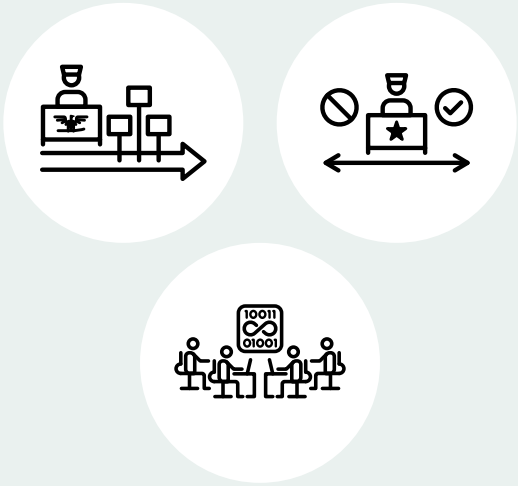
Software Acquisition *Go Bag*

Tactical Guide

03

Pack Light, Measure Right



Who should use this guide?

This guide is for everyone in a **Program Management Office** that develops, acquires, delivers, or sustains software capabilities regardless of which acquisition pathway or development model they follow. It is also helpful to **Decision Authority**/Leadership and **Product Teams** who deliver software capability. Whether you're building new mission software on the Software Acquisition Pathway, modernizing legacy systems, or integrating software as part of a larger hardware effort, your program will benefit from understanding how software metrics offer insights into the health, effectiveness, and efficiency of the mission capability your program develops.

See our User's Guide to learn more about roles.

What will I learn from this guide?

This Tactical Guide helps your program select only the measures that "earn their keep." Measurements should reduce the burden of software programs; they already carry enough weight in their daily operations. This guide supports your program on its journey of

- developing a lightweight metrics plan that can support complex programs as they evolve
- selecting a minimal but sufficient set of measures to get the insight you need
- implementing measurement approaches compatible with iterative delivery
- obtaining and using Product Team data to increase transparency and collaboration

We want your program to have the right metrics that provide better insight, early enough to matter.

Acronyms

ATO	Authority to Operate
CI/CD	Continuous Integration and Continuous Delivery
CIO	Chief Information Officer
DA	Decision Authority
DoDI	Department of Defense Instruction
DoW	Department of War
GPS	Global Positioning System
PMO	Program Management Office
OUSW (A&S)	Office of the Undersecretary of War for Acquisition and Sustainment
SWP	Software Acquisition Pathway

This guide is published with a companion Supplement called *Decoding SWP Metrics*. Future Supplements will explore deeper dives into identifying your program's unique metric needs, ensuring your program can obtain the right data, and getting the most out of your metrics.

Why should I care?

Metrics are about more than just compliance. They ground you in fact, enable informed decision making, and help your team understand performance while providing insight into program health. By understanding the why, what, and how behind metrics, your team can move from "just collecting data" to using metrics that enable better decision making, improve program outcomes, and deliver mission value.

Regardless of the pathway your program is using, we urge you not to view metrics as "extra work." It's imperative to see them as real-time evidence that depicts how the work is being accomplished and whether you will deliver timely and accurate warfighting capability. *You are the linchpin* that ensures your program plans, measures, and delivers software that creates real value for the warfighter and the missions they serve while also complying with policy.

What do I need to know?

Understanding how to use metrics to make program decisions is important. We'll start with examples from everyday life:

Think about the dashboard in your car.

The speedometer isn't there to satisfy the Department of Motor Vehicles. It provides real-time information that helps you stay safe, follow speed limit laws, and ultimately avoid tickets.

The fuel gauge is not a display the manufacturer simply provided. It helps you plan ahead for gas tank fill-ups so you don't run out of gas in the middle of nowhere.

The GPS isn't just for fun. It provides directions that help you reach your destination efficiently.

Metrics for your program should work the same way. They should

- guide your decisions
- make you aware of risks
- provide insights that help you navigate situations as they arise
- confirm you are on track to deliver mission-critical capability

Many of you lead or support a software acquisition program today. Whether you realize it or not, you are navigating a space where policy and regulation set boundaries, and modern commercial practices set the pace. It can be difficult to connect what policy expects, what strategies your program puts into place, what contracts enforce, and what Product Teams can realistically execute, all while delivering value to the mission. Software metrics give your PMO team the insights it needs to navigate within that very space.

Overview

The best metrics are those that enable insights into your program and its everyday decisions. Software problems rarely emerge suddenly. They grow quietly through slow bottlenecks in requirement refinement, the lack of automation in pipelines, test failures creeping upward, increasing backlog age, slower release cycles, increased time to resolve defects, and more. Without metrics and data, these indicators remain invisible until they manifest as a missed delivery, blown budget, or incomplete capability at an operational demonstration. Let's take a moment to align our understanding of key terms and definitions:

Measure
A variable to which a value is assigned as the result of measurement[1]

Data
A collection of values assigned to base measures, derived measures, or indicators[1]

Indicator
A measure that provides an estimate or evaluation of specified attributes derived from a model with respect to defined information needs[1]

> For simplicity, throughout this guide, we commonly use the term "metrics" to encompass measures, data, and indicators.

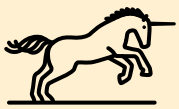
SWP policy requires programs to collect metrics to monitor progress and enable informed decision-making:

Section 3.3.b.(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."*[2]

Section 1.2.l

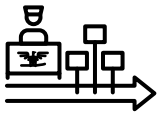
This section also requires programs to submit a specific set of metrics **semiannually** to aid OUSW (A&S) in monitoring the effectiveness of the SWP.[2]



Many programs say,
"I just need to collect the metrics required by the SWP."

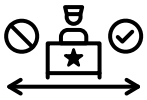
SWP-required metrics will get you off to a good start, but they are not sufficient by themselves for monitoring your program's progress, quality, and performance. The original intent of the SWP metrics was to help monitor and improve the pathway, not to measure program performance, but used correctly they can be helpful. Our Supplement, called *Decoding SWP Metrics*, provides a handy mapping that shows how the required SWP metrics can be an important part of your metrics plan and how it can enable key insights for Product Teams, program managers, DAs, and the policy owner.

Roles and Their Associated Metrics Responsibilities



Program Management Office

- Translates intent into execution by ensuring the program has real-time visibility and access to metrics, not just compliance artifacts
- Ensures the contract supports the collection and continuous improvement of metrics in partnership with the Product Team
- Uses metrics to find patterns not just problems to be proactive instead of reactive



Decision Authority

- Creates the conditions for speed and accountability by requiring metrics-based status and transparency not just documentation
- Ensures the PMO is driving decisions from live data instead of curated narratives and treats loss of visibility as a program risk not just an inconvenience

Product Team

- Delivers software with the built-in ability to provide metrics making progress, quality, and risk visible without extra effort
- Develops a single source of truth from the beginning by building transparency into the workflow and not treating it as an extra reporting layer; Should reflect reality in real-time
- Enables access to real-time data by default and restricts access only with clear and agreed upon justification
- Uses metrics to guide and improve their performance

Planning for Software Metrics

We use measurements to gain insight and enable planning, managing, and communicating that are objective. Measurement alone does not create control or improvements. While it may be possible to meet every traditional metric (i.e., on schedule, on budget, at cost), the program may find that it is not delivering real value to the User and the mission. Let's get started with understanding the ways that you can plan to use metrics in your software-intensive program.

The goal is not to create extra documentation; however, creating this lightweight metrics plan can support the metrics section of your acquisition strategy. A metrics plan also becomes a shared approach (to achieve alignment) across your program to collect, access, interpret, and act on data. It allows your team to understand progress over time, not just track activity and outputs. The metrics plan should serve as a living document that evolves with the program's maturity, tools, and delivery cadence.

Here are a few things to consider while planning your program's approach to metrics:

1. Align metrics to provide the insights you need for everyday decisions and to achieve outcomes.
If you are a SWP program, you have a Capability Needs Statement, User Agreement, and Value Assessment, all of which lay out the intended outcomes for your program. As you align your metrics to the desired outcomes, it's important to note that some metrics assess how the process supports the outcomes. Examples of these supporting metrics are provided in our Supplement called *Decoding SWP Metrics*.

A good test for determining if your metrics are aligned to outcomes is this heuristic: If a metric goes up or down and no action is taken, the metric isn't aligned to an outcome. Outcome-aligned metrics typically influence a decision, trigger a conversation, or change a behavior. This heuristic helps ensure that you are not only packing light, but you are also measuring right!

2. Start with questions, then use metrics to answer the questions.

Identify the information your program needs to make decisions and deliver mission value. Metrics should answer questions similar to the following. These example questions are derived from the DoW CIO's *The State of DevSecOps* report to give your program a head start:[3]

- Did we build the right thing?
- Did we build the product right?
- What do we want to understand?
- Is the product delivered quickly and frequently based upon user and mission need?
- Is the product adaptable to change?
- Is the product responsive to user feedback?
- Will I stay within cost?
- Will testing be completed on time?
- Will my ATO be ready in time?
- Will the different Product Teams be ready on time for the planned integration?
- Are the agreed-to capabilities going to make it into the release?
- Is the capability reliable for users under operational conditions?
- How quickly can we turn a requirement into a deployed capability?
- As we expand the user base, where are the fragile, high-risk areas of the codebase?

> This list is not all-inclusive; it is intended to guide your program. There is no end-to-end master metrics list. Your program's specific context is key to determining which questions are necessary to support your program needs.
>
> Your program may need to collect and monitor metrics about business and acquisition practices that support software development, such as hiring, obtaining clearances, and procuring equipment and facilities.

Example: The PMO might ask, "Can the development team reliably deliver software when we say it will?" You don't need 20 delivery metrics, just enough to answer the question. Let's break down the question:

Question	Metric
Can we deliver predictably?	Regular Deployment Frequency
Is the software stable when we deliver?	Change Fail Rate
When things break, do we recover fast?	Mean Time to Restore

Example: The DA/Leadership and stakeholders might ask, "Do I need to intervene, or can the team continue executing?" Let's see what metrics can answer that question as we break it down:

Question	Metric
Is software delivery stable?	Deployment Frequency + Change Fail Rate
Are we delivering value?	Value Assessment Rating
Are we prioritizing speed over quality?	Recidivism Rate

> Our Supplement *Decoding SWP Metrics* provides more insights and definitions for each of the metrics listed above.

3. Instrument your data sources once and reuse them often.

Whether your program has a single Product Team or multiple teams, it's important to establish how the teams work, what data sources they pull from, how the ticketing system works, and how the security tools and CI/CD pipeline are instrumented to enable real-time access to metrics. Manual counting or late-to-need data exchanges are no longer needed between the PMO and development teams. Our Supplement, *Decoding SWP Metrics*, provides a robust set of data sources for each metric. Here are a few examples of potential data sources you might use:

- CI/CD pipeline release dates
- a roadmap or work-tracking application that shows completed capabilities
- reports from a tasking tool
- release status data
- testing completion data
- failure reports from the field
- User-submitted value assessments

4. Track progress over time.
Metrics are most valuable when they show trends and trajectories. A single snapshot/moment in time never tells the full story. Longitudinal data (how metrics evolve across multiple reporting periods) reveals patterns, early warning signs, and opportunities for improvement. *A single metric value is rarely meaningful*; progress lives in trends, trajectories, and deltas over time.

> Programs don't fail because one metric is "red." They fail when trends go red and no one reacts. If you are not reviewing metrics in a time-series, you are not measuring progress; you are merely taking snapshots.

5. Tailor metrics to the audience.
Different stakeholders require different levels of insight. Identify who needs what data, when they need it, and why (i.e., for which decisions). The development team, PMO, executive leadership, and stakeholders all need different views of progress over time. As part of your metrics plan, it's important to clearly establish these needs and ensure that each set of stakeholders understands how to ask questions and interpret the data they access.

6. Treat data as a strategic asset.
Data informs decision making at all levels of the organization and program. To maximize value, data should be defined, collected, and curated. When using data, the following guidelines are useful; we originally published these in the *The State of DevSecOps* study:[3]

The best data is the data used every day. Operational data is used by the local organization to manage day-to-day business. This effort provides ongoing validation of the data's relevance and ensures it's up to date.

- **Manage to mission value, not metrics.**
  The metric is not the objective— it just tells you how you're doing against the mission objective. Use the metrics to guide toward the outcome. The focus is not just tracking technical metrics but understanding how they drive value for defense missions.
- **Don't rely on a single metric.**
  A single measure never tells the whole story. A variety of carefully chosen metrics paints a complete picture. While the same data should be used to derive insight at all levels, neither the same metrics nor the same analyses are appropriate for all purposes.
- **Data can be aggregated, but metrics can't.**
  Metrics have already combined data, often in complicated ways. Don't combine the data again without carefully checking the math. Often, the metric used is a proxy, and not a direct measure.

Planning and Putting Metrics to Use

So, how will you incorporate metrics into your program operations? Below we show you the steps you can take to use metrics in your program effectively. Your program's specific context is an important factor when operationalizing your metrics. Stay tuned for future Supplements that will provide greater detail.

1. Plan before you measure.
Before collecting data, take time to define what your program needs to learn, prove, and improve. Every metric should serve a clear purpose in helping your team understand value, risk, and performance. Metrics should not just satisfy policy requirements. This is a mindset shift you'll need to make: You're not measuring for oversight or compliance, you're measuring to ask better questions; drive learning; and make smarter, data-informed decisions. From the program's beginning to its end, you must work with Stakeholders and Users to understand not only their requirements but also their information needs. Once you understand both, your program can determine what metrics and data can help meet those needs.

2. Design the metrics architecture.
Don't wait until review time to pull screenshots and spreadsheets. Build metrics into how your program actually works. By automating data collection through your CI/CD tools, issue trackers, and test suites, your team has visibility into accurate, real-time insights with reduced effort. In addition, using an automated dashboard can also ease the burden of metrics analysis and provide different metrics to different roles within the PMO. Designing your metrics architecture early can turn reporting into a natural outcome of daily work, and the result is decision-ready information for everyone when it's needed most.

3. Create an implementation plan (with roadmaps, roles, and responsibilities).
Once you know what to measure and why, create a clear plan that defines who owns, manages, and interprets your metrics. Metrics don't stay static; they evolve and require maintenance to stay meaningful. A plan that assigns roles, establishes baseline data, and integrates metrics into contracts and reviews helps your team stay aligned, accountable, and ready to act on what the data shows.

4. Decide how to approach accessing metrics data and incentivizing improvements.

Access to data shouldn't be an afterthought. It's essential for transparency and trust. Build read-only access to development tools and dashboards into your acquisition and contracting strategies so that insights flow freely between the government and development teams. Use clear contract language and incentives to encourage timely access, reliable reporting, and continuous visibility into performance.

5. Build feedback mechanisms and sustain continuous improvement.

Collecting metrics isn't the endpoint, it's the beginning of continuous learning. To create real value, programs must use their data to establish feedback loops for continuous improvement. Connect development teams, the PMO, leadership, and Users to ensure everyone understands how decisions impact delivery and mission outcomes. When you embed these loops into processes, cadences, and governance routines, your program moves from simply reporting metrics to actively improving performance, adapting faster, and delivering greater mission value.

What's next?

Head on over to the Pack Your Go Bag link on our website to download the Supplements for this Tactical Guide[4]

• Supplement: *Decoding SWP Metrics*

Future Tactical Guides and Supplements are in progress, and they will provide more help with software acquisition.

Stay tuned!

How do I learn more?
On the Adaptive Acquisition Framework's website[5]

• learn more about the SWP

• learn more about SWP Metrics and Reporting

On the SEI's website

• read Actionable Data From the DevSecOps Pipeline[6]

• read The Current State of DevSecOps Metrics[7]

On the DoW CIO's website

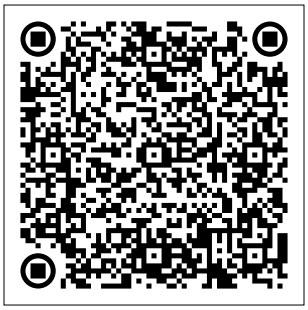
• read The State of DevSecOps Report

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 on the Contact page.

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

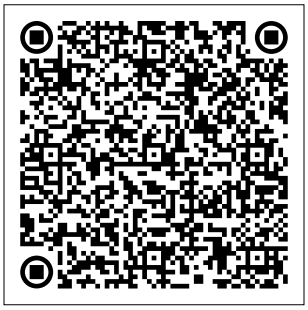
How can I stay engaged with the Software Acquisition Go Bag?
We don't want you to miss out on anything! Subscribe to our mailing list to participate in future Go Bag webinars and be notified about Tactical Guide and Supplement launches.

Don't take the journey alone! Invite your program office, leadership, stakeholders, and other co-workers 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 International Organization for Standardization, International Electrotechnical Commission, and Institute of Electrical and Electronics Engineers (ISO/IEC/IEEE). *Systems and Software Measurement Process*. ISO/IEC/IEEE 159392017. 2017. https://cdn.standards.iteh.ai/samples/71197/d6e958bc13ef44d188b9a5bfc3fc5042/ISO-IEC-IEEE-15939-2017.pdf

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

3 U.S. Department of Defense (DoD). The State of DevOps. DoD. 2025. https://dodcio.defense.gov/Portals/0/Documents/Library/DevSecOpsStateOf.pdf

4 Software Engineering Institute (SEI). Pack Your Go Bag. *SEI Website*. March 31, 2026 [accessed]. https://www.sei.cmu.edu/projects/gobag/pack-your-go-bag/

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

6 Nichols, Bill & Cohen, Julie B. Actional Data from the DevSecOps Pipeline [blog post]. *SEI Blog*. May 1, 2023. https://www.sei.cmu.edu/blog/actionable-data-from-the-devsecops-pipeline/

7 Nichols, Bill. The Current State of DevSecOps Metrics [blog post]. *SEI Blog*. March 29, 2021. https://www.sei.cmu.edu/blog/the-current-state-of-devsecops-metrics/

Software Acquisition *Go Bag*

About the SEI

The Software Engineering Institute (SEI) advances software as a strategic advantage for national security through research, development, and deployment of tools, technologies, and practices in software engineering, artificial intelligence (AI), cyber, and acquisition transformation. We serve the nation as a federally funded research and development center (FFRDC) sponsored by the U.S. Department of War (DoW).

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

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.

DM26-0408