DEPARTMENT OF THE AIR FORCE
WASHINGTON, DC

OFFICE OF THE ASSISTANT SECRETARY

MEMORANDUM FOR THE ACQUISITION ENTERPRISE

SUBJECT: Digital Building Code for the Transformation of Acquisition and Sustainment

In line with the National Defense Strategy, the Department of Defense's Digital Engineering Strategy, and Chief of Staff of the Air Force's Accelerate Change or Lose Paper, the Department of the Air Force (DAF) is implementing a Digital Transformation across the acquisition and sustainment enterprise. This is truly a cultural, business, and technological transformation. As such, it requires the continued support and engagement of the entire community: across all functional domains; from program trenches to headquarters staffs; from freshly hired graduates to seasoned veterans; and with government and industry collaboration. A Digital Transformation is the disruptive enabler we need to overcome our adversaries' rapidly increasing parity. Through this Digital Transformation, and the relentless spirit of Airmen and Guardians across the Department, we can and will transform our acquisition enterprise into one that securely delivers capability at the speed of relevance.

In response, the DAF laid out a strategic vision to improve its acquisition and sustainment practices through a Digital Transformation that includes Digital Engineering and Management, Agile Software Development, and Open System Architectures. This strategic vision promotes digitally enabled processes and replaces the linear, document-centric approach of today with a dynamic, model-centric approach. This new approach places emphasis on evolving and refining models as opposed to updating paper documents; obtaining appropriate intellectual property (IP) rights to prevent vendor-lock during sustainment; and developing and delivering capabilities in rapid, innovative, and agile ways. Implementing the Department's Digital Transformation involves artful execution of this "Digital Building Code." The Digital Building Code is intended to be a living set of best practices that will be updated and to capture lessons learned along the DAF's Digital Transformation journey.

Digital Acquisition and Sustainment hold the key to unleashing the speed and agility we need to field capability at the tempo required to win in future conflicts with peer competitors. The attached tabs give a point of departure for executing programs aligned with the Digital Transformation concept of operations.

Andrew P. Hunter
Assistant Secretary of the Air Force
(Acquisition, Technology & Logistics)

Frank Calvelli
Assistant Secretary of the Air Force
(Space Acquisition & Integration)

Attachments:
1. Digital Building Code for Digital Engineering and Management
2. Digital Building Code for Agile Software
3. Digital Building Code for Open Systems Architecture

ATTACHMENT 1

DIGITAL BUILDING CODE FOR DIGITAL ENGINEERING AND MANAGEMENT.

The key to employing Digital Engineering and Management is achieving **a measure of *authoritative* virtualization that automates, replaces or truncates real-world activities**. This is how you realize game-changing agility that Digital Acquisition and Sustainment can deliver for your program and our warfighters. In addition, it is also how you will realize the return on investment (ROI) for your digital transformation efforts.

The following guidance is provided to assist PEOs/PMs to determine and implement Digital Engineering:

1. **Develop digital models of systems**

    1.1. Build and maintain appropriate fidelity model-based representations of systems in commercial-off-the-shelf (COTS) architecture tools using Systems Modeling Language (SysML), or equivalent modeling language. This enables the effective exchange of information including requirements, system functions, and process flows between all organizations involved in the development process and through sustainment.

    1.2. Determine guidelines/use cases that will scope what needs to be modeled and to what fidelity given acquisition and sustainment strategies. Reference an established style guide to build and maintain the models. Consult the Air Force Digital Guide (https://usaf.dps.mil/teams/afmcde) for the latest guidance on the most suitable style guide.

    1.3. Encapsulate all necessary elements in models with appropriate tags to facilitate tracing between requirements, technical data, and certifications. Clearly link requirements to planned verification activities (e.g., technical reviews, certifications, testing plans, and procedures).

    1.4. Construct models to enable the following to be traced from the requirements: analysis of requirements, system architecture design, allocations, interfaces, certifications, and functional thread analysis.

    1.5. Include the capability to predict operational performance and quantify uncertainty in models of a system or subsystem in a simulated, representative environment.

2. **Develop a digital twin and digital thread**

    2.1. Establish and manage a digital thread that links models and digital artifacts and creates an authoritative source of truth. A program or a platform may be an integrator of multiple digital twins comprising the system. Update digital artifacts throughout the system lifecycle to maintain a digital twin of the physical system.

    2.2. Construct digital threads using a data architecture that defines the data, schemas, integration, transformations, storage, and workflows required to design and sustain

the system. The data architecture should also define naming conventions, data types/formats, integrity, archival/retention, required security, flows, pipelines, linkage with associated metadata, and transformations.

3. Implement an integrated digital environment

3.1. Use an integrated digital environment (IDE). An IDE is a compilation of data, models, and tools for collaboration, analysis, and visualization across functional domains. An IDE includes the methodology and specification for data, models, and tools arrangement with processes and procedures to exploit informational results.

3.2. An ideal IDE would include the following:

3.2.1. Development Platform: CloudONE, PlatformONE, and DataONE.

3.2.2. Architectural Modeling: COTS software such as CAMEO Systems Modeler, Sparx Enterprise Architect, or IBM Rhapsody.

3.2.3. Product Lifecycle Management (PLM): Siemens Teamcenter or approved alternative. For additional information: AF-PLM-CSO@us.af.mil.

3.2.4. System Performance Modeling and Operational Analysis: AFSIM or other M&S environments.

3.2.5. Requirements Management: COTS software such as DOORS or CAMEO Systems Modeler.

3.2.6. Data analyses and visualization: Tableau, Power BI, MATLAB, Python or other visualization tools

3.3. Determine and implement an IDE strategy that specifies preferred digital tools, considers tools accessibility and security considerations, and outlines the impact this strategy will have on internal and industry collaborations (e.g., tool integration, data interoperability). A cost benefit analysis should assess whether to acquire and deploy tools, use DoD High Performance Computing Modernization Program (HPCMP) resources, or negotiate the use of tools by industry performers. Tips for initial implementation of an IDE can be found at the Air Force Digital Guide: https://usaf.dps.mil/teams/afmcde.

4. Employ a tailored digital strategy for contracting with industry

4.1. Because digital transformation is in its early stages, contracting guidance is rapidly evolving. For the latest recommendations and templates, please see the Air Force Digital Guide contracting section: https://usaf.dps.mil/teams/afmcde/SitePages/Model-based-Contract-Language.aspx. Here practitioners can access information on “Key Digital Features” that should be considered during contracting actions. In addition, programs can access example contract language from other acquisitions. Additional contracting Tactics, Techniques, and Procedures (TTPs) on this and related topics will be added over time and available online through Air Force Contracting Central. The Digital Guide also includes guidance on IP and technical data for contract language.

5. Ensure organizational readiness for Digital Engineering

5.1. To ensure a central point of contact for tools and infrastructure needs, enable consistent implementation and coordination, and ensure sharing of lessons learned and collaboration, programs may designate the Chief Engineer, or an alternate, as the DE focal point within their organization. The DE focal point is responsible for specifying general digital engineering training, courses, and certifications for the program to ensuring an organizational minimum working knowledge of digital engineering, regardless of function. Examples of available workforce training for digital tools and related infrastructure, include:

5.1.1. MBSE modeling language: SysML based tools (or their equivalent). Also, consider other related training, such as UPDM, UAFP, UML, BPMN, and XML.

5.1.2. Cloud Environment: CloudONE and PlatformONE services. Training resources are available at https://software.af.mil/training.

5.1.3. DevSecOps processes: Training resources are available at https://software.af.mil/training.

5.1.4. Digital Engineering / Advanced Engineering Analysis: Applicable modeling techniques for applications such as structures, design/analysis (e.g., CAD, FEA, CFD), embedded software, electronics, and other disciplines as appropriate.

5.2. Select training and organizational readiness information can also be found in the Air Force Digital Guide at https://usaf.dps.mil/teams/afmcde.

6. Implement Digital Acquisition

6.1. Digital Engineering will fundamentally transform how we conduct systems engineering and acquisition processes. For example, all acquisition plans, program and technical reviews, and testing and certification processes will shift from a fundamentally document-based construct to one based on models and digital artifacts. Key steps toward this transformation for programs include:

6.1.1. Link model-based engineering activities and digital artifacts to acquisition planning in support of the Capability Development Document (CDD), Acquisition Strategy, Systems Engineering Plan (SEP), Life Cycle Sustainment Plan (LCSP), Test and Evaluation Master Plan (TEMP), and other acquisition artifacts. As program modelling implementation matures, programs should seek automated and model- based updates to these artifacts to eliminate stagnant acquisition information. Eventually, programs should strive to replace document-based acquisition information with sufficiently mature and authoritative models where appropriate. Any program not pursuing digital engineering principles should document their rationale in the acquisition strategy for Milestone Decision Authority (MDA) approval or redirection.

6.1.2. Leverage models to support acquisition reviews, including Milestone Reviews, In-Progress Reviews, and other acquisition reviews and program oversight activities.

6.1.3. Conduct Systems Engineering Technical Reviews (SETRs) (to include System Requirements Reviews, System Functional Reviews, Preliminary Design Reviews, Critical Design Reviews, and configuration audits) using models and digital artifacts in lieu of document-based artifacts to the maximum extent practicable. At technical reviews, when possible, programs should use information from the digital authoritative source of truth to assess risks, issues, opportunities, and mitigation plans in order to understand cost, schedule, and performance implications.

6.1.4. Trace and validate requirements based on models and digital artifacts to the maximum extent practicable.

6.1.5. Programs and the Developmental Test and Operational Test communities should engage early to determine strategy and planning for employing model-based test and evaluation activities. Verification and validation of models is critical to achieving *authoritative virtualizations* of systems.

6.1.6. Leverage models and digital artifacts for certification events (e.g., airworthiness, Authority to Operate (ATO)s, Information Assurance Certificates, safety, nuclear surety). Engage with certification offices to automate as much of these certification processes as practicable.

6.1.7. Leverage models and digital artifacts for planning and tracking reliability, maintainability, availability, sustainability, and other program technical performance measures.

6.1.8. Craft requests for proposals and resulting contracts to contain enforceable language that implement digital acquisition strategies and ensures deliverables are provided in the appropriate model-based and open formats.

6.2. Additional information on many of the above considerations can be found in the DAF Digital Guide at https://usaf.dps.mil/teams/afmcde.

7. Track Digital Maturity

7.1. Track progress using the DAF Digital Maturity Assessment to baseline, prioritize, and manage execution of program or organizational digital initiatives. These metrics can be used at the start of a program's journey to inform program or organizational digital implementation strategies, by PEOs to make comparisons and pool investments across portfolios, and to track progress toward successful implementation. The assessment can be found in the Air Force Digital Guide at https://usaf.dps.mil/teams/afmcde.

ATTACHMENT 2

DIGITAL BUILDING CODE FOR AGILE SOFTWARE

Over the past two years, we have seen software development transformation take root across the Air Force and Space Force in programs ranging from the F-16 to the Ground Based Strategic Deterrent (GBSD) to the Advanced Battle Management System (ABMS) to the T-38A. This transformation was propelled by adoption of the DevSecOps approach, agile software development, and open system architectures based on containerized microservices (orchestrated by Kubernetes and secured with Zero Trust). It will continue to expand through the use of common software development tech stacks that are converging around the CloudONE/PlatformONE environment. The payoffs have been game-changing for pathfinding programs. In 2020, the U-2 program made DoD history by becoming the first platform to push a software update to a jet while in flight (made possible via Kubernetes-deployed software containers). Just weeks later it became the first platform to put an artificial intelligence (AI) "operator" in control of a mission system with the deployment of the "ARTUμ" application. These transformational leaps forward attest to just how powerful this approach can be for existing programs as well as new ones.

It is now time to take this Agile Software transformation from experimental start-up phase to a coordinated, standards-based scale-up across the Department. System and component interoperability, code reusability, security assurance and continuous authority-to-operate (cATO), and other efficiencies – not to mention the Department-wide enablement of AI and machine learning (ML) – can only be fully realized if the Department converges around common development standards, many of which are outlined below.

The following standards employ open system architectures, ensuring the Department is postured to adapt as new technologies, methods, or needs arise. (For clarification, a modular open systems approach, or MOSA, is the *process* programs should leverage to achieve an open systems architecture [OSA]). Convergence on development standards does not mean innovation stops; rather, convergence around these development standards is what will unleash functional innovation at scale, and allow software development teams to focus on rapid development and deployment of new capabilities warfighters count on.

The following guidance is provided to assist PEOs/PMs to implement Agile and DevSecOps software development:

1. **Implement DevSecOps software development methodology and reference design**

    1.1. Adopt the use of Agile DevSecOps methodology as guided by the PEO C3BM (Command, Control, Communication and Battle Management) Office for all non-commercial software development, including development work performed by our Defense Industrial Base (DIB) partners.

    1.2. Move away from Waterfall-based development to Agile. Many programs are adopting Agile for their software development but leverage waterfall-like processes for their program management. This brings all the impediments of waterfall while not fully benefiting from the return on investment of Agile. Programs should adopt end-to-end Agile principles to the maximum extent practicable.

1.3. Implement the DoD Enterprise DevSecOps Reference Design: Cloud Native Computing Foundation (CNCF) Kubernetes along with or including industry partners.

1.3.1. Requirements in this reference document are continuously updated and precisely define the needs for DoD-wide reciprocity including Kubernetes, the Sidecar Container Security Stack (SCSS), and Open Container Initiative (OCI) compliant containers. This guidance is also updated to be consistent with the Defense Information Assurance (IA)/Security Accreditation Working Group (DSAWG) DevSecOps publications released by the DSAWG DevSecOps group, DoD CIO, and USD(A&S), including but not limited to Kubernetes Security Technical Implementation Guide (STIG), Container Security Requirements Guide (SRG), Container Hardening Guide, and cATO guidance documents.

1.4. For embedded systems and systems that use a real-time (RT) operating system (RTOS), only use RT hardware, RTOS, and RT software when necessary. Leverage open architecture, Kubernetes, and non-RT hardware to the maximum extent practicable. Programs should expend their best effort to decouple RT from non-RT software, and implement and improve the PlatformONE Big Bang instance (Kubernetes, Service Mesh, and containers) in RT systems as necessary.

2. Adopt the following common enterprise services and tooling standards

2.1. Leverage PlatformONE and discontinue building new or competing enterprise-wide Continuous Integration/Continuous Delivery (CI/CD) pipelines and DevOps or DevSecOps platforms. PlatformONE is a pay-per-use model which can provide significant cost savings to DAF programs.

2.1.1. Leverage either the ABMS – PlatformOnes's Party Bus (multi-tenant) or a dedicated PlatformONE Big Bang instance (dedicated platform) without "forking" its code to ensure Repo One remains the source of truth for its code base.

2.1.2. All software factories should leverage and contribute to the PlatformONE baseline on Repo One. PlatformONE is responsible for managing and enabling the environment, CI/CD pipeline, and Service Mesh layers. The software factories can focus on delivering mission capabilities by leveraging PlatformONE.

2.1.3. Existing and new DevOps, DevSecOps, CI/CD pipelines, and other software factory types should register with the PEO C3BM Office as a DAF software factory.

2.1.4. Programs leveraging these capabilities will need to use the Cloud Native Access Point, the DAF Zero Trust capability, to access Cloud providers and potentially on-premise environments when available. This will increase security, reduce the attack surface, and facilitate remote work, including for our DIB partners.

2.2. Software intensive programs and all ACAT I programs need to work with applicable test labs, nuclear surety authorities, airworthiness authorities, and other test/certification teams to deploy PlatformONE Kubernetes environments on premise to enable hardware in the loop testing using DevSecOps automation and flexibility.

2.3. Leverage Repo One as the centralized source code repository for all code (e.g., Infrastructure-as-Code, Configuration-as-Code, container source code, Kubernetes distributions) to enable code reuse across the Department and DIB partners.

2.3.1. Check Repo One to see if existing modular capability code already exists. Use Repo One capabilities to the maximum extent practicable. If an existing Repo One capability doesn't fully address program requirements, every effort should be made to contribute missing capabilities back to Repo One.

2.3.2. Avoid the "forking" of Repo One code. Forking is taking the source code from an open-source software program and developing an entirely new program. Instead of forking, contribute back to Repo One so the Department can leverage money already spent, consistent with Office of Management and Budget Memorandum M-16-21 (Federal Source Code Policy: Achieving Efficiency, Transparency, and Innovation through Reusable and Open-Source Software) to contribute to open-source projects and open sourcing of agency code.

2.4. Use only approved sources for DoD containers. Currently, Iron Bank and Registry One are the only approved sources for DoD containers (with DoD-wide reciprocity).

2.4.1. Additionally, programs are encouraged to contribute back to the Iron Bank container library to benefit the entire enterprise through reusability of code. The Iron Bank container onboarding guide is available at: https://repo1.dsop.io/dsop/dccscr/tree/master/contributor-onboarding

3. Implement the following organization staffing, leadership, and training guidance

3.1. Programs should designate a Chief Software Engineer (CSE) focal point within the organization to ensure a central point of contact for software and enable centralized coordination, sharing of lessons learned, and collaboration across programs. This focal point will also serve as a joint liaison between the Program Office and the PEO C3BM Office.

3.2. Continuous Learning is critical to ensure our talent, whether civilian, military or contractor, can keep up with software innovation. In a partnership with the Air Force Chief Information Officer, programs should continuously leverage training content provided within the Air Force Digital University.

3.3. The following training is recommended for Chief Engineers, Senior Material Leaders, Material Leaders, Program Managers, and Software Engineers:

3.3.1. Domain Driven Design (how to cut monolithic applications into micro-services, which is critical to cutting legacy systems into containers).

3.3.2. Test-Driven Development.

3.3.3. Strangler Pattern or Side Car Pattern (how to deliver new capabilities while refactoring legacy and not the other way around).

3.3.4. Microservice Architecture.

3.3.5. Prevention of Lock-In (ensure teams understand how to not get locked-in to Cloud providers and products).

3.3.6. Kubernetes (K8s).

3.3.7. Containers.

3.3.8. Kafka or any other queuing technology to capture logging and tracking information.

3.4. Leverage training content provided within the Air Force Digital University (https://software.af.mil/training).

4. Start tracking performance metrics for software factories and Agile teams

4.1. To demonstrate return on investment and effectiveness, programs should collect DevOps Research and Assessments (DORA) metrics and other data points, to include Deployment Frequency (DF), Mean Lead Time for changes (MLT), Mean Time To Recover (MTTR), and Change Failure Rate (CFR). Collected DORA metrics should be reported to the PlatformONE DevSecOps DORA metrics team to the maximum extent practicable.

A living repository of this standards information, documentation, and learning resources can be found at https://software.af.mil/dsop/documents/. There is also an Implementation Primer at this website, which outlines initial steps for applying this guidance along with answers to common questions. The PEO C3BM Office maintains this standards repository, and is available for any questions regarding this guidance at af.cso@us.af.mil.

ATTACHMENT 3

DIGITAL BUILDING CODE FOR OPEN SYSTEMS ARCHITECTURE

In defense acquisition, Open Architecture refers to adopting consensus-based standard interfaces, acquiring components and subsystems that comply with these interfaces, and integrating these components or subsystems using appropriate interface standards. Programs leverage a modular open systems approach (MOSA) to implement an open system architecture (OSA) for their systems. When implemented properly, an OSA creates a more agile, evolvable system and can bend cost, schedule, and performance curves back in our favor by driving increased competition, innovation, and adoption of mature technology from a broad range of sources – ultimately getting more innovative capability to the warfighter faster.

It is not enough to say we need to adopt Open Systems Architectures or move more programs to a specific government-owned architecture. Effectively implementing OSA requires fundamental changes in our business and technical processes in order to provide industry with the information required, focus our own energy on the activities required to drive change, and become smarter developers and buyers. A key enabler for OSA is the adoption of an open business model, which requires increased transparency to leverage the contributions of multiple contractors to share risk, maximize asset reuse, and reduce total ownership costs. We also need to resource our Program Offices and train our acquisition and engineering professionals in OSA design so they can drive this new approach. Finally, we all must ensure acquisition leadership prioritizes OSA *as much or more* than near-term cost, schedule, and performance.

It is important to understand what OSA means in the context of Digital Acquisition, and how it must be implemented. The following guidance is provided to assist PEOs/PMs in implementing OSA:

1. Implement an Open Systems Architecture

1.1. Programs shall be designed and developed, to the maximum extent practicable, with a modular open system approach (MOSA). To employ a MOSA, programs should ensure the proper level of logical, functional, and physical decomposition is performed and key modules are identified in order to leverage consensus-based standards at all appropriate interfaces and employ a system architecture that allows severable major system components and modular systems at the appropriate level to be incrementally added, removed, or replaced throughout the lifecycle.

1.2. Programs should consider generating and maintaining SysML, or equivalent open and widely adopted modeling language, architecture models of the platform, systems, subsystems, and components. Programs should ensure their modeling tools are chosen in such a way as to capture their major system interface information per the requirements laid out in the FY21 National Defense Authorization Act Section 804.

1.3. To facilitate the movement to OSAs, contracts should include relevant provisions to ensure the appropriate technical baseline documentation is made available digitally from the beginning, along with appropriate IP rights (or, in the alternative, specially negotiated licenses that will not adversely impact another platform's ability to reuse

that data – including platforms not procured by an Air Force or Space Force acquisition organization). Taking a "Smart IP" approach is an essential aspect of employing OSA because it permits the government to use, release, and disclose technical baseline documentation to product support contractors, thereby yielding cost savings or avoidance, schedule reduction, opportunities for technical upgrades to address emerging threats, and increased interoperability—all of which accelerate program agility. Unless a program's contracts implement OSA to an appropriate level of indenture of the weapon system's architecture, and the program acquires the necessary technical baseline documentation accompanied by the appropriate IP rights, the program will fail to reap the benefits of OSA. It is also essential to design the program's system architecture in a manner that is enticing to a broad ecosystem of developers, especially non-traditional commercial developers. Defining specific and appropriate proposal content and contract data requirements lists (CDRLs) will support development, delivery, and curation of OSA technical baseline models with appropriate government license rights for review at recurring program and design reviews; if architectural models are not yet feasible for your program, do the same for OSA technical baseline documentation.

1.3.1. Ensure OSA models are linked to CDDs (or equivalent), acquisition strategies, System Engineering Plans (SEPs), and Lifecycle Sustainment Plans (LCSPs) to identify: (a) to what level of indenture of the Work Breakdown Structure (WBS) for subsystems and components the program intends to implement OSA, (b) what CDRL deliverables awardees will deliver, and (c) what IP rights those awardees will grant to those IP deliverables. Those deliverables should include: (a) technical data that describes the weapon system's system and software architecture, (b) performance specifications describing the end-state functionality of all hardware components and computer software configuration items (CSCI) (or software units) comprising that weapon system, (c) modular system interfaces that define the shared boundary between those components and CSCIs, and (d) verification/validation data that demonstrates the contractor developed and produced the weapon system consistent with OSA requirements included in the contract. As described in Attachment 1, these document-based processes should shift to model-based and automated processes as we mature this digital transformation. Programs leveraging model-based approaches should implement style guides to ensure commonality across system development efforts.

1.4. As the Air Force and Space Force continue to develop a "top-down" digital architecture and programs build the "bottom-up" architecture, a more integrated and interoperable Joint Force will emerge. On occasion, architecture-level requirements will be derived from strategic level decisions. When that occurs, the PEO C3BM (Command, Control Communications and Battle Management) Office, AF Futures, and other HQ organizations will translate the senior guidance into architecture-level technical requirements. Therefore, program managers – in coordination with the PEO C3BM Office, headquarters Service staffs, and the relevant Major or Field Command may need to adjust Acquisition Program Baselines (APBs) in accordance with strategic level-derived architecture requirements.

2. Leverage Open Standards

2.1. In order to leverage the pace, scale, investment, and capability of the commercial innovation base, programs should adopt commercial technology and leverage commercial open (non-proprietary) standards to the maximum extent practicable. For example:

2.1.1. Commercial networking is based on a set of layered technologies, where the packet-based Internet Protocol (IP) is the most widely used "convergence" technology at the Network layer, which rides on top of a Data Link layer, which itself rides on a Physical layer. This enables rapid innovation at the lower layers and provides diversity in link technologies since packets can be routed over many different link technologies (e.g., Wi-Fi, fiber, cellular). The emergence of 5G technologies is an example of this rapid innovation. Therefore, when designing a network, programs should leverage IP and commercial network standards to the maximum extent possible.

2.2. In general, and in accordance with DoD Instruction 4120.24, programs should leverage commercial and consensus-based standards whenever possible. When neither an applicable commercial nor government standard exists, programs should attempt to update existing standards and grow them to meet their needs. Only when programs have exhausted these options should they seek development of a new OSA standard in partnership with industry or the PEO C3BM Office, the DAF Standardization Executive (SAF/AQR), and SAF/SQ (in the case of Space related standards). For additional information on leveraging existing industry and government standards, see DoDI 4120.24, *Defense Standardization Program (DSP)*, and AFI 60-101, *Materiel Standardization*. For additional information on updating existing government standards or creating new ones, see DoDM 4120.24.

2.3. Where no reasonable commercial standard exists, programs should leverage Government owned and managed interface and open architecture standard to ensure interoperability and ease of system integration and modernization. Specifically:

2.3.1. Open Mission Systems (OMS) is a non-proprietary open standard for integrating avionics subsystems and software services into mission packages. OMS promotes interoperability by allowing weapon systems, services, and subsystems to interact and communicate using common data formats. This interaction can occur within or between weapon systems; between platforms in sub-surface, surface, air, or space domains; or between ground segments. The OMS standard establishes a set of interface and compliance requirements that promote affordable technology refresh, capability evolution, and reuse. OMS is a consensus-based standard developed by an industry-led consortium in use since 2012. Therefore, when designing subsystems and services, and in accordance with the SAF/AQ and AFMC/CC co-signed memorandum *Use of Open mission Systems/Universal Command and Control Interface*,

programs should adopt the OMS Open Architecture Standard in cases where a commercial standard is unavailable or a poor fit.

2.3.2. Universal Command & Control (C2) Interface (UCI) is a messaging standard supporting machine-to-machine communication for mission-level C2. UCI provides a common message definition for performing mission operations and enables C2 coordination across weapon systems and weapon system elements such as sensors, vehicles, data products, and software. Therefore, when designing a C2 system, and in accordance with the SAF/AQ and AFMC/CC co-signed memorandum *Use of Open mission Systems/Universal Command and Control Interface*, programs should adopt the UCI Open Architecture Standard in cases where a commercial standard is unavailable or a poor fit. For Space-based systems, given their unique nature, the coordination and concurrence of SAF/SQ must also be sought and obtained.

2.3.3. Universal Armament Interface (UAI) is a non-proprietary open standard that fully defines the physical, logical, and mechanical interface between smart air to ground munitions and carriage systems and platforms that employ them. Therefore, when designing or integrating a smart air-to-ground munition, and in accordance with the SAF/AQ signed memorandum *Standardized Interface for USAF Air-to-Ground Weapons: Universal Armament Interface (UAI)*, programs should adopt the UAI standard in cases where a commercial standard is unavailable or a poor fit.

2.3.4. Many other military unique Open Architecture and Open Standards exist, or are maturing, across the DAF, such as Resilient Embedded GPS/INS (R-EGI), Sensor Open Systems Architecture (SOSA), Big Iron, Common Open Architecture for Radar Programs (COARPS), and the AFLCMC/EB Weapon Government Reference Architecture. Therefore, as programs are beginning to define and build their system architectures, they should leverage these existing and emerging architecture standards to the maximum extent practicable. A more exhaustive list of current efforts, with appropriate POCs and application areas, and educational content on Government Reference Architectures can be found at: https://usaf.dps.mil/teams/afmcde/SitePages/Government-Reference-Architecture.aspx

2.4. Interfaces and documentation often evolve over time based on emerging requirements, technology, and standards. In order to ensure interfaces and documentation are updated and maintained, maintain a list of key interfaces and document the standard used at these interfaces, including the justification for their selection.

3. **Designate, empower, resource, and train System Architects in Program Offices**

3.1. Programs should identify staff responsibilities for managing open architecture implementation and may wish to designate a System Architect.

3.2. To enable platforms and systems to work together as a family of systems (not simply systems), PEOs/TEOs should consider designating a Portfolio Architect responsible for guiding the implementation of Open Architecture across the PEO/TEO's portfolio.

3.3. Continuous learning is critical to ensuring our talent, whether civilian, military, or contractor, can keep up with innovation in technology to see where horizontal and portfolio gains can be made. DAU, AFIT, and the Open Architecture Management Office (OAMO), is continuously updating and augmenting training materials and accessibility. The following training material is recommended:

3.3.1. Air Force Life Cycle Management Center, EZA-064, Introduction to Open Architectures.

3.3.2. DoD Research and Engineering, Modular Open System Approach (MOSA) Reference Frameworks in Defense Acquisition Programs (https://ac.cto.mil/wp- content/uploads/2020/06/MOSA-Ref-Frame-May2020.pdf).

3.3.3. Defense Acquisition University, CLE019, Modular Open Systems Approach.

3.3.4. Other training as defined by the DAF Chief Architect in coordination with the OAMO.

3.3.5. Additional training and information will also be coming to the Air Force Digital Guide (https://usaf.dps.mil/teams/afmcde) and other DoD sites.

4. **Track and report architecture performance metrics**

4.1. To measure progress and track approaches that are working or require modification, programs and organizations should start collecting architecture metrics that can be assessed across teams with minimal disparity. The Portfolio Architects will develop a digital toolchain to minimize the burden of tracking metrics. The following metrics can be used to track implementation and maintenance of Open Systems Architecture:

4.1.1. Does the program leverage commercial open standards, and if so, what is the standard(s) and for what interface(s)?

4.1.2. Does the program leverage a Government Reference Architecture (GRA), and if so, is it being followed?

4.1.3. Does the verification/validation data demonstrate the contractor developed and produced the weapon system consistent with OSA requirements included in the contract?

4.2. Architecture-level metrics is a growing field with opportunity for improvement. PEO/TEOs are encouraged to recommend additional metrics.

No guidance can account for every situation our acquisition workforce will face. In general, programs can use the following litmus test to know whether they are taking the right approach to Open Systems Architecture to yield its transformative benefits:

"Can one or more qualified third parties add, modify, replace, remove, or support a component or subsystem of this system; and can a separate system or platform integrate and share data with my system, based on open standards and published interfaces?"

If the answer to this question is yes, the program is on the right path.