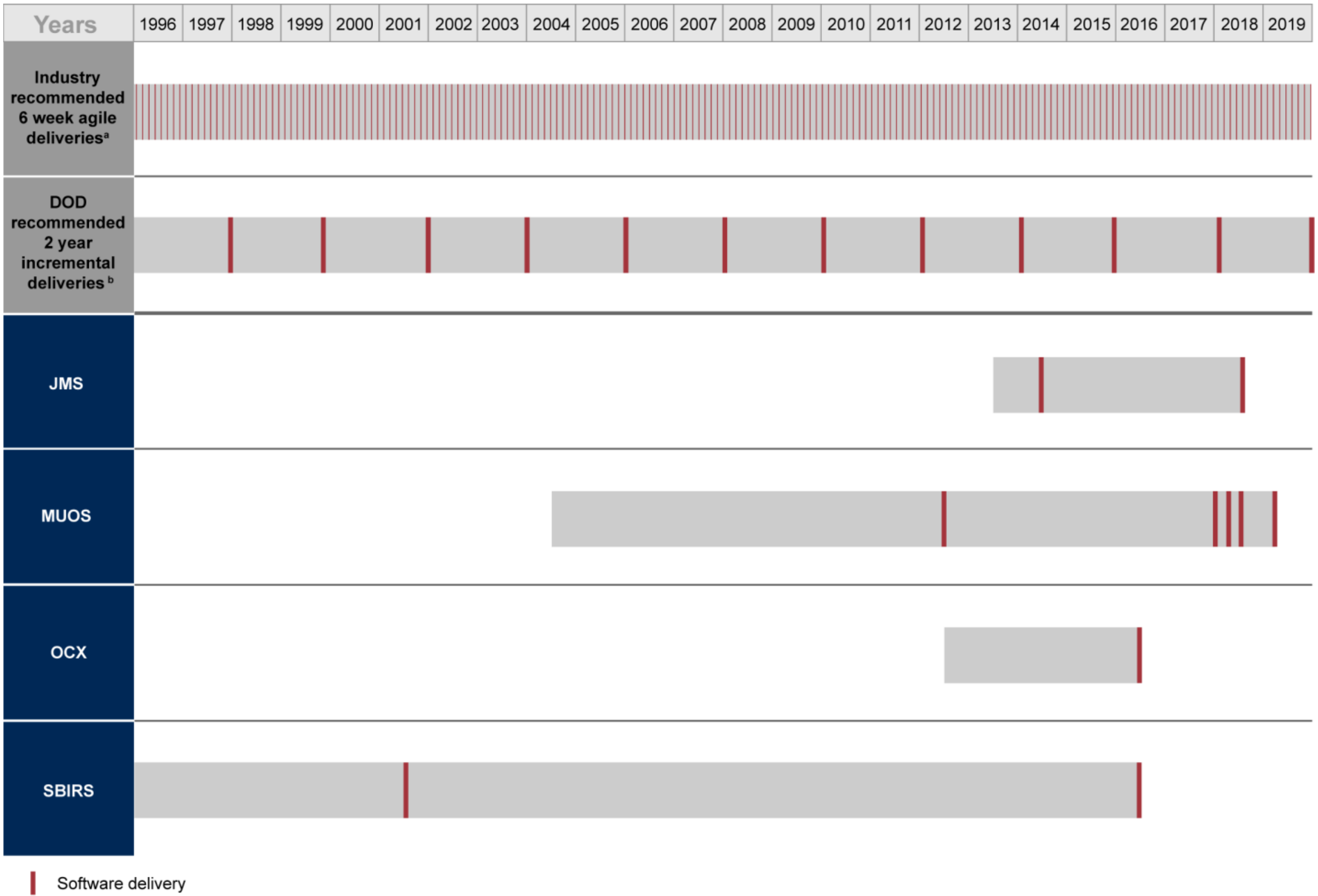
DOD officials and DODI 5000.02 point to the benefits of delivering smaller packages of software more frequently, but the four programs we examined have generally delivered them infrequently. DOD is beginning to take steps to address these issues, such as establishing an independent advisory panel and considering recommendations issued by the Defense Science Board on the design and acquisition of DOD software.

Selected programs continue to focus on infrequent deliveries. According to industry practices, short, quick deliveries allow a program to deliver useful, improved capabilities to the user frequently and continually throughout development. Within industry, iterations for Agile development approaches are typically up to 6 weeks, and working software is delivered to the user at the end of each iteration. In addition, DODI 5000.02 states that for incremental development increments should be delivered within 2 years.

While two programs in our review—JMS and MUOS—say they have undertaken elements of Agile development, which emphasize smaller deliveries of frequent software to users, they still struggled to move away from the long delivery schedules common to waterfall development. In addition, the two programs with incremental development—OCX and SBIRS—have not delivered within suggested DOD time frames.[50] See figure 7 below for program software deliveries.

[50] DODI 5000.02 states that Incrementally Deployed Software Intensive Systems software deploys full capability in multiple increments as new capability is developed and delivered, nominally in 1-to 2-year cycles. DODI 5000.02, § 5(c)(3)(d) (Aug. 31, 2018). The OCX program is using an iterative-incremental software development approach. For the purposes of this review, we compared the program with both iterative and incremental recommended time frames.

Figure 7: Selected Programs Are Not Delivering Software within Recommended Time Frames



JMS = Joint Space Operations Center Mission System
MUOS = Mobile User Objective System
OCX = Next Generation Operational Control System
SBIRS = Space-Based Infrared System

[a]Multiple industry studies recommend Agile deliveries within 1 to 6 weeks. We evaluated programs against the upper limit of this range.

Source: GAO analysis of Department of Defense (DOD) and industry documentation. | GAO-19-136

Further observations on each of the four programs follow:

- JMS program officials and documentation indicate that the program is using an Agile development approach to deliver smaller, rapid deliveries to minimize risk. According to JMS program documentation, software releases were to be delivered in 6-month intervals. However, the program only delivered actual working software once during development—a delivery of capability in 2014. The program was operationally accepted in late 2018. However, only 3 of 12 planned capabilities were accepted for operational use.

- The MUOS program used a traditional waterfall approach during development from 2004 to 2012 and has only had one overall software product delivery during that time. The program completed the software in 2012, yet continued to make changes during sustainment using the waterfall methodology and adopted an Agile approach in 2017 to address deficiencies. Since this adoption, it has delivered software more frequently—about every 3 months. This is a significant improvement over the delivery time frames during the MUOS waterfall development approach.
- The OCX program is using an “iterative-incremental” development approach. According to OCX software development plans, this approach was to enable early and frequent deliveries of capabilities. Specifically, the program plans for iterations to be completed every 22 weeks. However, since software development began in 2012, OCX has delivered just one increment of software, referred to by the OCX program as a block.
- The SBIRS program began in 1996, using a waterfall approach, and has had two deliveries of software. SBIRS Increment 1 was delivered in 2001, and the next increment, SBIRS Increment 2, Block 10, was delivered 15 years later, in 2016. The next increment, SBIRS Increment 2, Block 20, is expected to be delivered in 2019.

Part of the reason programs delivered larger software packages less frequently was the adherence to the process steps in the DODI 5000.02 that were designed under the waterfall approach. While DODI 5000.02 authorizes programs to tailor their acquisition procedures to more efficiently achieve program objectives, none of the programs that were trying to employ a newer development approach took steps to tailor procedures in order to facilitate development. For example, the OCX contractor said it was delayed by complying with technical reviews under a military standard for traditional waterfall approaches, such as the Preliminary Design Review, Critical Design Review, and others, but the OCX program did not alter these reviews, despite having flexibility to do so. The contractor told us a more tailored approach would enable execution of smaller iterations of software deliverables. Similarly, the JMS program office noted that it was not fully able to integrate Agile development practices because of all the different technical reviews, but JMS did not tailor these requirements to more efficiently achieve outcomes, despite flexibility to do so.

DOD officials have acknowledged these challenges and have recently begun recommending steps to address them. Officials we spoke with from Defense Digital Service, Director of Operational Test and Evaluation, and DOD leadership said that rapid development of software using newer software practices does not fit with the requirements of the DOD acquisition process. Further, DOD's Special Assistant for Software Acquisition said that DOD software development should be iterative, providing the critical capabilities in smaller, more frequent deliveries rather than delivering capabilities in a single delivery via traditional waterfall software development. In addition, other DOD officials we interviewed agreed that since DOD programs may not always know the full definition of a system's requirements until late in development, additional flexibility to tailor acquisition approaches could improve software acquisitions.

In acknowledging the challenges in moving from a waterfall model to a more incremental approach, various DOD groups have made recommendations to support delivery of smaller, more timely software deliverables:

- In February 2018, the Defense Science Board issued a series of recommendations to support rapid, iterative software development. The recommendations included requiring all programs entering system development to implement iterative approaches and providing authority to the program manager to work with users.[51]
- In April 2018, the Defense Innovation Board made recommendations to improve DOD software acquisitions, such as moving to more iterative development approaches that would deliver functionality more quickly.[52]
- In June 2018, the DOD Section 809 Panel recommended eliminating the requirements for Earned Value Management (EVM)—one of DOD's primary program planning and management tools—in Agile programs.[53] However, other DOD and industry guides state that Agile

[51] Department of Defense, Office of the Under Secretary of Defense for Research and Engineering, Defense Science Board, *Design and Acquisition of Software for Defense Systems* (February 2018).

[52] Department of Defense, Defense Innovation Board, *Ten Commandments of Software*, Version 0.14 (April 2018).

[53] Section 809 Panel, *Report of the Advisory Panel on Streamlining and Codifying Acquisition Regulations*, vol. 2 (June 2018).

programs can still report EVM if certain considerations are made, such as an Agile work structure that provides a process for defining work and tracking progress of this work against planned cost and schedule.[54]

Pursuant to the Fiscal Year 2019 National Defense Authorization Act, DOD is required, subject to authorized exceptions, to begin implementation of each recommendation submitted in the final report of the Defense Science Board Task Force on the Design and Acquisition of Software for Defense Systems by February 2020. For each recommendation that DOD is implementing, it is to submit to the congressional defense committees a summary of actions taken; and a schedule, with specific milestones, for completing implementation of the recommendation.[55] We intend to monitor DOD's progress in implementing the recommendations.

[54] National Defense Industrial Association, *An Industry Practice Guide for Agile on Earned Value Management Programs*, Ver. 1.1 (2017).

[55] John S. McCain National Defense Authorization Act for Fiscal Year 2019, Pub. L. No. 115-232, § 868 (2018).

Selected Program Offices Have Had Software-Specific Management Challenges but Are Taking Steps to Address Weaknesses

The programs we reviewed faced management challenges using commercial software, applying outdated software tools and metrics, and having limited knowledge and training in newer software development. DOD is taking steps to address these challenges.

Selected DOD Programs Face Difficulties Identifying the Effort Required by and Mitigating the Risk Associated with Commercial Software

DOD has previously encouraged DOD acquisition programs to use commercial software where appropriate. For example, in 2000 and in 2003, DOD policy encouraged considering the use of commercial software. In addition, regulations continue to emphasize consideration of commercial software suitable to meet the agency's needs in acquiring information technology. DOD officials said that, although the effort to maintain commercial software may be equivalent to developing such capabilities in-house, programs should still consider the use of commercial software because DOD and its contractors may lack the technical skillsets to develop a similar product.

What is Commercial Software?

Software that is ready-made and commercially available to the public.

Examples:

Microsoft Windows

Red Hat Enterprise Linux

VxWorks

Source: GAO Analysis of DOD and Industry Documentation | GAO-19-136

However, three of the programs we reviewed had difficulty integrating and maintaining modified commercial software during development:

- The JMS acquisition approach was to only use commercial and government-provided software with no new software development planned, but the commercial products selected were not mature and required additional development, contributing to schedule delays.
- The MUOS program underestimated the level of effort to modify commercial software, which increased cost and introduced schedule delays in completing both the ground system and the waveform. According to an Aerospace official who advised the program on software issues, the MUOS software development approach was to use a commercial software solution but with substantial modifications.[56] In particular, the MUOS contractor planned to take a

[56] The Aerospace Corporation is a Federally Funded Research and Development Center (FFRDC). FFRDCs are unique nonprofit entities sponsored and funded by the U.S. government to meet some special long-term research or development need which cannot be met as effectively by existing in-house or contractor resources.

commercial cellular system and substantially modify it for MUOS. This official, along with the MUOS program office, said that underestimating the level of effort to modify and integrate the commercial software has been the program's biggest challenge.

- In September 2015, we found that the OCX contractor was overly optimistic in its initial estimates of the work associated with incorporating open source and reused software. Further, according to the Air Force, OCX program managers and contractors did not appear to follow cybersecurity screening or software assurance processes as required.[57] For example, open source software was incorporated without ensuring that it was cybersecurity-compliant. These problems led to significant rework and added cost growth and schedule delays to address the cybersecurity vulnerabilities and meet cybersecurity standards. In addition, in an independent assessment of OCX, officials from the MITRE Corporation said that there is a lack of appreciation for the effort required for commercial software integration, stating that the level of effort is "categorically underestimated."[58]

Some program officials noted that commercial software updates led to system instability and increased costs. For example, OCX program officials said that updating an operating system version led to 38 other commercial software changes. Each of these changes had to be configured, which took considerable time and added cost to the program. Similarly, the SBIRS contractor said they have been concerned that updates to commercial software could create a domino effect of instability, and the risks could outweigh the benefits of the update. For example, if one commercial software product is updated and becomes unstable, instability may be introduced to other commercial software products and software components. On the other hand, not updating software products could lead to cybersecurity concerns. As we previously noted, developers of commercial software generally update software to address identified flaws and cybersecurity vulnerabilities.[59] We also reported in a review of weapon systems cybersecurity that, although there

[57] GAO-15-657.

[58] The MITRE Corporation is a FFRDC. In accordance with the Fiscal Year 2017 NDAA, MITRE conducted an Independent Program Assessment of the OCX program for DOD. National Defense Authorization Act for Fiscal Year 2017, Pub. L. No. 114-328, § 1622 (2016).

[59] GAO-19-128; and *Information Security: Effective Patch Management is Critical to Mitigating Software Vulnerabilities*, GAO-03-1138T (Washington, D.C.: Sept. 10, 2003).

are valid reasons for delaying or forgoing weapon systems patches, this means some weapon systems are operating, possibly for extended periods, with known vulnerabilities.[60]

In addition, the lifecycles of commercial software can contribute to management challenges when these products become obsolete. For example, in 2014, a MUOS Ground System Deep Dive review identified that 72 percent of the MUOS software was considered to be obsolete.[61] According to program officials, commercial software became obsolete before or soon after it was fielded, especially for operating systems and browsers, due to the long MUOS development cycle. Software obsolescence is also among the top risks of the OCX program and has contributed to additional costs during development.

DOD officials and others have started to acknowledge challenges in using commercial software. For example, as we previously reported in 2018, DOD has stated that many weapon systems rely on commercial and open source software and are subject to any cyber vulnerabilities that come with them.[62] While DOD states that using commercial software is a preferred approach to meet system requirements, some program officials we interviewed told us that the effort to modify and update commercial software is underestimated. DOD is working on helping programs understand commercial software risks. For example, in January 2018, DOD published a Guidebook for Acquiring Commercial Items. In addition, Defense Acquisition University offers several modules designed to address challenges in integrating commercial solutions.

[60] GAO-19-128.

[61] Obsolescence refers to the process or condition by which a piece of equipment, or software, becomes no longer useful, or a form and function which is no longer current; or is no longer supported by the supplier or original equipment manufacturer.

[62] GAO-19-128.

Selected DOD Programs Are Using Outdated Software Tools and Metrics but Are Updating Them

Three of the DOD programs we reviewed have experienced challenges in using outdated software tools or identifying appropriate performance metrics as they transition to newer software development approaches.

What are Software Tools?

Software Tools: Tools to aid in the development of software code creation, integration, and testing.

Source: GAO analysis of Industry Documentation | GAO-19-136

Contractors continue to rely upon outdated software tools and experience challenges. We found that three of the programs we reviewed used tools that are considered outdated and lack the flexibility needed for iterative development. Contractors for three of the four programs we reviewed have experienced software development challenges due to outdated tools:

- The SBIRS contractor uses a suite of tools that is considered outdated for newer commercial approaches. For example, one of these tools relies on a central database that, if corrupted, will stop development work and could take days or weeks to fix. According to the contractor, fixing this database has led to multiple periods of downtime and schedule delays.
- The MUOS contractor also uses a toolset that is considered outdated by commercial software development experts. The program moved to a newer Agile development approach in 2017 but has retained an older software development toolset. The MUOS contractor said they are heavily reliant on these tools for development and do not anticipate changing the toolset.
- The OCX contractor also uses tools that are considered outdated by commercial approaches. According to the contractor, these tools have been in place for many years, and switching over to a new set of tools would not be in the best interest of the program because it could be disruptive to ongoing development. Defense Digital Service experts said that a particular suite of tools used by the OCX contractor is outdated because the tools lack the flexibility needed for iterative development.
- Both MUOS and SBIRS contractors said that they have had to train new employees to use their outdated tools. For example, the SBIRS contractor told us that when new employees begin work on the SBIRS program, they already know how to use newer tools but have to be trained on the outdated tools used for SBIRS development.[63] The SBIRS contractor said this has affected retention of its workforce in

[63] For example, newer software development practices use a set of tools such as configuration management software, continuous integration systems, code repositories, and issue tracking systems.

some cases, and the program has allocated funding to transition to newer tools in order to better recruit and retain personnel.

What is Cloud-Based Testing?

Cloud-based testing uses cloud computing environments to simulate an application's real-world usage.

According to international standards, cloud testing can lead to cost savings, improved testing efficiency, and more realistic testing environments.

Source: GAO analysis of Industry Documentation | GAO-19-136

Two contractors have taken steps to update their software tools to increase automation and cloud-based testing but have not yet experienced the anticipated efficiencies:

- The OCX contractor is attempting to employ cloud-based testing and a DevOps approach. The contractor said it had to gain approval from the DOD Chief Information Office to employ commercial cloud-based testing for the unclassified portions of OCX but it has not gained similar approval for the classified portion.
- The SBIRS contractor is using a software testing tool that would allow for faster automated testing but is not yet realizing the full benefit of its use. The SBIRS testers did not use this tool in the way it was intended. Specifically, the contractor said that when the software was deployed to the testing environment, testers deactivated the software at the end of their shifts instead of allowing it to run continuously until the tests were complete. The contractor said the testers did this because there were concerns over unauthorized access to the system if no one was present. As a result, the contractor separated the tests into 8-hour segments rather than allowing the tests to run continuously, reducing the effectiveness and value of automated testing.

The Defense Science Board, Defense Innovation Board, and others have recommended DOD use tools that enable the developers, users, and management to work together daily. As noted, DOD is required to begin implementation of the recommendations made in the Defense Science Board report.

What are Software Metrics?

Software metrics are measurements which provide insight to the status and quality of software development.

Source: GAO analysis of Industry Documentation | GAO-19-136

Metrics may not support newer development approaches. We have previously found that leading developers track software-specific metrics to gauge a program's progress, and that traditional cost and schedule metrics alone may not provide suitable awareness for managing iterative software development performance.[64] Three programs have faced

[64] GAO, *Immigration Benefits System: U.S. Citizenship and Immigration Services Can Improve Program Management*, GAO-16-467 (Washington, D.C.: July 7, 2016); *Software Development: Effective Practices and Federal Challenges in Applying Agile Methods,* GAO-12-681 (Washington, D.C.: July 27, 2012); and Defense Acquisitions: *Stronger Management Practices Are Needed to Improve DOD's Software-Intensive Weapon Acquisitions,* GAO-04-393 (Washington, D.C: Mar. 1, 2004).

challenges in identifying and collecting metrics that provide meaningful insight into software development progress:

- JMS planned to collect traditional software development metrics to measure software size and quality, as well as Agile metrics that provide insight into development speed and efficiency. However, officials from the JMS government integrator managing sub-contracts said they lack regular reporting of metrics and access to data from subcontractors that would allow them to identify defects early. These officials said this was a challenge because the program has to run its own quality scans at the end of each sprint instead of being able to identify defects on a daily basis.
- MUOS program officials were able to receive Agile metrics from the contractor when they transitioned to Agile development, but they lacked access to the source data, which they said hindered their ability to oversee development.
- OCX program officials said they plan to use performance-based metrics throughout the remainder of the program. However, the metrics may not adequately track performance as intended. The Defense Contract Management Agency reviewed OCX metrics, particularly those related to DevOps, and expressed concern that program metrics may only measure total defects that were identified and corrected but may not provide insight into the complexity of those defects.

DOD is taking steps to identify useful software development metrics and ways to include them in new contracts. DOD is aware of challenges with metrics and is taking actions to address the issues. For example, the Defense Innovation Board is consulting with commercial companies to determine what metrics DOD should collect; and the Air Force’s Space and Missile Systems Center has tasked The Aerospace Corporation with examining how to apply software performance metrics in contracts for DOD space programs. DOD offices such as the Defense Science Board and DOD Systems Engineering, as well as several Federally Funded Research and Development Centers including the Software Engineering Institute and The Aerospace Corporation, have also attempted to identify new metrics in correlation with advances in software development approaches.

Two Program Offices Lacked Newer Software Development Knowledge, but DOD Is Working to Improve Training

Two program offices we reviewed experienced challenges due to limited software development knowledge:

- OCX experienced an extended period of inefficient processes because it lacked an understanding of newer approaches. According to Defense Digital Service, when the Office of Secretary of Defense advised the OCX program in May 2016, it discovered that neither the program office nor contractor had been aware of the benefits of automated testing. Defense Digital Service helped the OCX contractor automate a process that had been taking as long as 18 months to one in which the same process takes less than a day. If the program office had been aware of newer software approaches, it could have recognized these inefficiencies much earlier and avoided unnecessary schedule delays.
- The MUOS contractor lacked an "Agile advocate" in the program office, which undermined its ability to fully employ an Agile development approach.[65] For example, even after the contractor adopted an Agile approach, the program office directed the contractor to plan out all work across software builds in order to maintain control over requirements—similar to a waterfall approach but inefficient in Agile. According to the Software Engineering Institute, without an Agile advocate in a program's leadership, organizations tend to do a partial Agile or "Agile-like" approach.[66]

Program officials from the programs we reviewed said that while they have taken some software development training, more would be beneficial. The JMS program office said that there are external training courses available locally as well as trainings at Air Force's Space and Missile Systems Center, but neither are required. JMS program officials said that, while specific software training has not been required for the program outside of Defense Acquisition University certifications, courses on managing software-intensive programs would have been beneficial. Similarly, Defense Contract Management Agency officials told us that

[65] A practice of Agile development is to identify an Agile champion within senior management—someone with formal authority within the organization to advocate the Agile approach and resolve impediments. Similarly, another practice of Agile development is to ensure all teams include coaches or staff with Agile experience. This practice stresses the importance of including those with direct experience in applying Agile on each team.

[66] Software Engineering Institute, *Agile Software Teams: How They Engage with Systems Engineering on DoD Acquisition Programs,* Technical Note CMU/SEI-2014-TN-013 (July 2014).

OCX program officials would have benefited from more software development training. The MUOS program office said its training on software acquisition, software and systems measurement, software planning supportability and cost estimating, and software policies and best practices was sufficient, but the program office did not have newer software development training prior to transitioning to an Agile development approach.

DOD is working to improve software acquisition training requirements and update them to reflect changes in the software development industry. For example, in 2017, the Defense Acquisition University introduced a course on Agile software development that includes how Agile fits into the overall Defense Acquisition System and how to manage an Agile software development contract. DOD told us it is also working with the Defense Acquisition University to help inform a course on DevOps automation.

Conclusions

Software is an increasingly important enabler of DOD space systems. However, DOD has struggled to deliver software-intensive space programs that meet operational requirements within expected time frames. Although user involvement is critical to the success of any software development effort, key programs often did not effectively engage users. Program efforts to involve users and incorporate feedback frequently did not match plans. This was due, in part, to the lack of specific guidance on the timing, frequency, and documentation for user involvement and feedback. The lack of user engagement has contributed to systems that were later found to be operationally unsuitable.

Selected programs have also faced challenges in delivering software in shorter time frames, and in using commercial software, applying outdated software tools and metrics, and having limited knowledge and training in newer software development techniques. DOD acknowledges these challenges and is taking steps to address them.

Recommendations for Executive Action

We are making the following two recommendations to DOD:

The Secretary of Defense should ensure the department’s guidance that addresses software development provides specific, required direction on when and how often to involve users so that such involvement is early and continues through the development of the software and related program components. (Recommendation 1)

The Secretary of Defense should ensure the department's guidance that addresses software development provides specific, required direction on documenting and communicating user feedback to stakeholders during software system development. (Recommendation 2)

Agency Comments and Our Evaluation

We provided a draft of this product to the Department of Defense for comment. In its comments, reproduced in appendix II, DOD concurred. DOD also provided technical comments, which we incorporated as appropriate.

We are sending copies of the report to the Acting Secretary of Defense; the Secretaries of the Army, Navy, and Air Force; and interested congressional committees. In addition, the report will be available at no charge on GAO's website at http://www.gao.gov.

If you or your staff have any questions about this report, please contact me at (202) 512-4841 or ludwigsonj@gao.gov. Contact points for our Offices of Congressional Relations and Public Affairs may be found on the last page of this report. GAO staff who made major contributions to this report are listed in appendix III.

Jon Ludwigson

Jon Ludwigson
Acting Director, Contracting and National Security Acquisitions

List of Committees

The Honorable James M. Inhofe
Chairman
The Honorable Jack Reed
Ranking Member
Committee on Armed Services
United States Senate

The Honorable Richard Shelby
Chairman
The Honorable Richard Durbin
Ranking Member
Subcommittee of Defense
Committee on Appropriations
United States Senate

The Honorable Adam Smith
Chairman
The Honorable Mac Thornberry
Ranking Member
Committee on Armed Services
House of Representatives

The Honorable Peter Visclosky
Chairman
The Honorable Ken Calvert
Ranking Member
Subcommittee of Defense
Committee on Appropriations
House of Representatives

Appendix I: Objectives, Scope, and Methodology

Senate and House reports accompanying the National Defense Authorization Act for Fiscal Year 2017 contained provisions for GAO to review challenges in software-intensive Department of Defense (DOD) space systems, among other things. This report addresses, for selected software-intensive space programs, (1) the extent to which these programs have involved users and delivered software using newer development approaches; and (2) what software-specific management challenges, if any, these programs have faced.

To select the programs, we identified a non-generalizable, purposeful sample of four major defense programs representing different space military services where software is an essential component and where each program has experienced cost growth or schedule delays attributed, in part, to software challenges. We began our selection process with 49 DOD space programs from the U.S. Air Force and Navy services as identified by the Office of the Assistant Secretary of the Air Force for Space Acquisition and a GAO subject matter expert. We then narrowed our selection to 19 Major Defense Acquisition Programs (MDAP) and Major Acquisition Information System (MAIS) programs identified by DOD. Next, using information from prior GAO Annual Weapons Assessments, DOD Selected Acquisition Reports, DOD Defense Acquisition Executive Summary Reports, and the Defense Acquisition Management Information Retrieval system, we identified 15 programs that were software-intensive systems as defined in the international standard ISO/IEC/IEEE 42207. This standard states that a software-intensive system is one where software contributes essential influences to the design, construction, deployment, and evolution of the system as a whole. From these 15 programs, 8 were found to have had cost growth or schedule delays attributed, in some part, to software development.[1] We further analyzed these 8 programs for unit cost or schedule breaches as defined in 10 U.S.C. § 2433 and 10 U.S.C. § 2366b, ultimately resulting in 7 programs.[2] Finally, from these 7 programs, we chose a purposeful sample of 5 programs, ensuring representation from different DOD services and Acquisition Categories.

These programs are:

- Family of Advanced Beyond Line-of-Sight Terminals (FAB-T); Air Force MDAP
- Next Generation Operational Control System (OCX); Air Force MDAP
- Joint Space Operations Center Mission System Increment 2 (JMS); Air Force MAIS
- Mobile User Objective System (MUOS); Navy MDAP
- Space-Based Infrared System (SBIRS); Air Force MDAP

We were unable to assess FAB-T software issues with the same level of detail as the other programs we reviewed because, despite prior software

[1] These cost and schedule breaches are, or in the case of schedule breaches, were, defined as: Significant Nunn-McCurdy Unit Cost Breach: The cost growth threshold, as it relates to the current Acquisition Performance Baseline, is an increase of at least 15 percent over the program acquisition unit cost (PAUC) or average procurement unit cost (APUC) for the current program as shown in the current Baseline Estimate. The cost growth threshold, as it relates to the original Acquisition Performance Baseline, is an increase of at least 30 percent over the PAUC or APUC for the original program as shown in the original Baseline Estimate. Significant Change, as it relates to the original estimate: schedule change that will cause a delay of more than 6 months but less than 1 year; an increase in the estimated development cost or full life-cycle cost for the program by at least 15 percent but less than 25 percent; or a significant, adverse change in the expected performance of the MAIS to be acquired. Critical Change, as it relates to the original estimate: the system has failed to achieve a Full Deployment Decision within 5 years after the Milestone A decision for the program or, if there was no Milestone A, the date when the preferred alternative is selected for the program (excluding any time during which program activity is delayed as a result of a bid protest); a schedule change will cause a delay of 1 year or more; the estimated development cost or full life-cycle cost for the program has increased 25 percent or more; or a change in expected performance will undermine the ability of the system to perform the functions anticipated (i.e., the expected failure to meet a threshold Key Performance Parameter).

[2] The MDAP definitions for significant and critical unit cost breaches are based on unit cost growth as defined in 10 U.S.C. § 2433. The MAIS program definitions for significant and critical changes are based on schedule, cost, or expected performance of the program were defined in 10 U.S.C. 2445c prior to repeal by the National Defense Authorization Act for Fiscal Year 2010, Pub. L. No. 111-84, § 846 (2009).

challenges, the program stated it does not have documentation that separately tracks software-related requirements or efforts. This brought our total to 4 selected programs.

To address the objectives, we interviewed officials from the Undersecretary of Defense for Acquisition and Sustainment, Office of the Deputy Assistant Secretary of Defense for Systems Engineering, Office of Cost Assessment and Program Evaluation, Office of the Director of Operational Test and Evaluation, Defense Digital Service, Defense Innovation Board, and the Office of the Assistant Secretary of the Air Force for Space Acquisition. We also interviewed officials from the selected program offices and their respective contractors, subcontractor, integrator, space systems users, a DOD test organization, and Federally Funded Research and Development Centers. In addition, we conducted a literature search using a number of bibliographic databases, including ProQuest, Scopus, DIALOG, and WorldCat. We reviewed documentation that focused on software-intensive major military acquisitions. We conducted our search in March 2018.

To determine how effectively selected DOD software-intensive space programs have involved users and adopted newer software development approaches, we reviewed applicable DOD policies, guidance, and federal statute that identify characteristics of user engagement. These sources were the Department of Defense Instruction (DODI) 5000.02; Office of the Secretary of Defense Report to Congress, *A New Approach for Delivering Information Technology in the Department of Defense;* and National Defense Authorization Act for Fiscal Year 2010. We supplemented this with Defense Science Board and Defense Innovation Board documentation, and other industry analyses. We then reviewed relevant program plans and documentation, such as human engineering and human systems integration plans, standard operating procedures, acquisition strategies, software development plans, and other program user engagement guidance to identify plans for user engagement. We then conducted interviews with space system users and analyzed software development documentation to evaluate the extent to which programs met these DOD user engagement characteristics. We also analyzed user feedback reports to identify trends in user feedback. We also examined DOD and OMB guidance and applicable leading practices to identify time frames for delivering software under incremental and iterative software development approaches, and we compared these time frames to program performance.

To determine what software-specific management challenges, if any, selected programs faced, we reviewed reports and studies on software tools and metrics used to manage software programs, including GAO reports,[3] DOD policies and guidance,[4] and studies from the Software Engineering Institute.[5] We then reviewed program documents, such as Software Development Plans, System Engineering Plans, System Engineering Management Plans, Software Resource Data Reports, Test and Evaluation Master Plans, Master Software Build Plans, and Obsolescence Plans, as applicable, as well as contracts and Statements of Work. We reviewed defect metrics and reports on amounts of new, reused, inherited, and commercial software; test and evaluation reports; program management reports; and external program assessments. We also evaluated program retrospectives and DOD reports on leading practices to understand how programs are making efforts to address challenges in these areas. We spoke with contractors and an applicable subcontractor and government integrator, program officials, and officials from Federally Funded Research and Development Centers to understand program issues, including program office and contractor training requirements.

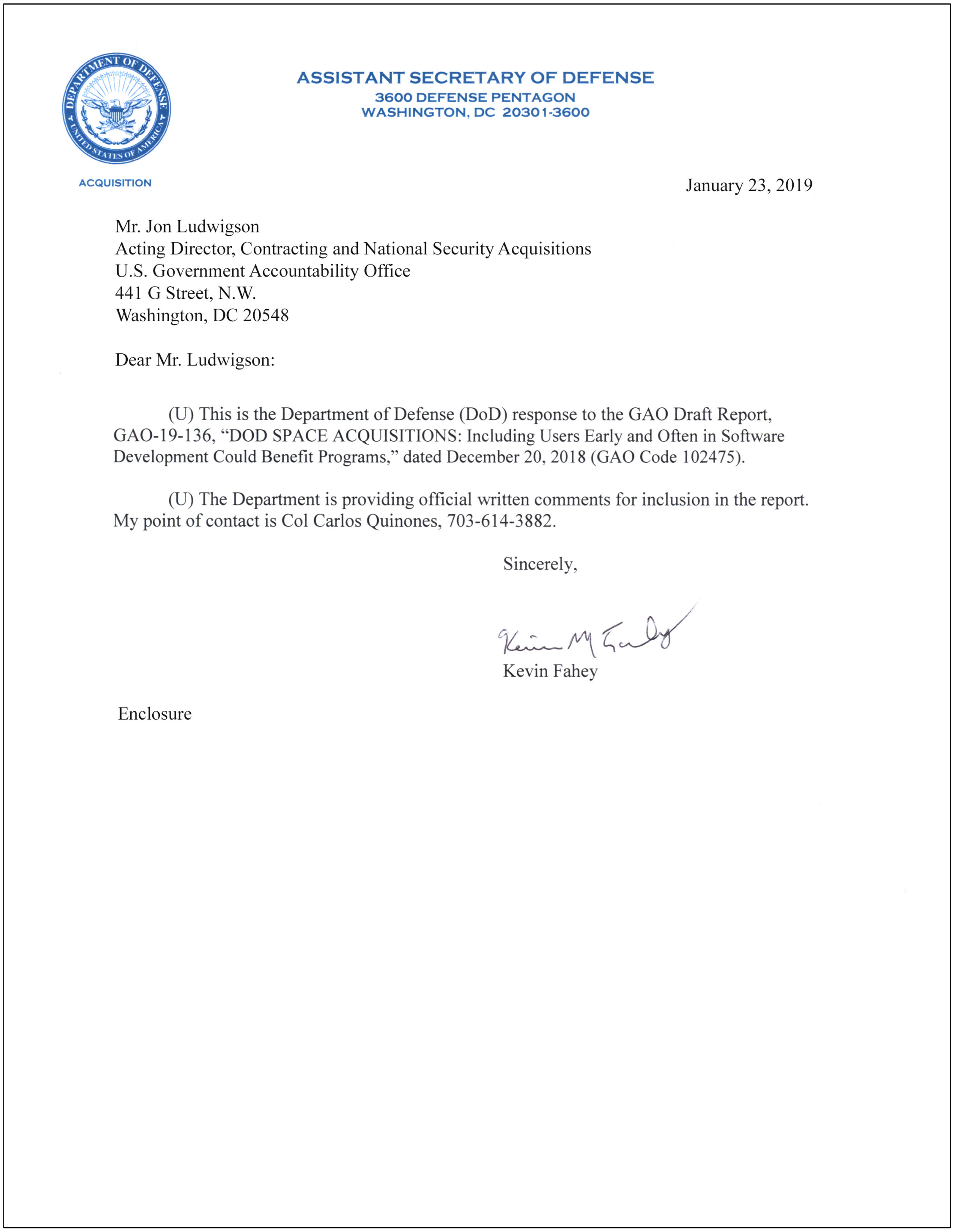
[3] GAO, Immigration Benefits System: *U.S. Immigration Services Can Improve Program Management,* GAO-16-467 (Washington, D.C.: July 7, 2016); *Standards for Internal Control in the Federal Government,* GAO-14-704G, *(*Washington, D.C.: September 2014); *Software Development: Effective Practices and Federal Challenges in Applying Agile Methods,* GAO-12-681 (Washington, D.C.: July 27, 2012); *Information Technology: Critical Factory Underlying Successful Major Acquisitions,* GAO-12-7 (Washington, D.C.: Oct. 21, 2011); and *Defense Acquisitions: Stronger Management Practices Are Needed to Improve DOD's Software Intensive Weapon Acquisitions,* GAO-04-393 (Washington, D.C.: Mar. 1, 2004).

[4] DOD Instruction 5000.02, *Operation of the Defense Acquisition System*, (Aug. 31, 2018); DOD 5000.4-M-2, *Software Resources Data Report (SRDR) Manual,* (February 2004); ISO/IEC/IEEE 12207, *International Standard: Systems and Software Engineering—Software Life Cycle Processes*, ISO/IEC/IEEE 12207:2017(E), (November 2017); Office of the Assistant Secretary of the Navy (Research, Development and Acquisition), *Guidebook for Acquisition of Naval Software Intensive Systems*, (September 2008); Office of the Assistant Secretary of the Navy (Research, Development and Acquisition), *Software Criteria and Guidance for Systems Engineering Technical Reviews (SETR),* (September 2010); Air Force Space Command Space and Missile Systems Center Standard, Software *Development*, SMC-S-012, (Jan. 16, 2015).

[5] Software Engineering Institute, *DOD Software Factbook: Software Engineering Measurement and Analysis Group, Version 1.1*, (December 2015); Software Engineering Institute, *Isolating Patterns of Failure in Department of Defense Acquisition,* Technical Note CMU/SEI-2013-TN-014, (June 2013); Software Engineering Institute*, Using Software Development Tools and Practices in Acquisition,* Technical Note CMU/SEI-2013-TN-017, (December 2013); Software Engineering Institute, *Agile Metrics: Progress Monitoring of Agile Contractors,* Technical Note CMU/SEI-2013-TN-029; (January 2014).

We conducted this performance audit from November 2017 to March 2019 in accordance with generally accepted government auditing standards. Those standards require that we plan and perform the audit to obtain sufficient, appropriate evidence to provide a reasonable basis for our findings and conclusions based on our audit objectives. We believe that the evidence obtained provides a reasonable basis for our findings and conclusions based on our audit objectives.

Appendix II: Comments from the Department of Defense



ASSISTANT SECRETARY OF DEFENSE
3600 DEFENSE PENTAGON
WASHINGTON, DC 20301-3600

ACQUISITION

January 23, 2019

Mr. Jon Ludwigson
Acting Director, Contracting and National Security Acquisitions
U.S. Government Accountability Office
441 G Street, N.W.
Washington, DC 20548

Dear Mr. Ludwigson:

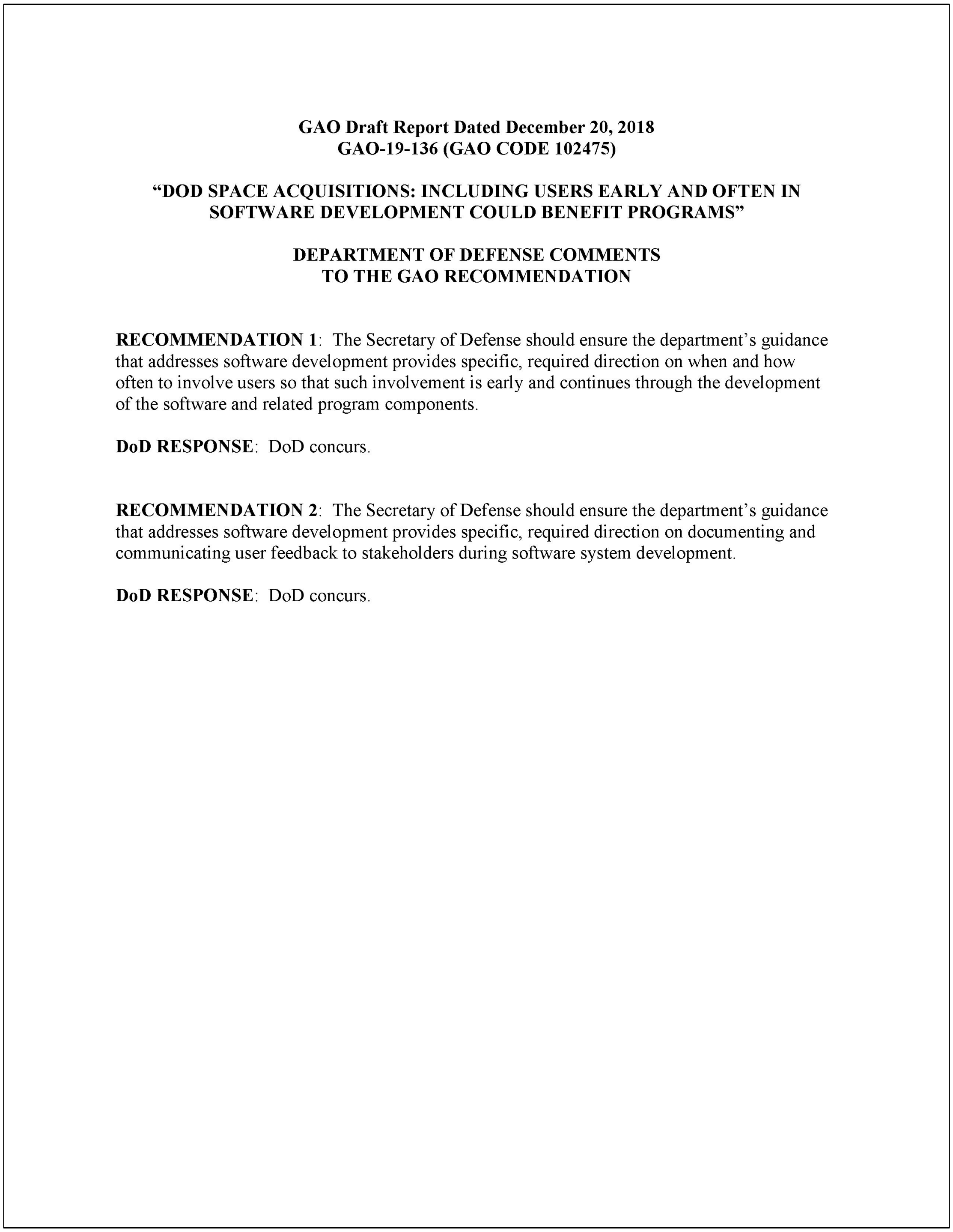
(U) This is the Department of Defense (DoD) response to the GAO Draft Report, GAO-19-136, "DOD SPACE ACQUISITIONS: Including Users Early and Often in Software Development Could Benefit Programs," dated December 20, 2018 (GAO Code 102475).

(U) The Department is providing official written comments for inclusion in the report. My point of contact is Col Carlos Quinones, 703-614-3882.

Sincerely,

Kevin M Fahey
Kevin Fahey

Enclosure



GAO Draft Report Dated December 20, 2018
GAO-19-136 (GAO CODE 102475)

"DOD SPACE ACQUISITIONS: INCLUDING USERS EARLY AND OFTEN IN SOFTWARE DEVELOPMENT COULD BENEFIT PROGRAMS"

DEPARTMENT OF DEFENSE COMMENTS TO THE GAO RECOMMENDATION

RECOMMENDATION 1: The Secretary of Defense should ensure the department's guidance that addresses software development provides specific, required direction on when and how often to involve users so that such involvement is early and continues through the development of the software and related program components.

DoD RESPONSE: DoD concurs.

RECOMMENDATION 2: The Secretary of Defense should ensure the department's guidance that addresses software development provides specific, required direction on documenting and communicating user feedback to stakeholders during software system development.

DoD RESPONSE: DoD concurs.

Appendix III GAO Contacts and Staff Acknowledgments

GAO Contact

Jon Ludwigson, (202) 512-4841 or ludwigsonj@gao.gov

Staff Acknowledgments

In addition to the contact named above, Raj Chitikila, Assistant Director; Pete Anderson, Erin Carson, Jordan Kudrna, Matthew Metz, Roxanna Sun, and Jay Tallon made key contributions to this report. Assistance was also provided by Mathew Bader, Virginia Chanley, Susan Ditto, Sarah Gilliland, Carol Harris, Harold Podell, Andrea Starosciak, Anne Louise Taylor, and Alyssa Weir.

Related GAO Products

Weapon Systems Cybersecurity: DOD Just Beginning to Grapple with Scale of Vulnerabilities. GAO-19-128. Washington, D.C.: October 9, 2018.

Weapon Systems Annual Assessment: Knowledge Gaps Pose Risks to Sustaining Recent Positive Trends. GAO-18-360SP. Washington, D.C.: April 25, 2018.

Information Technology: Agencies Need to Involve Chief Information Officers in Reviewing Billions of Dollars in Acquisitions. GAO-18-42. Washington, D.C.: January 10, 2018.

Global Positioning System: Better Planning and Coordination Needed to Improve Prospects for Fielding Modernized Capability. GAO-18-74. Washington, D.C.: December 12, 2017.

Information Technology Reform: Agencies Need to Improve Certification of Incremental Development. GAO-18-148. Washington, D.C.: November 7, 2017

Space Acquisitions: DOD Continues to Face Challenges of Delayed Delivery of Critical Space Capabilities and Fragmented Leadership. GAO-17-619T. Washington, D.C.: May 17, 2017.

Defense Acquisitions: Assessment of Selected Weapon Programs. GAO-17-333SP. Washington, D.C.: March 30, 2017.

Immigration Benefits System: U.S. Immigration Services Can Improve Program Management. GAO-16-467. Washington, D.C.: July 7, 2016.

GPS: Actions Needed to Address Ground System Development Problems and User Equipment Production Readiness. GAO-15-657. Washington, D.C.: September 9, 2015.

Defense Acquisitions: Assessments of Selected Weapon Programs. GAO-15-342SP. Washington, D.C.: March 12, 2015.

Defense Major Automated Information Systems: Cost and Schedule Commitments Need to Be Established Earlier. GAO-15-282. Washington, D.C.: February 26, 2015.

Standards for Internal Control in the Federal Government. GAO-14-704G. Washington, D.C.: September 2014.

Software Development: Effective Practices and Federal Challenges in Applying Agile Methods. GAO-12-681. Washington, D.C.: July 27, 2012.

Information Technology: Critical Factors Underlying Successful Major Acquisitions. GAO-12-7. Washington, D.C.: October 21, 2011.

Space Acquisitions: Development and Oversight Challenges in Delivering Improved Space Situational Awareness Capabilities. GAO-11-545. Washington, D.C.: May 27, 2011.

Significant Challenges Ahead in Developing and Demonstrating Future Combat System's Network and Software. GAO-08-409. Washington, D.C.: March 7, 2008.

Space Based Infrared System High Program and its Alternative. GAO-07-1088R. Washington, D.C.: September 12, 2007.

Defense Acquisitions: Stronger Management Practices Are Needed to Improve DOD's Software-Intensive Weapon Acquisitions. GAO-04-393. Washington, D.C.: March 1, 2004.

Information Security: Effective Patch Management is Critical to Mitigating Software Vulnerabilities. GAO-03-1138T. Washington, D.C.: September 10, 2003.

Test and Evaluation: DOD Has Been Slow in Improving Testing of Software-Intensive Systems. GAO/NSIAD-93-198. Washington, D.C.: September 29, 1993.

Mission-Critical Systems: Defense Attempting to Address Major Software Challenges. GAO/NSAID-93-13. Washington, D.C.: December 24, 1992.

Space Defense: Management and Technical Problems Delay Operations Center Acquisition. GAO/IMTEC-89-18. Washington, D.C.: April 20, 1989.

(102475)

GAO's Mission

The Government Accountability Office, the audit, evaluation, and investigative arm of Congress, exists to support Congress in meeting its constitutional responsibilities and to help improve the performance and accountability of the federal government for the American people. GAO examines the use of public funds; evaluates federal programs and policies; and provides analyses, recommendations, and other assistance to help Congress make informed oversight, policy, and funding decisions. GAO's commitment to good government is reflected in its core values of accountability, integrity, and reliability.

Obtaining Copies of GAO Reports and Testimony

The fastest and easiest way to obtain copies of GAO documents at no cost is through GAO's website (https://www.gao.gov). Each weekday afternoon, GAO posts on its website newly released reports, testimony, and correspondence. To have GAO e-mail you a list of newly posted products, go to https://www.gao.gov and select "E-mail Updates."

Order by Phone

The price of each GAO publication reflects GAO's actual cost of production and distribution and depends on the number of pages in the publication and whether the publication is printed in color or black and white. Pricing and ordering information is posted on GAO's website, https://www.gao.gov/ordering.htm.

Place orders by calling (202) 512-6000, toll free (866) 801-7077, or
TDD (202) 512-2537.

Orders may be paid for using American Express, Discover Card, MasterCard, Visa, check, or money order. Call for additional information.

Connect with GAO

Connect with GAO on Facebook, Flickr, Twitter, and YouTube.
Subscribe to our RSS Feeds or E-mail Updates. Listen to our Podcasts.
Visit GAO on the web at https://www.gao.gov.

To Report Fraud, Waste, and Abuse in Federal Programs

Contact:

Website: https://www.gao.gov/fraudnet/fraudnet.htm

Automated answering system: (800) 424-5454 or (202) 512-7700

Congressional Relations

Orice Williams Brown, Managing Director, WilliamsO@gao.gov, (202) 512-4400,
U.S. Government Accountability Office, 441 G Street NW, Room 7125,
Washington, DC 20548

Public Affairs

Chuck Young, Managing Director, youngc1@gao.gov, (202) 512-4800
U.S. Government Accountability Office, 441 G Street NW, Room 7149
Washington, DC 20548

Strategic Planning and External Liaison

James-Christian Blockwood, Managing Director, spel@gao.gov, (202) 512-4707
U.S. Government Accountability Office, 441 G Street NW, Room 7814,
Washington, DC 20548