Software Acquisition *Go Bag*

What is the Software Acquisition Pathway (SWP)?

Who should use this document?

You! This document is for **everyone** who wants to learn more about the DoW's SWP. Whether you are a member of leadership, software personnel, engineering personnel, program manager, contracts specialist, costing specialist, logistics specialist, tester, warfighter/user, requirements owner, development contractor, or sustainment organization representative, this is for you!

What will I learn from this document?
You will learn the basics about the SWP in an easy-to-follow guide. If you're already familiar with the SWP, you can skip to our first Tactical Guide called *SWP: Ready, Set, Go!*

What do I need to know?

What is the SWP?
The SWP is a DoW policy. It is documented in DoD Instruction 5000.87.[1]

What is the SWP for? Why was it created?
The SWP's objective is "to facilitate rapid and iterative delivery of software capability to the user."[2]

Before the SWP policy was released in October 2020, most acquisition policy and guidance was designed for hardware-centric programs using milestones that measured readiness to proceed to the next phase.

The SWP was designed for software-intensive systems. It uses industry best practices and modern software development approaches to deliver software capability rapidly and iteratively. It was created using lessons learned from many years of studies and pilot programs.[3,4]

Acronyms

ADM	Acquisition Decision Memorandum
CARD	Cost Analysis Requirements Description
CNS	Capability Needs Statement
DA	Decision Authority
DoD	Department of Defense
DoDI	Department of Defense Instruction
DoW	Department of War
JCIDS	Joint Capability Integration and Development System
MDAP	Major Defense Acquisition Program
MVCR	Minimum Viable Capability Release
MVP	Minimum Viable Product
PM	Program Manager
SEI	Software Engineering Institute
SWP	Software Acquisition Pathway
UA	User Agreement
VA	Value Assessment
WAU	Warfighting Acquisition University

Why has adopting the SWP become a big focus?
The March 6, 2025 memo *Directing Modern Software Acquisition to Maximize Lethality* by the Secretary of War, explains "It is a top priority for DoD to reform its acquisition processes in order to acquire, deliver, and iterate on our weapon and business systems—including software—at speed and scale for our Warfighter."[5] You can also learn more at our article on this topic at *New DoD Memo Accelerates Software Acquisition Modernization*.[6]

Is the SWP different from the way that we acquired software in the past?
Yes! The SWP was intentionally designed to remove traditional acquisition constraints to more closely align to commercial software development practices. The SWP

- provides a tailorable process for software acquisition that you can use to rapidly produce software to meet your mission needs
- encourages your program to take a risk-based approach and tailor acquisitions to your program needs
- requires the knowledge and use of modern software development methods such as Agile, Lean, DevSecOps, and automated testing (explained in more detail on the right)
- requires your program to accelerate software deployment to at least annually, with a goal of weekly or monthly where mission appropriate
- encourages your users to be involved in every stage of the process, including the assessment of the value the program office delivered to support the mission capability
- provides relief from certain administrative processes—no MDAP treatment and a streamlined JCIDS process of no longer than 45 days

> We know that some acquisition and administrative processes are changing, and we're tracking that carefully. We'll update this guide when new processes, like a replacement for JCIDS, are released.[7]

- uses a simplified acquisition strategy but still includes considerations such as cybersecurity, a modular open systems approach, model-based engineering, human-centered design, and enterprise services

Agile? Lean? DevSecOps? What is that about?
The SWP requires the knowledge and use of modern software development methods such as Agile, Lean, DevSecOps, and automated testing. It also requires the use of enterprise services. Below, we explain what these terms mean:

Modern Software Development practices include, for example, Lean, Agile, and DevSecOps. These practices focus on the rapid, iterative development and delivery of software with strong and frequent user engagement. Small cross-functional software development teams integrate planning, design, development, testing, security, delivery, and operations with continuous improvement to maximize automation and deliver value more frequently to the user.[8]

Agile is both a mindset and an approach to software development that emphasizes collaboration, adaptability, iterative delivery, and customer value. Agile is guided by the values and principles originally defined in the Agile Manifesto.[9] You may hear about different types of Agile software development. The term "Agile" describes a family of iterative and incremental approaches, such as Scrum and Kanban as well as frameworks for scaling.

Lean is a methodology for continuous process improvement which focuses on workflow, customer value, and eliminating process waste. Lean is uniquely different from traditional process improvement strategies because its primary focus is eliminating non-value-added activities.[10]

DevSecOps is 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 testing is a key DevSecOps differentiator, since security and functional capabilities are tested and built simultaneously.[11]

Enterprise Services are 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 generally rests with officials above the PM.

What are the key elements or artifacts of the SWP?
The SWP uses key elements (or artifacts) that you might not be familiar with. Below, we provide their names and definitions:

Definitions derived from DoDI 5000.87 unless otherwise noted

A program ***Backlog*** identifies 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.

A ***Capability Needs Statement*** (CNS) is a high-level capture of mission deficiencies, or enhancements to existing operational capabilities, features, interoperability needs, legacy interfaces, and other attributes, that provide enough information to define various software solutions as they relate to the overall threat environment.

A ***Cost Estimate*** (CE) is a generic term meant to cover the various types of cost estimates that might be required for the SWP. These estimates include independent cost estimates, program cost estimates, and component cost estimates.[12]

A ***Minimum Viable Capability Release*** (MVCR) is 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.

A ***Minimum Viable Product*** (MVP) is an early version of software to deliver or field basic capabilities to users to evaluate and provide feedback on. Insights from MVPs help shape product scope, requirements, and design.

A ***Product Roadmap*** is a high-level visual summary that maps out the vision and direction of product offerings over time. A product roadmap describes the goals and features of each software iteration and increment.

A ***Release*** is a grouping of capabilities or features that can be used for demonstration or evaluation. A release can be internal to development for integration, testing, or demonstration; or it can be external to development for system test or user delivery. A release can be based on a time block or on product maturity.

A ***User Agreement*** is 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.

A ***Value Assessment*** is 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.

How do the key elements (or artifacts) work together?
In Figure 1 (below) we explain how each key element interacts with the other key elements. The interactions can be complex, but don't get too concerned. The main point is to show the influence between them, so you understand that they aren't all stand-alone key elements.

Figure 1. Influence of Key Elements

What are the SWP phases?

The SWP doesn't have traditional acquisition milestones. It has two phases: planning and execution. Figure 2 depicts those two phases.

The planning phase is meant to last fewer than six months but can be adjusted to meet program needs.

The execution phase includes the delivery of an MVP and MVCR, which we explained previously.

Future releases shown in Figure 2 are intended to depict iterative capability releases throughout the life of the program.

There are a lot of details in Figure 2, but don't worry. You'll see this diagram again in future Tactical Guides and Supplements when we explain more about the artifacts, activities, and events that happen in these two SWP phases.

What are the SWP sub-paths?

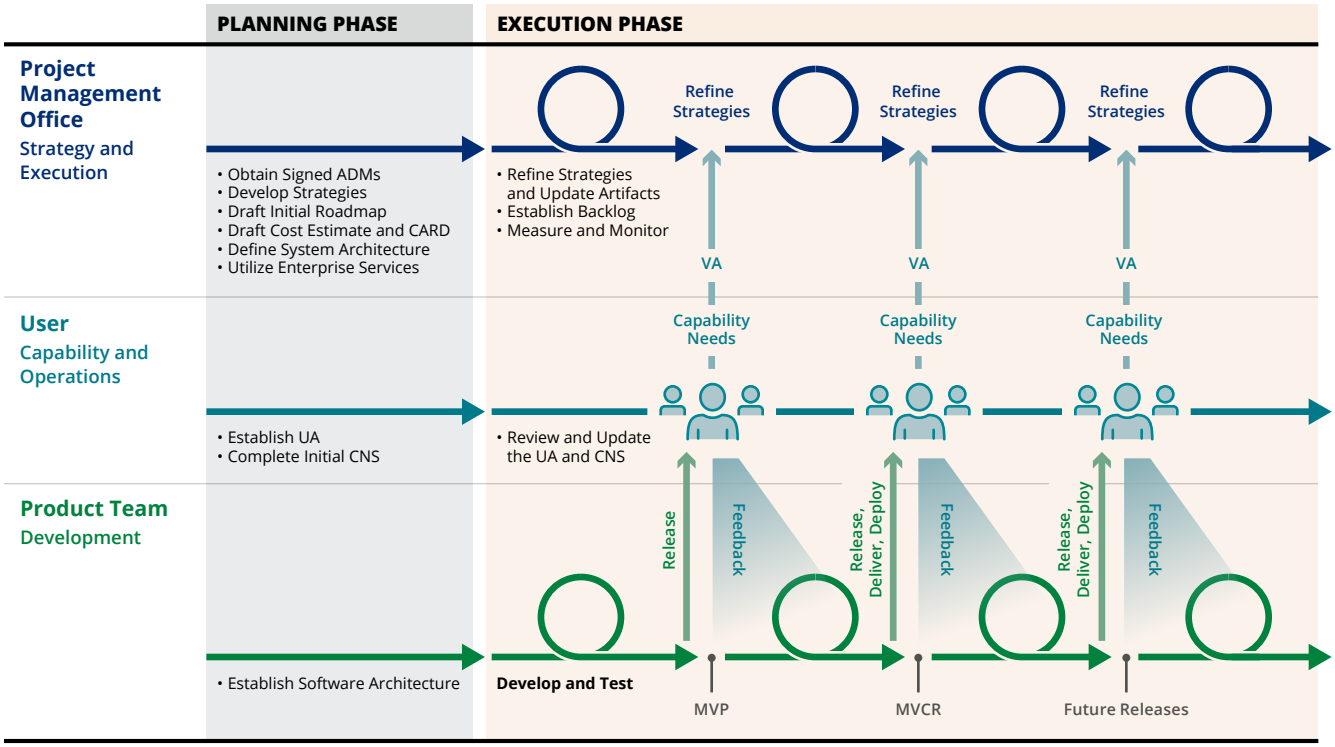
Since there are different types of software that might be on the SWP, there are sub-paths that are customized to account for those differences. In Table 1, we describe how each sub-path is used.

Table 1. SWP Sub-paths

Icon	Sub-path	Used for
	Applications	Rapid development and deployment of custom software running on commercial hardware and cloud computing platforms
	Embedded	Rapid development and insertion of upgrades and improvements into software embedded in weapon systems and other military-unique hardware systems
	Business	Rapid development and deployment of critical business system capabilities

The two phases as well as the artifacts and activities in the SWP are the same no matter which sub-path your program chooses. There are some nuances that you'll need to plan for depending on your sub-path. Stick with us, and we'll explain this in future Tactical Guides.

Figure 2. Software Acquisition Pathway - DoDI 5000.87*



*This diagram does not include all interactions and cadences.

What's next?

This document introduces you to the basics of the SWP. If you're ready to move to the next step, head on over to our website to download our first Tactical Guide, *SWP: Ready, Set, Go!*

Future Tactical Guides will help you with completing activities involved in the SWP and developing SWP artifacts. Stay tuned!

How do I learn more?

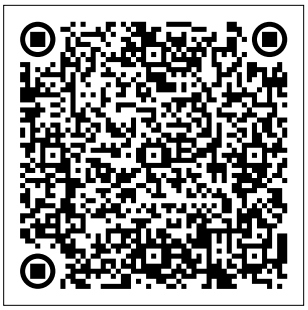
- Visit our Go Bag website at sei.cmu.edu/projects/gobag.
- Unclear about all the groups and roles involved in the SWP? Grab our *Software Acquisition Go Bag User's Guide* from the Pack Your Go Bag link.
- Learn more about the SWP on WAU's Adaptive Acquisition Framework website.[2]

What if I have questions or want to provide feedback?
We hope you do because your Feedback will help us support your journey! Send us questions or feedback at our Contact Us link.

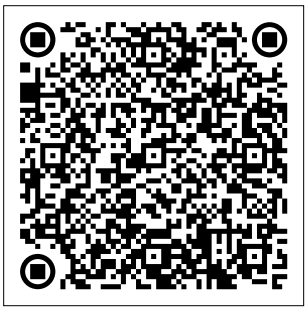
What if I need more help?
Reach out to us at our Contact Us link so that we can discuss how the SEI might assist with your specific challenges.

How can I stay engaged with Software Acquisition Go Bag?
We don't want you to miss out on anything! Subscribe to our mailing list to participate in future Go Bag webinars and be notified about Tactical Guide and Supplement launches.

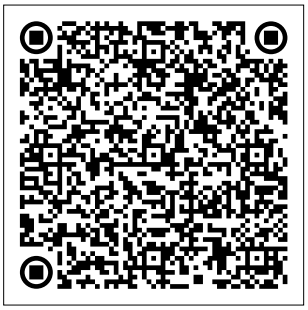
Don't take the journey alone! Invite your program office co-workers, leadership, and stakeholders to join you on this journey by recommending that they also subscribe.



Learn more about the Go Bag project »



Send questions or feedback to our Contact page »



Subscribe to our mailing list for updates »

Endnotes

1 Department of Defense. Operation of the Software Acquisition Pathway. DoDI 5000.87. October 2, 2020. https://www.esd.whs.mil/Portals/54/Documents/DD/issuances/dodi/500087p.PDF?ver=virAfQj4v_LgN1JxpB_dpA%3D%3D

2 Warfighting Acquisition University. Software Acquisition. *Adaptive Acquisition Framework Website*. October 7, 2025 [accessed]. https://aaf.dau.edu/aaf/software/

3 Defense Innovation Board. Software Acquisition and Practices Study. *DIB Website*. October 7, 2025 [accessed]. https://innovation.defense.gov/software/

4 Department of Defense Office of the Under Secretary of Defense for Acquisition and Sustainment. *Agile Software Acquisition Guidebook: Best Practices & Lessons Learned from the FY18 NDAA Section 873/874 Agile Pilot Program*. February 27, 2020. https://www.dau.edu/sites/default/files/Migrated/CopDocuments/AgilePilotsGuidebook%20V1.0%2027Feb20.pdf

5 Secretary of Defense. Directing Modern Software Acquisition to Maximize Lethality [memo]. March 6, 2025. https://media.defense.gov/2025/Mar/07/2003662943/-1/-1/1/DIRECTING-MODERN-SOFTWARE-ACQUISITION-TO-MAXIMIZE-LETHALITY.PDF

6 Software Engineering Institute. New DoD Memo Accelerates Software Acquisition Modernization. *SEI Website*. March 13, 2025. https://www.sei.cmu.edu/news/new-dod-memo-accelerates-software-acquisition-modernization/

7 Roque, A. et al. Pentagon Terminating JCIDS Process as Part of Larger Acquisition Reform: Memo. *Breaking Defense Website*. August 22, 2025. https://breakingdefense.com/2025/08/pentagon-terminating-jcids-process-as-part-of-larger-acquisition-reform-memo/

8 This explanation was derived from DoD Instruction 5000.87: Department of Defense. Operation of the Software Acquisition Pathway. DoWI 5000.87. October 2, 2020. https://www.esd.whs.mil/Portals/54/Documents/DD/issuances/dodi/500087p.PDF?ver=virAfQj4v_LgN1JxpB_dpA%3D%3D

9 Beck, K. et al. Manifesto for Agile Software Development. Agile Manifesto Website. 2001. https://agilemanifesto.org/

10 Department of Defense. Implementation and Management of the DoD-Wide Continuous Process Improvement/Lean Six Sigma (CPI/LSS) Program. DoDI 5010.43. July 17, 2009. https://www.esd.whs.mil/Portals/54/Documents/DD/issuances/dodi/501043p.pdf

11 This explanation was derived from DoD Instruction 5000.87: Department of Defense. Operation of the Software Acquisition Pathway. DoDI 5000.87. October 2, 2020. https://www.esd.whs.mil/Portals/54/Documents/DD/issuances/dodi/500087p.PDF?ver=virAfQj4v_LgN1JxpB_dpA%3D%3D

12 This definition was derived from DoD Instruction 5000.73: Department of Defense. Cost Analysis Guidance and Procedures. DoDI 5000.73. October 24, 2024. https://www.esd.whs.mil/Portals/54/Documents/DD/issuances/dodi/500073p.pdf

Software Acquisition *Go Bag*

About the SEI

Always focused on the future, the Software Engineering Institute (SEI) advances software as a strategic advantage for national security. We lead research and direct transition of software engineering, cybersecurity, and artificial intelligence technologies at the intersection of academia, industry, and government. We serve the nation as a federally funded research and development center (FFRDC) sponsored by the U.S. Department of Defense (DoD) and are based at Carnegie Mellon University, a global research university annually rated among the best for its programs in computer science and engineering.

Contact Us

CARNEGIE MELLON UNIVERSITY
SOFTWARE ENGINEERING INSTITUTE
4500 FIFTH AVENUE; PITTSBURGH, PA 15213-2612

sei.cmu.edu
412.268.5800 | 888.201.4479
info@sei.cmu.edu

©2025 Carnegie Mellon University | 6554 | C 11.20.2025 | S 12.11.2025

This material is based upon work funded and supported by the Department of Defense under Air Force Contract Nos. FA8702-15-D-0002, and FA870225DB003 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center.

DM25-1562