DoD Instruction 5000.87

Operation of the Software Acquisition Pathway

Originating Component:	Office of the Under Secretary of Defense for Acquisition and Sustainment
Effective:	October 2, 2020
Releasability:	Cleared for public release. Available on the Directives Division Website at https://www.esd.whs.mil/DD/.
Incorporates and Cancels:	Under Secretary of Defense for Acquisition and Sustainment Memorandum, "Software Acquisition Pathway Interim Policy and Procedures," January 3, 2020
Approved by:	Ellen M. Lord, Under Secretary of Defense for Acquisition and Sustainment

Purpose: In accordance with the authority in DoD Directive 5135.02, this issuance establishes policy, assigns responsibilities, and prescribes procedures for the establishment of software acquisition pathways to provide for the efficient and effective acquisition, development, integration, and timely delivery of secure software in accordance with the requirements of Section 800 of Public Law 116-92.

TABLE OF CONTENTS

FIGURES

SECTION 1: GENERAL ISSUANCE INFORMATION

1.1. APPLICABILITY.

This issuance applies to OSD, the Military Departments, the Office of the Chairman of the Joint Chiefs of Staff and the Joint Staff, the Combatant Commands, the Office of Inspector General of the Department of Defense, the Defense Agencies, the DoD Field Activities, and all other organizational entities within the DoD (referred to collectively in this issuance as the "DoD Components").

1.2. POLICY.

a. The overarching management principles that govern the defense acquisition system (DAS) are described in DoD Directive 5000.01 and DoD Instruction (DoDI) 5000.02. The objective of the DAS is to implement the national defense strategy, through the development of a more lethal force based on U.S. technological innovation and a culture of performance that yields a decisive and sustained U.S. military advantage. To achieve that objective, DoD will employ an adaptive acquisition framework (AAF) comprised of multiple acquisition pathways. The AAF supports the DAS with the objective of delivering effective, resilient, supportable, and affordable solutions to the end user while enabling execution at the speed of relevance.

b. The software acquisition pathway is for the timely acquisition of custom software capabilities developed for the DoD. Software programs that meet the definition of a covered Defense Business System (DBS) should use the DBS pathway in accordance with DoDI 5000.75 but may elect to incorporate this pathway for custom developed software.

c. Programs executing the software acquisition pathway are not subject to the Joint Capabilities Integration and Development System (JCIDS), and will be handled as specifically provided for by the Vice Chairman of the Joint Chiefs of Staff, in consultation with Under Secretary of Defense for Acquisition and Sustainment (USD(A&S)) and each service acquisition executive.

d. Programs executing the software acquisition pathway will not be treated as major defense acquisition programs even if exceeding thresholds in Section 2430 of Title 10, United States Code. See Section 800 of Public Law 116-92.

e. Programs using the software acquisition pathway will demonstrate the viability and effectiveness of capabilities for operational use not later than 1 year after the date on which funds are first obligated to develop the new software capability. New capabilities will be delivered to operations at least annually to iteratively meet requirements, but more frequent updates and deliveries are encouraged where practical. For programs using the embedded software path, this annual update applies after initial operational acceptance of the system in which the software is embedded and should be aligned with the associated system's schedule. Before the operational acceptance of the system in which the software is embedded, software deliveries will be delivered to an operationally representative environment at least annually.

f. Programs will require government and contractor software teams to use modern iterative software development methodologies (e.g., agile or lean), modern tools and techniques (e.g., development, security, and operations (DevSecOps)), and human-centered design processes to iteratively deliver software to meet the users' priority needs. These modern approaches will also instrument software such that critical monitoring functions related to the health, security, and operational effectiveness of the software can be automated to the maximum extent practicable.

g. Software development will be done in active collaboration with end users, representing key user groups, to ensure software deliveries address their priority needs, maximize mission impact, and undergo regular assessment of software performance and risk.

h. Leveraging existing enterprise services, if available, is preferred over creating unique software services for individual programs. These may be procured from the DoD, the DoD components, other government agencies, or commercial providers, and leverage category management solutions and enterprise software agreements.

i. Cybersecurity and program protection will be addressed from program inception throughout the program's lifecycle in accordance with applicable cybersecurity policies and issuances. A risk-based management approach will be an integral part of the program's strategies, processes, designs, infrastructure, development, test, integration, delivery, and operations. Software assurance, cyber security, test and evaluation are integral parts of this approach to continually assess and measure cybersecurity preparedness and responsiveness, identify and address risks and execute mitigation actions.

j. Intellectual property (IP) will be addressed from program inception throughout the program's lifecycle in accordance with DoDI 5010.44 and other applicable DoDIs. IP considerations will be integrated with, and support, all other program strategies to ensure return on government investment and enhance competitive options for development, integration, test, deployment, modernization, modular open systems approaches, and product support of software-intensive systems.

k. Software development testing, government developmental testing, system safety assessment, security certification, and operational test and evaluation will be integrated, streamlined, and automated to the maximum extent practicable to accelerate delivery timelines based on early and iterative risk assessments. Maximum sharing, reciprocity, availability, and reuse of results and artifacts between the various testing and certification organizations is encouraged.

l. Programs using the software acquisition pathway will report a set of data to the Office of the USD(A&S) on a semi-annual basis as defined in the AAF Software Acquisition Pathway Guidance located at https://aaf.dau.edu/aaf/software/. Data reported under this pathway will be used to monitor the effectiveness of the pathway and will not be used for program oversight.

SECTION 2: RESPONSIBILITIES

2.1. USD(A&S).

The USD(A&S):

a. Serves as the decision authority (DA) for special interest programs in the software acquisition pathway on a by-exception basis when the size, cost, complexity, performance, joint, or congressional interest warrants additional oversight.

b. Directs acquisition programs to use another acquisition pathway if the software acquisition pathway is not deemed appropriate.

2.2. UNDER SECRETARY OF DEFENSE FOR RESEARCH AND ENGINEERING.

The Under Secretary of Defense for Research and Engineering:

a. Consults with the USD(A&S) as appropriate on policies and guidance for the software acquisition pathway.

b. Guides the development of science and technology activities related to next generation software and software reliant systems for the DoD.

c. Advises the USD(A&S) on software assurance, program protection, developmental testing and evaluation, program risks, and other software areas as appropriate.

d. Advises DoD Components on program planning that anticipates the evolution of software capabilities to meet the changing threats, technology insertion, and interoperability including addressing technology gaps for DoD programs.

2.3. UNDER SECRETARY OF DEFENSE (COMPTROLLER)/CHIEF FINANCIAL OFFICER, DEPARTMENT OF DEFENSE.

The Under Secretary of Defense (Comptroller)/Chief Financial Officer, Department of Defense:

a. Consults with the USD(A&S) as appropriate on policies and guidance for the software acquisition pathway.

b. Reviews and advises the DoD on funding for programs using the software acquisition pathway through the DoD's tailored planning, programming, budgeting, and execution processes.

2.4. DOD CHIEF INFORMATION OFFICER.

The DoD Chief Information Officer:

a. Consults with the USD(A&S) as appropriate on policies and guidance for the software acquisition pathway.

b. Maintains a current knowledge resource listing DoD and DoD Component Enterprise Capabilities available to programs employing the software pathway.

c. Coordinates with USD(A&S), the Vice Chairman of the Joint Chiefs of Staff, and DoD Component chief information officers across the DoD on enterprise services, cybersecurity, supply chain risk management, interoperability, and applicable software acquisition procedures.

2.5. DIRECTOR, OPERATIONAL TEST AND EVALUATION (DOT&E).

The DOT&E:

a. For programs on the DOT&E oversight list, approves the adequacy of test strategies (including the projected level of funding) and test plans for operational test and evaluation in connection with the program.

b. Assesses results of operational test and evaluation conducted by the programs on the DOT&E oversight list for test adequacy, and whether the results of adequate tests confirm the program is effective, suitable, and survivable for operational use.

2.6. DIRECTOR, COST ASSESSMENT AND PROGRAM EVALUATION.

The Director, Cost Assessment and Program Evaluation:

a. Consults with the USD(A&S) as appropriate on policies and guidance for the software acquisition pathway.

b. Advises the USD(A&S) on schedules, resource allocation, affordability, systems analysis, and cost estimation for programs using the software acquisition pathway.

c. Establishes policies and procedures for the collection of cost data and conduct of cost estimates for programs using the software acquisition pathway.

2.7. DOD COMPONENT HEADS.

The DoD Component heads:

a. Establish streamlined and coordinated requirements, budget, and acquisition processes to support iterative requirements development; continuous user engagement and feedback; and rapid fielding of software applications and of software upgrades to software embedded in systems.

b. Oversee their software acquisition programs through their component acquisition executives (CAEs) and DA.

c. Through the CAE, consult with the USD(A&S) as appropriate on policies and guidance for the software acquisition pathway. CAEs serve as DA for programs using the software acquisition pathway unless the USD(A&S) designates the program as a special interest program or delegated to a designated official. CAEs tailor and continuously improve software acquisition procedures within their component to enable rapid and effective acquisition and delivery of software capabilities. This includes delegating decisions and approvals to the lowest level practicable.

d. Through the DA, the DoD Component heads oversee programs that use the software acquisition pathway in accordance with this issuance and related component policies. DAs designate a program manager (PM) for each program using the software acquisition pathway and supports them in tailoring and streamlining processes, reviews, and decisions to enable speed of capability delivery. DAs are responsible for providing required program data to the USD(A&S) to support management and continuous improvement of the software acquisition pathway.

2.8. VICE CHAIRMAN OF THE JOINT CHIEFS OF STAFF.

The Vice Chairman of the Joint Chiefs of Staff:

a. Consults with the USD(A&S) as appropriate on policies and guidance for the software acquisition pathway.

b. Maintains a library of capability needs statements (CNSs) for programs using the software acquisition pathway.

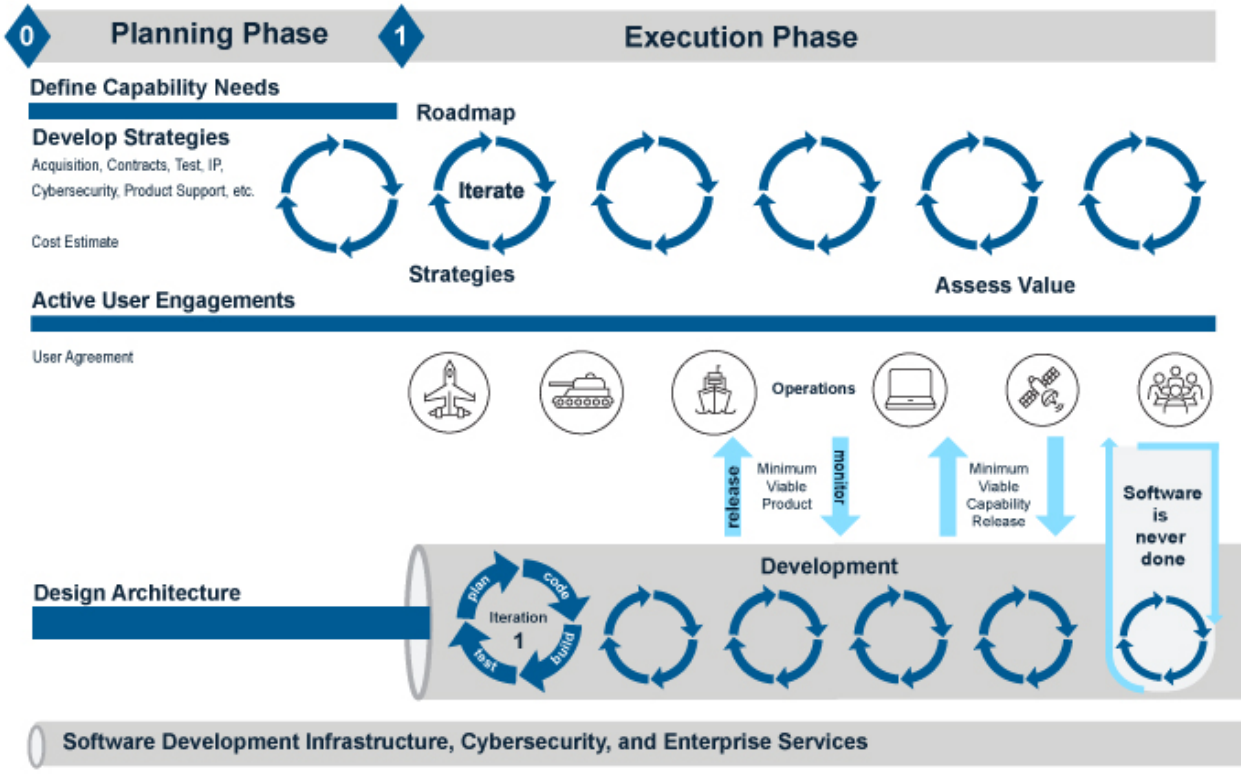
c. Advises the DoD CIO and other DoD Component heads on interoperability across the joint force, cybersecurity of military networks, and alignment with future warfighting concepts.

SECTION 3: PROCEDURES

3.1. GENERAL PROCEDURES.

a. A rapid, iterative approach to software development reduces costs, technological obsolescence, and acquisition risk. To allocate resources to the most relevant capability needs, DoD or DoD component leadership will make software acquisition and development investment decisions within a framework that addresses tradeoffs between capabilities, affordability, risk tolerance, and other considerations. The software acquisition pathway has two phases: planning and execution. Figure 1 outlines key activities and artifacts of the two phases that enable rapid and iterative software development and delivery.

Figure 1. The Software Acquisition Pathway



b. There are two paths within the software acquisition pathway: applications and embedded software. Except where specifically noted, the guidance in this issuance applies to both paths equally.

(1) The applications path provides for rapid development and deployment of software running on commercial hardware, including modified hardware, and cloud computing platforms.

(2) The embedded software path provides for the rapid development, deployment, and insertion of upgrades and improvements to software embedded in weapon systems and other

military-unique hardware systems. The system in which the software is embedded could be acquired via other acquisition pathways (e.g., major capability acquisition).

c. The DA will document the decision and rationale for a program to use the software acquisition pathway in an acquisition decision memorandum.

d. Existing acquisition programs may elect to update their acquisition strategy to transition to the software acquisition pathway or use it in addition to their current acquisition pathway. The PM and applicable stakeholders will identify, and the DA will approve, a transition approach to tailor processes, reviews, and documentation to effectively deliver software capabilities.

e. Value assessments will be performed at least annually after the software is fielded to determine if the mission improvements or efficiencies realized from the delivered software are timely and worth the current and future investments from the end user perspective. More frequent value assessments are encouraged if practical.

3.2. PLANNING PHASE.

a. Purpose.

The purpose of this phase is to better understand the users' needs and plan the approach to deliver software capabilities to meet those needs.

b. Phase Description.

(1) The planning phase will be guided by a draft CNS developed by the operational community. The sponsor will approve the CNS before the execution phase starts.

(a) The program office and related stakeholders will actively engage users throughout the software lifecycle to understand their mission deficiencies, required enhancements to existing operational capabilities, cybersecurity requirements, features, interoperability needs, legacy interfaces, intelligence needs, threat intelligence, and other desired attributes.

(b) All required capabilities in the CNS should be prioritized to effectively guide the software development.

(c) Periodic review of the CNS should occur at least as often as each value assessment to determine if updates are warranted.

(2) During the planning phase, the DA will select a PM and may establish a new program office or assign an existing program office to shape and plan the software acquisition. The PM will develop a constrained, tailored set of strategies to acquire, develop, and deliver the software capabilities, and will obtain the necessary resources (e.g., people, funding, technology) to effectively execute the strategies.

(3) The program should begin developing the software design and architecture, leveraging existing enterprise services as much as possible.

(a) The program will also consider the development environment, processes, automated tools, designs, architectures, and implications across the broader portfolio, component, and joint force.

(b) The chosen software development methodology will incorporate continuous testing and evaluation, resiliency, and cybersecurity, with maximum possible automation, as persistent requirements and include a risk-based lifecycle management approach to address software vulnerabilities, supply chain and development environment risk, and intelligence threats throughout the entire lifecycle.

(c) The program may develop prototypes or initial capabilities to explore possible solutions, architecture options and solicit user and stakeholder feedback.

(4) The DA will validate there are appropriate strategies, analysis, and resources are in-place to successfully transition from the planning phase to the execution phase. The approved artifacts required to enter the execution phase include the CNS, user agreement (UA), acquisition strategy, test strategy, and cost estimate. The DA may have a meeting with key stakeholders to review the program's strategies to make the decision. Programs using the embedded software path align their strategies with the programs into which they will be integrated.

(5) Programs using the software acquisition pathway will be identified in component and DoD program lists and databases within 60 calendar days of initiating the planning phase in accordance with DoD's implementation of Section 913 of Public Law 115-91 on acquisition data analysis.

c. Requirements.

(1) Programs using the software acquisition pathway are not subject to JCIDS, except pursuant to a new process as discussed in Paragraph 2.8.a., but must be effective in capturing users' needs, priorities, and environment. The sponsor will oversee development of a draft CNS to support the initiation of a software acquisition and use of this pathway. The CNS should be clear and concise. Programs using the embedded path will align the CNS with the requirements documents of the system(s) the software will be embedded.

(2) Each DoD Component will create streamlined requirements processes to develop, coordinate, and approve the CNS commensurate with the size, scope, risks, threats, and urgency of need. The Joint Staff will determine if joint equities are involved and will execute an expedited joint validation process if necessary. Insights gained during the planning phase will be incorporated into the CNS before approval. The PM will actively engage with the sponsor during the CNS development to ensure operational and technical feasibility. The sponsor will approve the CNS before entry into the execution phase. Approval of the CNS should be delegated to the lowest level practical based on the size, risk, complexity, and interdependency of the software needs.

(3) Current acquisition programs with approved JCIDS documents that transition to the software acquisition pathway may continue to use them as the basis of requirements or develop a CNS to capture current, software-unique needs. The sponsor will periodically update the CNS throughout development as required to reflect the current high-level operational needs for the software solution. The sponsor will submit the approved CNS document into the DoD Knowledge Management and Decision Support system for awareness and archival.

(4) Frequent user engagements are critical to the success of modern software development to ensure delivered software capabilities address their priority needs.

(a) The sponsor and PM will develop a UA before the execution phase to gain commitment to continuous user involvement and assign decision-making authority for the development and delivery of software capability releases. Decisions include defining and prioritizing required capabilities, tradeoffs of software features and cadence, user acceptances, and readiness for operational deployment.

(b) The UA will commit proper resourcing of operational user involvement to provide acquirers, developers, and testers insights into the operational environment, provide feedback on interim and fielded software, support test and evaluation, and shape future requirement details (e.g., user stories and features).

d. Acquisition Strategy.

(1) PMs will develop and execute an approved acquisition strategy. The acquisition strategy is an integrated plan that identifies the overall approach to rapidly and iteratively acquire, develop, deliver, and sustain software capabilities to meet the users' needs.

(2) Consistent with modern software development practices, the acquisition strategy and related program documentation will be tailored to what is needed to effectively manage the program.

(a) The PM will actively collaborate with program stakeholders and functional experts in developing the acquisition strategy given the current environment, priorities, risks, and approach.

(b) The DA will approve the acquisition strategy to include process and documentation tailoring.

(c) The acquisition strategy for an embedded software system must align with and may be included as part of the platform acquisition strategy.

(d) The PM will mature the strategy to the point where it has sufficient rigor for the DA to approve beginning development, and will continuously refined it throughout the acquisition lifecycle.

(3) Key elements of the acquisition strategy include, but are not limited to:

(a) Risk-based business and technical management approach to rapidly and iteratively deliver software capabilities balanced against quality, security, intelligence threats, system safety, performance, and other factors.

(b) Roadmap and cadence for software deliveries to operations including:

1. Demonstrating the viability and effectiveness of capabilities for operational use not later than 1 year after the date on which funds are first obligated.

2. Continuously delivering capabilities to operations at least annually thereafter.

3. If using the embedded software path, aligning and integrating with the development and fielding for the systems in which the software is embedded.

(c) Flexible and modular contract strategy that enables software development teams to rapidly design, develop, test, integrate, deploy, and support software capabilities.

(d) Planned use of government personnel and resources for software activities.

(e) Tailoring of acquisition processes to adopt modern software development practices (e.g., lean, agile, DevSecOps).

(f) Planned use of existing enterprise services, infrastructure, and resources.

(g) High level test strategies, coordinated with the test and evaluation community, to validate software quality, integration and automation of testing, along with planned test platforms, resources, and infrastructure.

(h) Architecture strategies to enable a modular open systems approach that is interoperable with required systems.

(i) Cybersecurity strategies in accordance with the applicable cybersecurity policies and issuances which include recurring assessment of the supply chain, development environment, processes and tools, continuous automated cybersecurity test and operational evaluation to provide a system resilient to offensive cyber operations.

(j) IP, training, and product support strategies; and records management requirements in accordance with the appropriate DoDIs to ensure lifecycle supportability.

(k) The PM's strategy to ensure that the program is conducted in accordance with all applicable laws and regulations (e.g., Division E of Public Law 104-106, safety, sustainment, communication waveform management and standardization, and airworthiness) throughout the lifecycle.

e. IP Strategy.

In accordance with DoDI 5010.44, each program will develop and regularly update an IP strategy based on the unique characteristics of the program. The IP strategy will identify and

describe the management of delivery and associated license rights for all software and related materials necessary to meet operational, cybersecurity, and supportability requirements. The IP strategy must support and be consistent with all other government strategies for design, development, test and evaluation, operation, modernization, and long-term supportability of the software, protection of the software supply chain, and must be implemented via appropriate requirements in the contracts.

(1) The PM should understand the rights and obligations of both the government and industry, as well as the system and software architecture and lifecycle requirements, in order to shape program strategies and negotiate for computer software deliverables and license rights.

(2) The IP strategy will include, to the maximum extent practicable, negotiation for and periodic delivery of: all executables, source code, associated scripts, build procedures, automation scripts, tools, databases, libraries, test results, data sets, firmware, training materials, and any other elements necessary to integrate, test and evaluate, debug, deploy, and operate the software application in all relevant environments (e.g., development, staging, and production). Data sets and records should include those that support operations and mission-related decisions. Furthermore, it should address delivery of all software components where the government will have rights to the source code, such as open source software and software developed at government expense; and a list of all third-party software components included in the software. The delivery of software source code should support activities such as compilation and debugging, and future requirements for software sustainment over the lifecycle of the program.

(3) The IP strategy should address collaboration with other potential developers and users of software, in cases where the government will be taking delivery of, and potentially modifying the source code, to reduce unnecessary duplication. For example, the strategy should address when and how the program will either accept or provide improvements to software component source code from other government agencies or to an open-source project managed outside the DoD. To the extent practicable, the PM will avoid creating program-specific versions of software components.

(4) The PM will implement mechanisms to ensure any restrictive markings on software and software documentation deliverables markings and assertions conform to contract terms and conditions before acceptance by the government.

(5) The PM will approve the use of any commercial or proprietary software that has not been previously identified in the IP strategy before its insertion into the software developed for the government. Before approval, the PM will assess the software licensing agreement against the IP strategy to ensure that any government unique IP rights are negotiated and enumerated in the contract license agreement. The PM will comply with the license requirements associated with all IP associated with commercial or proprietary software.

(6) The PM, as much as practicable, will require that any commercial or proprietary software used in or interoperable with software developed for the government has documented open interfaces to allow for technology insertion, and to support the use of modular open systems approaches. PM's will ensure that a holistic approach is used to ensure the government's requirements are satisfied; this will be addressed in detail in Service policy and guidance.

(7) The IP strategy should identify technological areas where IP may result from government investment and treat those appropriately.

(8) The PM should require delivery of all the source code for software developed at the government expense, including all software capability descriptions (e.g., features, story points, use cases) and all as-built architecture and design products, traceability products, interface definitions including interfaces to proprietary software elements, and any other requisite documentation. Delivery timelines should plan accordingly for transition to a different contractor or to the government. This facilitates managing program risk, and supports options for software transition to another organization for sustainment. The PM will define the software transition plan in a lifecycle support plan and should identify the point of transition in the product roadmap.

f. Test Strategy.

(1) The test strategy defines the streamlined processes by which capabilities, features, user stories, use cases, etc., will be tested and evaluated to satisfy developmental test and evaluation criteria and to demonstrate operational effectiveness, suitability, interoperability, and survivability, including cyber survivability for operational test and evaluation. The strategy will:

(a) Identify key independent test organizations and their roles and responsibilities and establish agreements on how they will be integrated early into the planning and development activities throughout the software lifecycle.

(b) Encourage and identify test artifacts that can and will be shared across the testing and certification communities (e.g., developmental test and evaluation, operational test and evaluation, system safety assessments, and security certification).

(c) Identify the tools and resources necessary to assist in data collection and transparency to support developmental test and evaluation and operational test and evaluation objectives.

(d) For programs executing the embedded software path, include a safety critical risk assessment and mitigation strategy, including any safety critical implications.

(e) Include a strategy to assess software performance, reliability, suitability, interoperability, survivability, operational resilience, and operational effectiveness.

(f) Programs using the embedded software path will align test and integration with the testing and delivery schedules of the overarching system in which the software is embedded, including aligning resources and criteria for transitioning from development to test and operational environments.

(2) Automated testing and operational monitoring should be used as much as practicable for user acceptance and to evaluate software suitability, interoperability, survivability, operational resilience, and operational effectiveness. The DA will approve the test strategy and DOT&E will be the final approver on test strategies for programs on the DOT&E Oversight List. The test strategy should include information on the verification, validation, and accreditation

authority, approach, and schedule for models and simulations in accordance with applicable modeling and simulation policies. Continuous runtime monitoring of operational software will provide health-related reporting (e.g., performance, security, anomalies), as well as additional data collection opportunities to support test and continuous operational test.

g. Cost Estimates.

Before the execution phase begins, a cost estimate must be developed for the program.

(1) The cost estimate will be developed in accordance with DoDI 5000.73. The estimate should consider the technical content of the program described in the CNS, UA, acquisition strategy, and test strategy.

(2) The initial cost estimate must be completed before entry into the execution phase and must be updated annually.

(3) Where applicable, cost and software data reporting, to include software resources data reports, must be submitted in accordance with the policies and procedures in DoDI 5000.73.

(4) Cost estimates for programs using the embedded pathway will align and integrate with those of the systems in which the software is embedded.

h. Life Cycle Product Support Strategy.

(1) The PM will develop a product support strategy in accordance with applicable DoDIs that treats software development as the continuing evolution of capability across the lifetime of the system, rather than assume discrete "acquisition" and "sustainment" phases. Such a strategy will incorporate early integration of key stakeholders and planning for supportability of the software from program inception, in order to facilitate software maintenance upgrades and evolution in key activities throughout the development. If using the embedded software path, the product support strategy should be aligned with the overall sustainment strategy for the weapon system. The strategy should consider concurrent program activities that may span multiple funding appropriations.

(2) The strategy will address contracting for tailored technical data in order to enable seamless transition of the software and its support to another organization, if and when needed. The strategy will discuss how key enabling resources (e.g., a continuous authority to operate (cATO), if applicable, automated test environments and support, or a selected development environment) will transition to government or other sources of software engineering competence. The strategy will include how any transitions allow for continuous testing and monitoring, and address the need to provide subject matter experts and/or ensure all software engineering staff are trained in the tools, techniques, and environments.

3.3. EXECUTION PHASE.

a. Purpose.

The purpose of this phase is to rapidly and iteratively design, develop, integrate, test, deliver, and operate resilient and reliable software capabilities that meet the users' priority needs.

b. Phase Description.

(1) Programs will assemble software architecture, infrastructure, services, pipelines, development and test platforms, and related resources from enterprise services and development contracts. Leveraging existing services from enterprise services and development contracts will be preferred over acquiring new services to the extent consistent with the program acquisition strategy and IP strategy.

(2) Programs will maximize use of automated software testing and security accreditation, continuous integration and continuous delivery of software capabilities, and frequent user feedback and engagement. Programs will consider the program's lifecycle objectives and actively manage technical debt. Programs will use modern, iterative software practices to continuously improve software quality (e.g., iteratively refactor design and code, reduce cybersecurity vulnerabilities, and create effective modular open systems approaches to support future capabilities). Programs using the embedded software path will align test and integration with the overarching system testing and delivery schedules.

(3) The sponsor and program office will develop and maintain a product roadmap to plan regular and iterative deliveries of software capabilities. The product owner and program office will also develop and maintain program backlogs that identify detailed user needs in prioritized lists. The backlogs allow for dynamic reallocation of current and planned software releases. Issues, errors, threats, and defects identified during development and operations, including software updates from third parties or suppliers, should be captured in the program's backlogs to address in future iterations and releases. Regular stakeholder feedback and inputs will shape the product roadmap and program backlogs.

(4) The PM and the sponsor will use an iterative, human-centered design process to define the minimum viable product (MVP) recognizing that an MVP's definition may evolve as user needs become better understood. Insights from MVPs help shape scope, requirements, and design.

(5) The PM and the sponsor will use an iterative, human-centered design process to define a minimum viable capability release (MVCR) if the MVP does not have sufficient capability or performance to deploy into operations. The MVCR delivers initial warfighting capabilities to enhance mission outcomes. The MVCR for applications programs must be deployed to an operational environment within 1 year after the date on which funds are first obligated to acquire or develop new software capability including appropriate operational test. If the MVP version of the software is determined sufficient to be fielded for operational use, the MVP will become the MVCR.

(a) Subsequent capability releases will be delivered at least annually. Software updates to address cybersecurity vulnerabilities will be released in a timely manner, potentially including out of release cycle as needed, per the program's risk based lifecycle management approach.

(b) Programs should deploy embedded software upgrades at least annually to an environment (e.g., development, staging, or operations) consistent with the overarching weapon system testing delivery strategy.

(6) Programs will continuously improve or refine software development processes, practices, tools, and program strategies to reflect them. They should employ small empowered teams and scale larger efforts across multiple teams. This includes integrating and aligning efforts across government and software development organizations. Continuous user feedback and self-assessments help balance investments between short-term capability deliveries and longer-term enduring solutions.

(7) Software development testing, government developmental testing, and operational testing will be integrated, streamlined, and automated to the maximum extent possible to accelerate delivery timelines based on risk strategies. Automated test scripts and test results will be made available to the test community so that critical verification functions (e.g., performance, reliability), and validation functions (e.g., effectiveness, suitability and survivability) can be assessed iteratively and incrementally.

(8) Automated cyber testing and continuous monitoring of operational software will be designed and implemented to support a cATO or an accelerated accreditation process to the maximum extent practicable; and will be augmented with additional testing where appropriate in accordance with cybersecurity policies, and in coordination with the assigned authorizing official. All safety critical software standards and guidance apply for programs using the software acquisition pathway. Programs will implement recurring cybersecurity assessments of the development environment, processes and tools.

(9) Cybersecurity and software assurance will be integral to strategies, designs, development environment, processes, supply chain, architectures, enterprise services, tests, and operations. Continuous and automated cybersecurity and cyber threat testing will identify vulnerabilities to help ensure software resilience throughout the lifecycle. PMs will work with stakeholders to provide sufficient controls to enable a cATO where appropriate. Ensuring software security includes:

(a) Secure development (e.g., development environment, vetted personnel, coding, test, identity and access management, and supply chain risk management).

(b) Cybersecurity and assurance capabilities (e.g., software updates and patching, encryption, runtime monitoring, and logging).

(c) Secure lifecycle management (e.g., vulnerability management, rigorous and persistent cybersecurity testing, and configuration control).

(10) IP considerations will be tracked and managed, and the IP strategy continuously updated accordingly, throughout the execution phase. For example, any changes to the planned use of government-funded and privately-funded modules or components should be reflected in the required listings of asserted restrictions, and the inspection and acceptance of deliverables should include a verification that any markings are consistent (e.g., both conforming and justified) with the anticipated restrictive markings.

(11) Each program will develop and track a set of metrics to assess and manage the performance, progress, speed, cybersecurity, and quality of the software development, its development teams, and ability to meet users' needs. Metrics collection will leverage automated tools to the maximum extent practicable. The program will continue to update its cost estimates and cost and software data reporting from the planning phase throughout the execution phase.

(12) The sponsor and user community will perform a value assessment at least annually on the software delivered. The sponsor will provide feedback on whether the mission improvements or efficiencies realized from the delivered software capabilities are timely and worth the investment. The feedback should be informed by test and evaluation results. The DA, sponsor, and PM will use the value assessments to assess progress on the program, update strategies, designs, and inform resourcing decisions.

(13) The PM will iteratively develop and verify technical training materials that are synchronized with software deliveries throughout the software development lifecycle. The PM will deliver training materials that ensure that receiving users and military units can be trained to the appropriate level of proficiency and readiness to successfully execute the individual and collective tasks necessary to accomplish the mission supported by the software. The PM will deliver technical operator and maintainer manuals required to operate and maintain the system. Digital delivery of software manuals and automated training will be allowed and preferred. Every effort should be made to include all updated software manuals and automated training that are iteratively improved with each new release of software capabilities.

GLOSSARY

G.1. ACRONYMS.

ACRONYM	MEANING
AAF	adaptive acquisition framework
CAE	component acquisition executive
cATO	continuous authority to operate
CNS	capability need statement
DA	decision authority
DAS	Defense Acquisition System
DevSecOps	development, security, and operations
DBS	Defense Business System
DoDI	DoD instruction
DOT&E	Director, Operational Test and Evaluation
IP	intellectual property
JCIDS	Joint Capabilities Integration and Development System
MVCR	minimum viable capability release
MVP	minimum viable product
PM	program manager
UA	user agreement
USD(A&S)	Under Secretary of Defense for Acquisition and Sustainment

G.2. DEFINITIONS.

Unless otherwise noted, these terms and their definitions are for the purpose of this issuance.

TERM	DEFINITION
AAF	A series of acquisition pathways to enable the workforce to tailor strategies to deliver better solutions faster. The AAF acquisition pathways provide opportunities for milestone decision authorities, DAs, and PMs to develop acquisition strategies and employ acquisition processes that match the characteristics of the capability being acquired.

TERM	DEFINITION
backlog	Program backlogs that identify detailed user needs in prioritized lists. The backlogs allow for dynamic reallocation of scope and priority of current and planned software releases. Issues, errors, and defects identified during development and operations should be captured in the program's backlogs to address in future iterations and releases.
capability	Higher level solutions typically spanning multiple releases. Capabilities consist of multiple features to facilitate implementation.
cATO	The core concept of cATO is to build software security into the software development methodology so that the authority to operate process (as with the testing process) is done alongside development. If done correctly, an authority to operate is nearly guaranteed once the software is release ready.
CNS	A high-level capture of mission deficiencies, or enhancements to existing operational capabilities, features, interoperability needs, legacy interfaces and other attributes that provides enough information to define various software solutions as they relate to the overall threat environment.
DA	The official responsible for oversight and key decisions of programs that use the software acquisition pathway in accordance with this issuance and related component policies. The official designates a PM and supports them in tailoring and streamlining processes, reviews, and decisions to enable speed of capability delivery. The official may be the Defense Acquisition Executive, CAE, or the Program Executive Officer, or other designated official by the CAE.
DBS	Defined in Section 2222 of Title 10, United States Code.
DevSecOps	An organizational software engineering culture and practice that aims at unifying software development, security, and operations. The main characteristic of DevSecOps is to automate, monitor, and apply security at all phases of the software lifecycle: plan, develop, build, test, release, deliver, deploy, operate, and monitor. In DevSecOps, testing and security are shifted left through automated unit, functional, integration, and security testing – this is a key DevSecOps differentiator since security and functional capabilities are tested and built simultaneously.
embedded software	Software with a dedicated function within a larger mechanical or electrical system, often with real-time computing constraints, or software applications embedded in a platform (e.g., air vehicle,

TERM	DEFINITION
	ground vehicle, or ship). In the context of this issuance, embedded software does not apply to firmware or software dedicated to controlling devices.
end user	Those who will ultimately use the software solution. Users convey operational concepts, requirements, and needs, participate in continuous testing activities, and provide feedback on developed capabilities.
enterprise services	Services that have the proper scope to play a productive role in automating business processes in enterprise computing, networking, and data services. Enterprise services include technical services such as cloud infrastructure, software development pipeline platforms, common containers, virtual machines, monitoring tools, and test automation tools. Responsibility for these functions is generally above the PM.
features	A service or distinguishing characteristic of a software item (e.g., performance, portability, or functionality) that fulfills a stakeholder need and includes benefit and acceptance criteria within one release. Features are used to complete capabilities and are comprised of multiple stories (or tasks, use cases, etc.).
government developmental testing	Testing intended to verify and demonstrate how well the system under development meets its technical compliance requirements, to provide data to assess developmental risk for decision making, and to ensure that the technical and support problems identified in previous testing have been corrected.
interoperability	The ability of systems, units or forces to provide data, information, materiel, and services to, and accept the same from, other systems, units, or forces and to use the data, information, materiel and services so exchanged to enable them to operate effectively together. Interoperability includes information exchanges, systems, processes, procedures, organizations, and missions over the life cycle and must be balanced with cybersecurity
modern software development practices	Practices (e.g., lean, agile, DevSecOps) that focus on rapid, iterative development and delivery of software with active user engagements. Small cross-functional software development teams integrate planning, design, development, testing, security, delivery, and operations with continuous improvement to maximize automation and user value.

TERM	DEFINITION
MVCR	The initial set of features suitable to be fielded to an operational environment that provides value to the warfighter or end user in a rapid timeline. The MVCR delivers initial warfighting capabilities to enhance some mission outcomes. The MVCR is analogous to a minimum marketable product in commercial industry.
MVP	An early version of the software to deliver or field basic capabilities to users to evaluate and provide feedback on. Insights from MVPs help shape scope, requirements, and design.
operational acceptance	When one or more military units decides to use the software in military operations as informed by test and evaluation.
product owner	A role on the program or development team that works closely with the user community to ensure that the requirements reflect the needs and priorities of the user community, and align to the mission objectives.
product roadmap	A high-level visual summary that maps out the vision and direction of product offerings over time. It describes the goals and features of each software iteration and increment.
release	A grouping of capabilities or features that can be used for demonstration or evaluation. A release may be internal for integration, testing, or demonstration; or external to system test or as user delivery. A release may be based on a time block or on product maturity.
software-intensive	A system in which software represents the largest segment in one or more of the following criteria: system development cost, system development risk, system functionality, or development time.
sponsor	The individual that holds the authority and advocates for needed end user capabilities and associated resource commitments.
task	Individual activities to be completed to satisfy a user story or use case (e.g., implement code for a specific feature or complete design for a specific feature).
technical debt	Consists of design or implementation constructs that are expedient in the short term but that set up a technical context that can make a future change costlier or impossible. Technical debt may result from having code issues related to architecture, structure, duplication, test coverage, comments and documentation, potential bugs, complexity,

TERM	DEFINITION
	coding practices, and style which may accrue at the level of overall system design or system architecture, even in systems with great code quality.
UA	A commitment between the sponsor and PM for continuous user involvement and assigned decision making authority in the development and delivery of software capability releases.
use case	In software and systems engineering, a use case is a list of actions or event steps, typically defining the interactions between a user and a system (or between software elements), to achieve a goal. Use cases can be used in addition to or in lieu of user stories.
user acceptance	Verification by operational users that software is capable of satisfying their stated needs in an operationally representative environment.
user story	A small desired behavior of the system based on a user scenario that can be implemented and demonstrated in one iteration. A story is comprised of one or more tasks. In software development and product management, a user story is an informal, natural language description of one or more features of a software system. User stories are written from the perspective of an end user or user of a system.
value assessment	An outcome-based assessment of mission improvements and efficiencies realized from the delivered software capabilities, and a determination of whether the outcomes have been worth the investment. The sponsor and user community perform value assessments at least annually, to inform DA and PM decisions.

REFERENCES

Defense Acquisition University Software Acquisition Pathway Guidance[1]

DoD Directive 5000.01, "The Defense Acquisition System," September 9, 2020

DoD Directive 5135.02, "Under Secretary of Defense for Acquisition and Sustainment (USD(A&S))," July 15, 2020

DoD Instruction 5000.02, "Operation of the Adaptive Acquisition Framework," January 23, 2020

DoD Instruction 5000.73, "Cost Analysis Guidance and Procedures," March 13, 2020

DoD Instruction 5000.75, "Business Systems Requirements and Acquisition," February 2, 2017, as amended

DoD Instruction 5010.44, "Intellectual Property (IP) Acquisition and Licensing," October 16, 2019

Public Law 104-106, Division E, "The Clinger-Cohen Act of 1996," February 10, 1996

Public Law 115-91, Section 913, "The National Defense Authorization Act for Fiscal Year 2018," December 12, 2017

Public Law 116-92, Section 800, "The National Defense Authorization Act for Fiscal Year 2020," December 20, 2019

United States Code, Title 10

1 Available at https://aaf.dau.edu/aaf/software/