TEST AND EVALUATION CHAPTER 5: SOFTWARE ACQUISITION

Table of Contents

1. Software Acquisition Pathway Overview

1.1 Introduction

The Software Acquisition Pathway is used for the timely acquisition of software capabilities developed for the DoD. Programs using the Software Acquisition Pathway are required to deliver the first increment of viable and effective capability no later than one year after funds are obligated, after which new capabilities must be delivered to operations at least annually to iteratively meet requirements, but more frequent updates and deliveries are encouraged where practical.[21]

Testing organizations should be involved with the acquisition program early and continually throughout its lifecycle to support effective and efficient evaluations and delivery timelines. Contractor development testing, government developmental testing, system safety assessment, security certifications, and operational test and evaluation should be integrated, streamlined, and automated to the maximum extent practicable to enable rapid analysis of test data and evaluation of system operational effectiveness, suitability, and survivability to inform the decision authorities. Maximum sharing, reciprocity, availability, and reuse of test results and artifacts among testing and certification organizations are necessary for success.

This chapter describes T&E community involvement throughout the Software Acquisition Pathway lifecycle.

1.2 Software Acquisition Pathway Description

There are two paths within the Software Acquisition Pathway: applications and embedded software. This T&E guidance applies to both paths. Unique considerations for the embedded software path are highlighted throughout the document.

- **Applications Path.** Provides for rapid development and deployment[22] of software running on commercial hardware, including modified hardware and cloud computing platforms.[23]
- **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).[24]

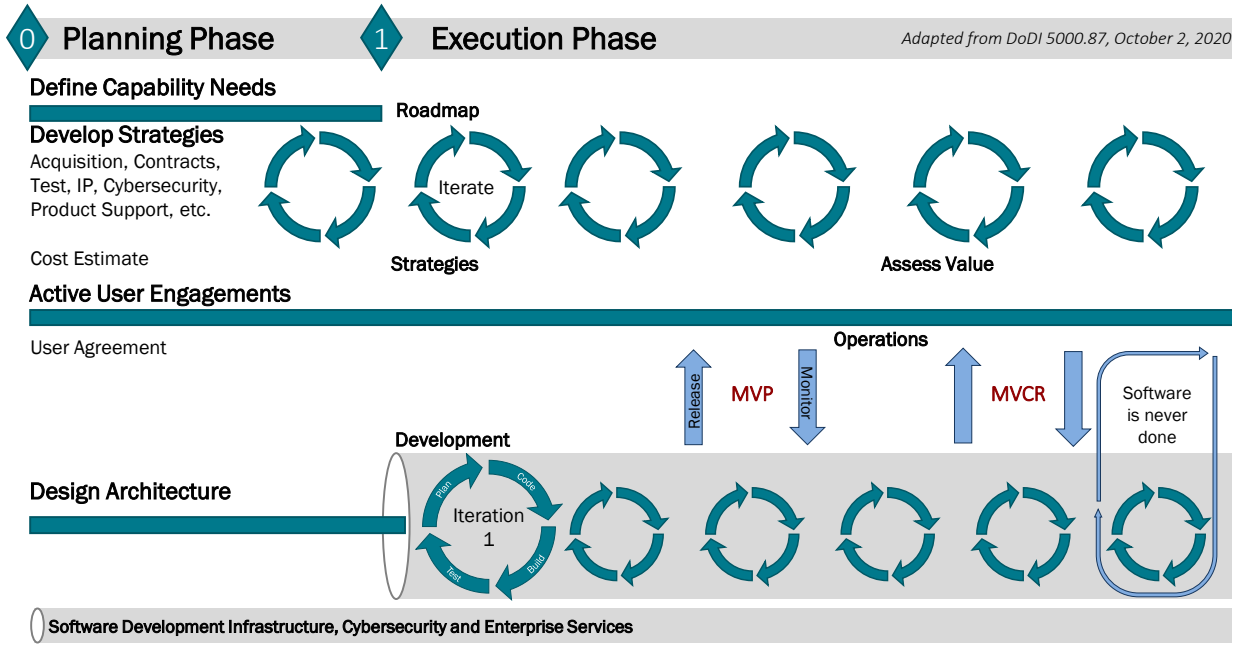
Independent of the path, the Software Acquisition Pathway has two phases: planning and execution, depicted in Figure 1.

[21] DoDI 5000.87

[22] Deployment is when the code reaches the operational users.

[23] DoDI 5000.87

[24] DoDI 5000.87



Acronyms: DoDI – DoD Instruction; IP – Intellectual Property; MVP – Minimum Viable Product; MVCR – Minimum Viable Capability Release

Figure 1. Software Acquisition Pathway

1.2.1 Planning Phase

The purpose of the Planning Phase is to better understand users' needs and plan the approach to deliver software capabilities to meet those needs.[25] As the Planning Phase sets the conditions for success, test teams should be involved early in the program during the Planning Phase to establish and document how testing will be accomplished. Details of T&E Community involvement during the Planning Phase are discussed in Section 2.

1.2.2 Execution Phase

During the Execution Phase, the software is designed, developed, integrated, tested, delivered, deployed, operated, and monitored. Programs will spend the majority of their life cycles in the Execution Phase. Activities during this phase will be guided by the product roadmap, which identifies goals and features of the software.

The Software Acquisition Pathway stresses the concept of iterative development, which includes iterative software development methods (e.g., Agile, DevSecOps) tools, and automation (e.g., automated test scripts). Readers can consult the *DoD Enterprise DevSecOps Fundamentals*[26] published by the DoD Chief Information Officer (CIO) for more information on methods and tools, which provides a compendium of universal concepts related to DevSecOps, as part of a library of guidebooks, playbooks, and reference designs.[27] The DevSecOps library provides deep knowledge and industry best practices that can directly benefit program offices and intermediate

[25] DoDI 5000.87, pg. 9

[26] https://dodcio.defense.gov/Portals/0/Documents/Library/DoDEnterpriseDevSecOpsFundamentals.pdf

[27] Library of documents is available here: https://dodcio.defense.gov/Library/

staff. In particular, there is a document on DevSecOps Tools and Activities that testers should reference for potential use in testing strategies. For additional information on Agile concepts and terms, readers can refer to the DAU Agile 101 Primer[28] and Agile Software Acquisition Guidebook[29].

The iterative process is highlighted through the "Plan, Code, Build, Test" components of each development cycle, as labeled in Iteration 1 of Figure 1. It includes delivering and deploying software in small increments that build on each other.

As shown by the "test" component of each development cycle, testing occurs throughout the iterative development process. This includes contractor testing and independent government testing. For programs using the embedded software path, this testing should be aligned with the system in which the software is embedded. Details of government testing and test team involvement throughout the Execution Phase are discussed in Section 3.

1.2.2.1Minimum Viable Product (MVP)

The MVP is developed during the Execution Phase and is an "early version of the software to deliver or field basic capabilities to the users to evaluate and provide feedback on. Insights from MVPs help shape scope, requirements, and design."[30] Note that the MVP is not intended to be fielded for operational use.

T&E of the MVP is discussed in Section 3.3.1.

1.2.2.2Minimum Viable Capability Release (MVCR)

The MVCR is developed during the Execution Phase and contains "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."[31] The MVCR delivers initial warfighting capabilities to enhance some mission outcome and is intended to be fielded to an operational environment for operational use.

The MVCR must be deployed to an operational environment within one 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 may become the MVCR.

T&E of the MVCR is discussed in Section 3.3.2.

1.2.2.3 Value Assessments

During the Execution Phase, "the sponsor"[32] 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

---

[28] https://www.dau.edu/cop/it/DAU%20Sponsored%20Documents/Agile%20101%20v1.0.pdf

[29] https://www.dau.edu/cop/it/DAU%20Sponsored%20Documents/AgilePilotsGuidebook%20V1.0%2027Feb20.pdf

[30] DoDI 5000.87, Glossary

[31] DoDI 5000.87, Glossary

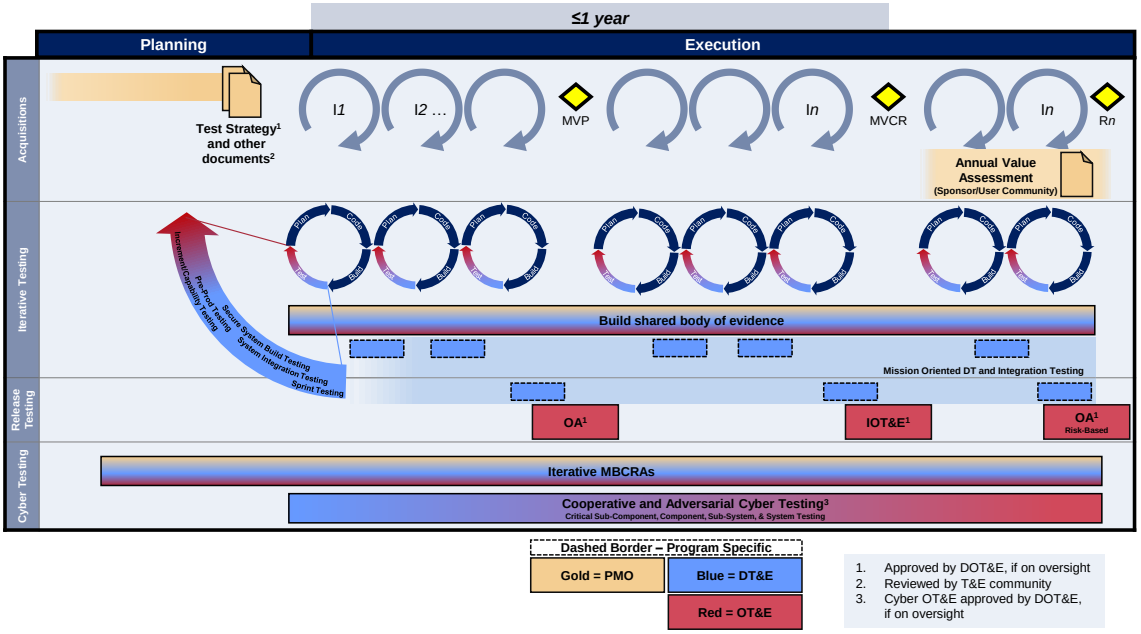
[32] DoDI 5000.87 defines the sponsor as "the individual that holds the authority and advocates for needed end user capabilities and associated resource commitments." This guidance identifies further roles within the sponsor organization.

capabilities are timely and worth the investment. The feedback should be informed by test and evaluation results."[33] Support from T&E for these value assessments is discussed in Section 3.5.

1.3 Software Acquisition Pathway T&E Overview

During the Planning Phase, the test teams should be involved in developing acquisition documents and establishing the testing infrastructure, tools, and data requirements. The T&E Strategy is developed and written during this phase, and approved by DOT&E if on oversight.

During the Execution Phase, contractor and independent government test teams should continuously test and evaluate the software being developed. Figure 2 summarizes how this guidance envisions T&E across the Software Acquisition Pathway. It highlights both testing



activities and evaluation products throughout the acquisition lifecycle. The testing and evaluation shown in this figure is described throughout this chapter. The iterative testing line is described throughout Section 3.2. The release testing line is described in Section 3.3. Cyber testing is briefly described in Section 3.2.6.

Acronyms: DOT&E – Director, Operational Test and Evaluation; T&E – Test and Evaluation; DT&E – Developmental Test and Evaluation; OT&E – Operational Test and Evaluation; OA – Operational Assessment; IOT&E – Initial Operational Test and Evaluation; MBCRA – Mission Based Cyber Risk Assessment

Figure 2. T&E Aligned with Software Acquisition Pathway

1.4 Test and Evaluation Working-level Integrated Product Team (WIPT)

The T&E WIPT coordinates top-level planning for all test events, and assists in the evaluation of test results to support systems engineering and programmatic decision-making.

The T&E WIPT is conducted in an open forum that includes the test and evaluation subject matter experts responsible for supporting the Program Manager (PM) on all aspects of the test and evaluation effort, including:

[33] DoDI 5000.87, pg. 18

- T&E program strategy, design, development, oversight
- Analysis, assessment, and reporting test results

The PM should charter the T&E WIPT during the Planning Phase so that it is involved with the program's Acquisition Strategy, contract requirements for T&E, and test plan development. The T&E WIPT also assists the PM in managing the T&E program throughout the lifecycle of the software acquisition. The PM should also ensure that T&E resources and other requirements needed to adequately plan and execute the T&E program are coordinated with the T&E community (to include operational test community). T&E resources should be articulated in requests for proposals (RFPs) and other acquisition documents that will affect the contractual requirements and availability of information to the T&E WIPT.

The T&E WIPT consists of representatives from all organizations responsible for providing or overseeing the T&E Strategy and its execution. In particular, the T&E WIPT should include test data stakeholders such as systems engineering, the Lead Developmental Test Organization, Chief Developmental Tester, Operational Test Agency (OTA), D,DTE&A (for programs on DTE&A oversight), DOT&E (for programs on DOT&E oversight), cybersecurity lead, interoperability evaluator, the Capability Owner, and applicable certification authorities. Roles and responsibilities for T&E WIPT members and participants should be documented in a T&E WIPT Charter.

In developing the T&E Strategy, the T&E WIPT should ensure it is executable and aligns with the acquisition strategy, T&E policy (DODI 5000.89), and relevant T&E focus area chapters in the T&E Enterprise Guidebook. T&E Strategy development, content, and approval is described further in Section 2.1.

The T&E WIPT should participate in requirements definition and refinement activities to understand the rationale behind the requirements, and to ensure their measurability, testability, and achievability. These activities should address both high-level needs and evolving requirements. The PM should ensure that the T&E WIPT is enabled to coordinate with the requirements authority to clarify any requirements found untestable.

1.5 Roles and Responsibilities

1.5.1 Developmental Test Teams

In iterative development, increased collaboration among independent test teams and developers and users is required.

- Development teams will lead lower level tests such as unit tests, whereas independent test teams will lead integration and acceptance tests. Results from all testing should be captured in a shared body of evidence, a data repository to store test data that all parties can use for independent evaluation.
- Test teams should be involved up front to ensure they get the data they need from the developmental test process.
- Test teams should strive to maintain a tempo for release testing in sync with the development team(s) by using automation for functional, performance, and regression testing.

- Government test teams should develop a robust T&E at the feature and release level with end-to-end mission threads and employing actual users. Refer to Sections 3.2.2 through 3.2.4.
- Evaluating and adjusting the DT&E strategy within the T&E Strategy to stay current with the Capability Needs Statement (CNS).[34] For programs on oversight, D,DTE&A will monitor and adjust the DT&E strategy and oversight involvement.

Development testing will likely employ automated test tools for functional and cyber testing, which will require the government testers to understand and use these tools.

1.5.2 Operational Test Teams

OT&E concentrates on appropriately scoped, dedicated tests while integrating information from all sources to provide usable data that meet stakeholder needs and inform decision makers. The OT&E effort during this phase includes participating via the test activities of each iteration and through dedicated tests to build a shared body of evidence.

Appropriately scoped OT&E aligns with deployment decisions associated with the MVP and the MVCR. Following the MVCR, OT&E continues to follow a risk-informed approach that scopes tests and evaluations to the capabilities delivered. Software Acquisition Pathway programs will spend the majority of their lifetime in risk-informed OT&E following the MVCR.

A risk-appropriate OA is usually required in support of every limited deployment.[35] The OTA should conduct this risk assessment based on DOT&E and Service guidance. For programs on DOT&E oversight, DOT&E approves the risk assessment and operational test plans.

The OT&E strategy includes:

- Scoping the tests to match the capability delivered and proposed for deployment for operational use, and identifying opportunities for OT&E involvement within all Execution Phase activities. The OTA will consult the PMO and DOT&E, for programs on oversight, to scope the tests. DOT&E approves the operational test plans for programs on oversight.
- Providing operational evaluations to inform decisions and products of the Software Acquisition Pathway, including deployment of software releases and program decisions; an important product to support is the annual Value Assessment.
- Evaluating and adjusting the OT&E strategy within the T&E Strategy to stay current with the Capability Needs Statement (CNS); for programs on oversight, the OTA and DOT&E will monitor and adjust the OT&E strategy and oversight involvement.
- Embedding into the software development and testing process during the Execution Phase to collect data from development needed to scope OT&E and support operational evaluations; embedding includes having continuous visibility into the development process, but does not imply that OT&E should develop the software.

[34] Refer to Table for definition of the CNS.

[35] DoDI 5000.89, p30

Embedding OT&E within the development process requires OT&E participation via electronic and physical presence in the activity of the software pipeline or factory. This includes:

- Monitoring the tests that occur throughout the development pipeline to understand and trust the veracity of the automated and manual testing results to support operational evaluations (the OTA should independently Verify, Validate, and Accredit (VV&A)[36] any automated test capabilities that will provide data supporting operational evaluations)
- Participate in defining test requirements that include end-to-end mission threads
- Ensuring the pedigree of test processes establishes the trust for integrating across different types of testing and remotely monitoring tests
- Monitoring the deployment of new software to the production or live environment to inform the evaluation of capability deployment
- Confirming the presence and functionality of deployment procedures provides for continuity of operations, especially for programs deploying software in short time frames, such as continuous delivery strategies

Additionally, the supply chain for the software includes the pipeline, and how its characteristics affect the software. Testers should conduct cybersecurity testing of the pipeline processes that could lead to exploitation of the software under development, and evaluate how the process for moving software from staging to production will affect deployment and influence cyber defensive operations training. The PM should provide testers with information about the software supply chain and pipeline to support test planning and evaluation.

For each increment, even those not intended for deployment, the OTA should observe testing to determine the applicability of the data for OT&E, including the mapping of that data to the critical assessment areas, and identify gaps in data that will inform test planning for future iterations. The OTA should provide a summary of these items to the PM and, for programs on DOT&E oversight, DOT&E.

1.5.3 Additional Software Pathway Roles and Responsibilities

Iterative software development introduces new roles with the user being represented early and throughout the development process. Figure 3 presents a notional description of how these new user roles relate to the traditional software acquisition roles. This is not intended to be a comprehensive list capturing the responsibilities of all stakeholders in the program community, but rather the key relationships between operations/requirements and acquisition leaders.

[36] Testers should refer to the Modeling and Simulation Focus Area for additional information on VV&A.

Figure 3. User Roles in Iterative Software Development

Operational Sponsor. The senior leader that champions the operational needs/requirements and funding, the Operational Sponsor represents the DoD organization(s) that will be the eventual users of the software solution, and:

- Defines the desired value that the solution will provide
- Approves the high-level Capability Needs Statement (CNS)
- Approves the User Agreement (UA) with the PM and provides users for the PM
- Identifies the Product Owner and co-chairs value assessments
- Ensures users and stakeholder inputs are captured and integrated into value assessments

Product Owner.[37] Representing the Operational Sponsor at the program level, the Product Owner:

- Develops the CNS to sufficient detail to guide the execution phase and develops UA, in coordination with the PM, to identify user resources to support the execution phase
- Is responsible for day-to-day requirements management
- Coordinates user community representation and participates with them in requirements identification and prioritization. Works with the PM to scope the MVP/MVCR and manages and prioritizes the program backlog
- Approves acceptance at the feature or release level and validates releases and user acceptance tests
- Works with the Product Owner(s) assigned to the program; leads the periodic value assessment of the software solution

User Community. A group of personnel allocated to support the program through the UA that represent the various persona who will employ the system in military operations.

- Provides acquisition and development communities insights into the operational environment.
- Provides meaningful feedback and evaluation of software developed.
- Participates in demonstrations and testing activities.

[37] Note that some Agile Development documents identify a "Product Owner" as part of the development team, which is different from this Product Owner. The Product Owner on the PM's development team is the PM's interface to the user community to ensure the requirements reflect the needs and priorities of the user and align with mission objectives

2. T&E During the Planning Phase

The purpose of the Planning Phase is to better understand users' needs and plan the approach to deliver software capabilities to meet those needs. During this phase, various stakeholders are developing documentation, summarized and defined in Table, and the testing infrastructure, tools, and data. This section explains the role of T&E in this process needed to set the conditions for success during the Execution Phase.

Table 2. Planning Phase Documents

Artifact	Description	Developed by
Test and Evaluation Strategy a	Defines the processes by which capabilities, features, user stories, use cases, etc. will be tested and evaluated to satisfy developmental test and evaluation criteria, and defines the processes by which the system will be tested to demonstrate operational effectiveness, suitability, interoperability, and survivability.	Program Manager with the T&E WIPT
Capability Needs Statement (CNS) a	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.	Sponsor with support from the Program Manager
User Agreement (UA) a	A commitment between the Sponsor and Program Manager for continuous user involvement and assigned decision-making authority in the development and delivery of software capability releases.	Sponsor and Program Manager
Acquisition Strategy a	An integrated plan that identifies the overall approach to rapidly and iteratively acquiring, developing, delivering, and sustaining software capabilities to meet users' needs.	Program Manager
Intellectual Property (IP) Strategy a	Identifies and describes 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 should 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 should be implemented via appropriate requirements in the contracts.	Program Manager
Cost Estimate	Developed in accordance with DoDI 5000.73 (Cost Analysis Guidance and Procedures). The estimate should consider the	Program Manager

Artifact	Description	Developed by
	technical content of the program described in the CNS, UA, acquisition strategy, and test strategy.	
Request for Proposals	A document used in negotiated acquisitions to communicate government requirements to prospective contractors and to solicit proposals.	Program Manager

a DoDI 5000.87, "Operation of the Software Acquisition Pathway"
b DAU Glossary

2.1 T&E Strategy

The purpose of the T&E Strategy is to guide the activities of test organizations in planning and executing an effective and efficient test process in support of the program and major program decision. The T&E Strategy is the high-level test planning document for the Software Acquisition Pathway.

The T&E Strategy serves as a contract between the PM and all T&E stakeholders for T&E roles and responsibilities, and resources. The T&E Strategy captures processes by which capabilities, features, user stories, use cases, etc., will be tested and evaluated to verify technical requirements; it should also capture the process by which the operational effectiveness, suitability, and survivability of the system will be evaluated. This testing process should be integrated between the contractor testing, developmental testing, and operational testing teams to provide a holistic view of the system. The T&E Strategy should capture the missions the system is intended to perform, evaluation of the system in the context of a unit equipped with it, and all interfacing systems.

The T&E Strategy should identify evaluation focus areas and critical assessment areas from which test teams derive their data requirements to support major program decisions. However, additional critical assessment areas may be included in system evaluation. Further, the T&E Strategy describes the test events and activities that will provide the data necessary to evaluate the system and support acquisition, technical, and program decisions – termed an Integrated Decision Support Key (IDSK) in DoDI 5000.89, which outlines the integrated approach to testing. The T&E Strategy should describe how these data will be accumulated to build a shared body of evidence to support evaluations of the system. Refer to Section 0 for more information about establishing and maintaining the shared body of evidence. Descriptions of cyber testing in the T&E Strategy should align with content described in the Cyber T&E Focus Area.

The Decision Authority approves the T&E Strategy prior to the program entering the Execution Phase. For programs on DOT&E oversight, DOT&E is the final approver for the T&E Strategy.[38] The T&E Strategy should be updated as needed to align with the current Capability Needs Statement.

[38] DoDI 5000.87, p 14

2.1.1 Data

The T&E Strategy should identify the data required to adequately evaluate the system's technical, functional, and operational performance to inform acquisition, technical, and program decisions, and outline an integrated approach to properly size test events and share data. In addition, it should define the conditions under which these data will be collected, and any tools required to manage the data and perform the testing.

2.1.2 T&E Resources

The T&E Strategy should determine the T&E resources required to support it (e.g., facilities, ranges, operational force structure, cyber ranges and test teams, instrumentation and associated support, automated testing tools, software systems integration labs, modeling and simulation (M&S), including the organization that will validate the models, and costs). The strategy should also identify shortfalls that will require investments to meet T&E infrastructure sufficiency, and plan for any Verification, Validation, and Accreditation (VV&A) activities required to accredit the T&E infrastructure for operational test events. T&E funding in the resources section should be consistent with the cost estimate and budget submissions.

2.2 T&E Content and Interests in Other Planning Phase Documents

While the T&E Strategy is the main testing deliverable during the Planning Phase, it relies heavily on each of the other documents outlined in Table. The T&E community should work with the acquisition community on these documents to incorporate needed T&E information. This section highlights T&E content and involvement of test teams in the development of each of these documents.

2.2.1 Capability Needs Statement (CNS)

Test teams should be involved with CNS development early to fully understand the desired capabilities and ensure that these requirements focus on the mission capability. The test teams should also work with their engineering counterparts to ensure requirements traceability from the capability level requirements to user stories exists so that the test teams can evaluate the system.

Test teams should:

- Define the level of requirements best suited for government T&E
- Understand what constitutes "value" and how that will be measured at value assessments after deployments (annually)
- Ensure cyber and interoperability needs are clearly defined in the CNS

While the Software Acquisition Pathway does not require the Joint Capabilities Integration and Development System (JCIDS) process, test teams can use the JCS Cyber Survivability Endorsement and Implementation Guide[39] to define cyber attributes within the CNS.

[39] JCS Cyber Survivability Endorsement and Implementation Guide

2.2.2 User Agreement (UA)

The Software Acquisition Pathway emphasizes user involvement in development. The DoDI 5000.87 requires that software development be done in active collaboration with end users, representing key user groups, to ensure software deliveries address their priority needs, and undergo regular assessment of software performance and risk. The goal of this continuous user engagement and feedback process is to develop software that best satisfies users' needs.

During the UA development, test teams should:

- Ensure the UA includes user participation in government testing to serve as test operators and provide feedback, including support from users and units for test and evaluation as needed
- Establish early contact with the user community and understand how the users expect the system to work

2.2.3 Acquisition Strategy

The Acquisition Strategy should sufficiently describe the development and decisions to convey what information/data testing needs to provide, and when, as well as account for test and evaluation when identifying resource needs.

The Acquisition Strategy sets the schedule within an initial product roadmap (Section 3.1) for delivering the initial capability and the subsequent cadence for delivering additional capabilities. Test teams should:

- Ensure that test requirements and data delivery for the contractor are thoroughly outlined and included with more detail in the RFP
- Ensure that time is allotted in the program schedule for independent government T&E
- Ensure that the Acquisition Strategy addresses a robust cyber T&E program, including the supply chain
- Understand the decision points that will require test data to make informed decisions (for embedded software, this includes the decision points for the system on which the software is embedded)

In addition, the Acquisition Strategy requires a high-level T&E Strategy that describes plans for verification and validation of software quality, integration and automation of testing, and citing the required test platforms, resources, and infrastructure. Test teams should:

- Ensure the test and evaluation strategy portion of the Acquisition Strategy provides a clear description of the test approach, including any M&S needs, so that it can be included in program planning and the separate T&E Strategy document. For embedded systems, this should align with the testing strategy for the system on which the software is embedded.

2.2.4 Cost Estimate

The cost estimate should consider the technical content of the program described in the CNS, UA, Acquisition Strategy, and T&E Strategy. Test teams should ensure that the cost estimate includes all the resources necessary to conduct testing.

2.2.5 Intellectual Property (IP) Strategy

Test teams should provide input to the IP strategy on the ownership of data generated (such as contractor-generated test results) during all phases of testing that would allow building a shared body of test evidence, available to the program throughout its lifecycle. The PM should further consult with the T&E community to determine any access needed to support independent testing and include these accesses in the IP strategy as needed.

2.2.6 Request for Proposal (RFP)

The RFP defines what the government expects from the contractor; if it is not in the RFP and the eventual contract, you will not get it. The T&E Strategy is a source document for the RFP.[40] The PM should consult with test teams to ensure that the RFP supports data collection for government T&E.

At a minimum, a draft T&E Strategy should be included as an attachment to the RFP to clearly tell the contractors what the government-intended testing is.

The test teams should ensure that the following items and activities are requested:

- Government access to contractor test events, M&S, test tools, test data repositories, and test environments
- Contractor test plans, procedures, reports, and data
- Contractor support for government testing

2.3 Test Infrastructure, Tools, and Data

In addition to the documentation, the Test Infrastructure is also established during the Planning Phase, during which test teams should work with the PM as they develop the infrastructure to identify how test data from different environments can be used to support evaluations. For example, data used to support operational test and evaluation should be generated within an operationally representative environment. If the environment is not operationally representative, limitations to the environment should be enumerated and operational evaluations should be based on the context of the environment in which the data were generated. Section 2.3.1 describes different environments and evaluation of these environments for different testing uses.

Likewise, these environments may need to be instrumented with different testing tools to gather metrics needed to support evaluations. Section 2.3.2 describes this instrumentation and evaluation of tools for different testing uses.

Lastly, in order to establish an integrated evaluation approach, data should be shared among all parties. Section 0 describes this sharing of data.

2.3.1 Test Infrastructure

2.3.1.1 Pipelines and Software Factories

To provide for continuous integration and delivery of software to the customer, modern software development has adopted infrastructure frameworks. These frameworks allow for consistent and

[40] DoDI 5000.89, p14

repeatable processes to be followed, and provide visibility into those processes, enabling a level of trust to all functional personnel throughout the lifecycle. Developers, test teams, security engineers, product owners, and users have a common foundation on which to build their processes. It is this core orchestration framework, commonly called a "pipeline," that provides this foundation.

- A pipeline is a collection of tools, processes, and environments designed to move code from the development environment to the production environment. These environments connect much like a physical assembly line, with the output of one environment becoming the input of the next.
- A software factory contains multiple pipelines equipped with tools, process workflows, scripts, and environments to produce a set of software deployable artifacts with minimal human intervention. It automates the activities in the develop, build, test, release, and deliver phases of the Development, Security, and Operations (DevSecOps) lifecycle.

Pipelines and software factories have a cadence for developing and delivering software. Testing, including operationally representative test cases, should be included in this cadence to support the software development.

2.3.1.2 Test Environments

Within the pipelines, there may be a number of environments where testing may take place:

- **Sandbox:** A sandbox, not necessarily in the pipeline for DevSecOps, is an isolated environment to prevent any possible damage to other environments. It is used for early adversarial cyber-testing and may be used for experimentation.
- **Development (Dev):** The Dev environment is for the development of code and is the environment where iterative development teams normally operate. Test teams, as part of the development team, use the Dev environment to test software units.
- **Integration:** The integration environment is where the software units from multiple development teams come together for testing at a higher level (e.g., features or capabilities). While the software developer may own the integration environment, it is a good place for government test teams to observe and collect some test data that may reduce the need for repeated testing later.
- **Test:** Sometimes called “quality assurance (QA),” the test environment is where developmental and integrated testing is conducted at the system or system-of-systems level. It is normally the last opportunity for developmental testing (except for possible User Acceptance Testing) prior to release to the operations team. The software that comes out of the test environment bears the mark of quality from the development and testing starting at the development team level.
    - The Test Environment should represent the production environment as closely as possible, including monitoring capabilities and the ability to simulate realistic system usage. It might not, however, connect to production environments of interfacing systems within the system of systems.
    - The Test Environment may instead connect to test environments of interfacing systems. Program offices should plan early to coordinate access to interfacing system test environments. These interfacing systems provide the basis for initial interoperability testing. When testing within these environments, it is important to not only test the transmission and receipt of messages, but also the effect of these

messages on the interfacing system. If interfacing test environments are not available, it is incumbent on the program to obtain or develop models or simulations of the interfacing systems and incorporate them into their test environment.
    - Enterprise, artificial intelligence, and machine learning capabilities all rely on ingesting data from multiple data sources, and data integration efforts should run parallel with the software development to make sure the data is in useful form when the software is ready. The data integration should begin in the development of each iteration, but needs to be demonstrated in comprehensive DT before OT.
    - M&S may be employed to represent a production environment that is difficult to replicate or does not currently exist, such as hardware platforms with long build times, data feeds, and interfacing systems. Some systems (such as weapons systems) will need extensive M&S to properly simulate mission conditions.

- **Pre-production (Pre-Prod):** Sometimes called "staging" or "soaking," this is the environment for user acceptance testing, a testing event that verifies the operation of the software in a production-representative environment, including representative cyber threats, prior to full release.
- **Production (Prod):** The production environment, used for live operations with real operators, is often the environment for formal operational testing, and supports the acceptance of the software by the government and the "go live" decision to shift the new software to live operations.

T&E success during the Execution Phase depends on the PM, working with the T&E WIPT, identifying the environments necessary to execute testing for the evaluation focus areas during the Planning Phase and establishing them to the extent feasible. The ability of T&E to remain involved and responsive to the anticipated cadence of software development using continuous integration and delivery starts with testing in operationally representative environments. This is applicable for both the applications and embedded paths.

The environments used to conduct testing for OT&E should represent the production environment as closely as possible, including monitoring capabilities and realistic system use. This requires a high-fidelity representation of the interfacing systems that form the system of systems with the program of record.

The OTA should VV&A the pre-production environments and tools planned for OT&E use before the program enters execution to support the software development cadence. This VV&A should consider data collection, interfacing systems and databases, networks, simulated environments, simulated users, and ranges.

2.3.2 Test Tools

2.3.2.1 Test Automation

Software testing, both functional and cyber testing, should be automated as much as possible to support continuous integration and delivery. The scale of software and the associated testing is

too large for only manual testing, and activities such as regression testing[41] and testing of the routine and repeated human interfaces can benefit greatly from automation. Continuous development, integration, and delivery of software cannot be accomplished without automated T&E.

Automated test tools fall into two major categories:

- **Test management tools** automate the process of test planning, scheduling, tracking, and reporting test events.
- **Test execution tools** automate the process of executing test cases or procedures on the system under test.

The Test Lead should work with the contractor to fully understand the contractor's tools and ensure tools that support OT&E are independently VV&A'd for use. Government test teams should be trained with these tools so they can use their outputs across the software development process to inform evaluations.

Frequently, there will be automated tools supporting multiple phases in the development pipeline, and interoperability among these tools can become a problem. Using known frameworks for pipelines and software factories, as discussed earlier, can help overcome these issues, as it should be inherent in the infrastructure, though testers will still need to identify the data mapping from the automated tools to the evaluation areas.

Automated testing is for government as well as contractor test teams, and using the same tools as the contractor is advantageous for the government (e.g., easier to replicate events when necessary). In some cases, government test teams should become experts in the tools used by both the contractor and other government teams. The automated tools should also provide visibility into the continuous testing occurring within the pipelines so that stakeholders can gain confidence on the quality of the development process.

2.3.2.2 Tools for Data Collection and Reduction

The test teams should first identify the measures to evaluate the system, as well as the data needed and conditions under which it will be collected. These conditions should include injecting operationally representative input values and providing simulated environments to emulate the outcome of the given injects. Additionally, capturing of user interaction with the system should be automated to the extent practical. Having identified the data needs, they should then identify the tools necessary to produce the identified test conditions and collect, reduce, and analyze the data. This should include an evaluation of currently available options and existing system software eco-systems and infrastructure. The needed tools should be integrated into the software pipeline to provide the necessary data. A tabletop exercise can assist in confirming the feasibility of the proposed plans, tools, and methodology.

The test teams should work with the PM as applicable to ensure these tools are available and resourced. The use of these tools should be included in the T&E Strategy.

[41] Regression testing is re-running functional and non-functional tests to ensure that previously developed and tested software still performs after a change.

2.3.3 Test Data: Shared Body of Evidence and Data Repository

During the Planning Phase, the PM should establish a secure data repository to store test data and provide access to all test teams so that they can review, use, and input these test data. Throughout the software development, T&E should be building a shared body of test evidence to support technical, functional, and operational performance evaluations. Relevant test data gathered through all testing should be included in this test data repository. The OTA should maintain the authoritative data for OT&E.

3. T&E During the Execution Phase

Following the Planning Phase, the program will enter the Execution Phase, the purpose of which is to rapidly and iteratively design, develop, integrate, test, deliver, operate, and monitor resilient and reliable software capabilities that meet users' priority needs.[42] The Execution Phase comprises a series of iterations of "plan, code, build, test" to develop software that meets users' needs. As a cyclic and iterative development, it is important to have both DT&E and OT&E test teams involved in testing throughout the Execution Phase to support their independent evaluations.

The Software Acquisition Pathway delivers software in small increments at a prescribed cadence, and T&E should be integrated with that cadence. The result is testing that is continuous throughout the product's lifecycle, with several types of test conducted during the delivery cadence. At times, the program may hold increments from deployment to be combined with others and deployed as a larger release, which may require discrete testing. Test teams should plan for both continual and discrete testing.

Testing should be scheduled based on the product roadmap. Details of this roadmap and its use for T&E are detailed in Section 3.1.

The testing during the Execution Phase can be divided into two areas: testing throughout the development and testing of individual releases. These both support independent government evaluations. Sections 3.2 and 3.3 describe the two areas of testing, respectively. Section 3.4 further describes how data collected from monitoring fielded software can support evaluations.

Lastly, the annual value assessments should be informed by test and evaluation results. Section 3.5 describes how T&E may work with the Program Office to collect data that supports these assessments. The value assessment does not replace operational testing, and the of which is addressed in Section 3.3.

3.1 Product Roadmap

The product roadmap is derived from the Capability Needs Statement and breaks down the required capabilities into epics and features[43]. The product roadmap is "a high-level visual summary that maps out the vision and direction of product offerings over time. It describes the

[42] DoDI 5000.87, p16

[43] An epic is a large body of work to be completed during development. Epics are further decomposed into smaller features and user stories. Refer to the Agile 101 document for additional information: https://www.dau.edu/cop/it/DAU%20Sponsored%20Documents/Agile%20101%20v1.0.pdf

goals and features of each software iteration and increment."[44] An iteration typically refers to a unit of time, whereas an increment refers to a unit of software.

"Programs use the product roadmap to communicate when capability is projected to be delivered. A product roadmap provides a rolling calendar-based view of key capabilities/feature sets to be delivered in the near term (10–12 weeks) through the coming 12–18 months for a product/service, and a high-level description of capabilities to be delivered annually. The roadmap is considered a product schedule."[45]

As a product schedule, the roadmap assists the testers in identifying what epics and features will be developed and tested over time, and thereby influences the detailed test planning and schedule. Figure 4 is a notional roadmap showing how epic and feature development results in capability delivery over time. Though not shown in this notional figure, product roadmaps should define a time period for each iteration. Note that, as with other iterative development plans, it is flexible and subject to change to meet users' emerging requirements and priorities. Test teams should be aware of this flexibility and be prepared to respond as needed. For operational testing, this may include revising risk assessments based on capabilities planned for delivery compared to the capabilities needed. PMs should communicate changes to test teams as they are made to enable adequate response.

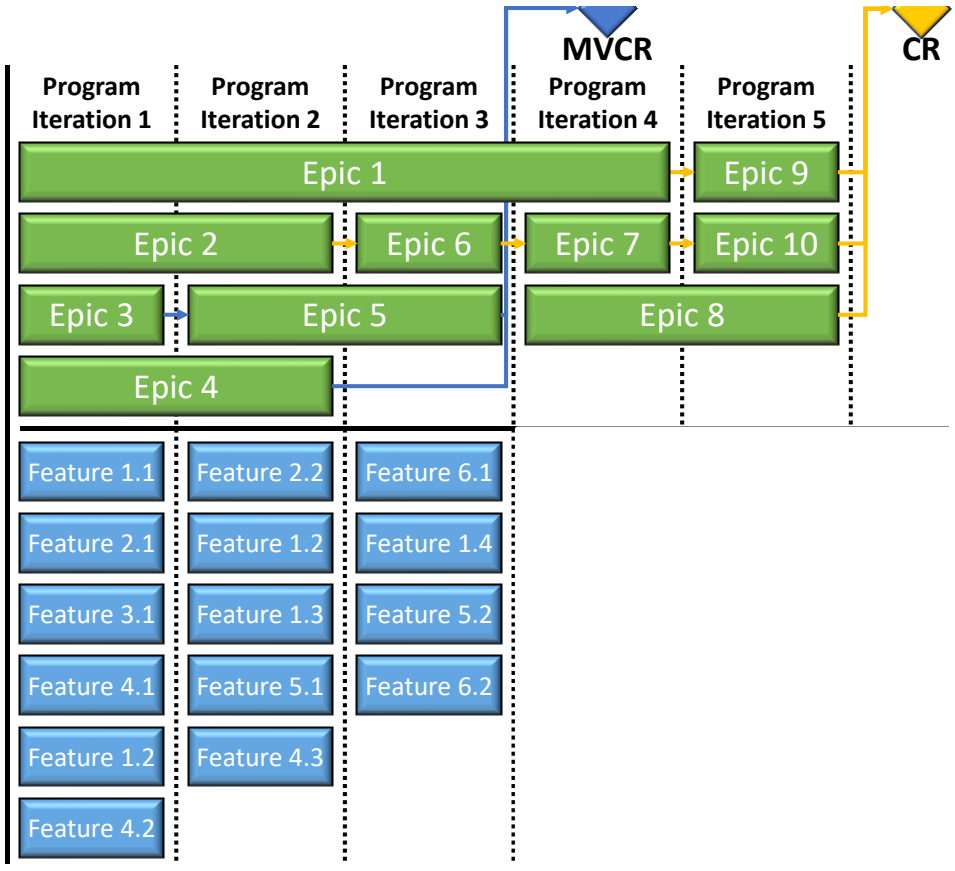


Figure 4. Notional Roadmap[46]

[44] DoDI 5000.87, Glossary

[45] DAU AAF Software Pathway Define Capability Needs

Other things to note from Figure 4 are:

- Capability Releases (CRs)[47], including the Minimum Viable Capability Release (MVCR), are prime candidates for independent government T&E (including OT&E). Refer to Section 3.3 for additional information about scoping capability release testing.
- Not every program increment needs to be deployed to the users.
- Completed epics may not coincide with the next capability release (e.g., Epics 2 and 6 in Figure 4 going to the next CR after the MVCR).

3.2 T&E throughout Iterative Software Development

Government test teams should participate in the iterative development process to review and accept testing conducted iteratively as sufficient, in order to reduce the scope of future government testing. To facilitate this, program offices should ensure government test teams have visibility into contractor testing activities and, results, and complete access to the issue tracking system.

The goal of observing and participating in the planning and demonstration is to capture test data to build a shared body of evidence that can be used as part of the government evaluation to verify whether detailed requirements at the story and feature level are satisfied. Educating the developer on test practices/techniques can be a good practice for improving quality. The intent is to incorporate test cases and scenarios of interest to the government early in the development process and thereby avoid having to re-test these requirements as a subsequent government test. Manual penetration testing and interactive application security testing at the end of each sprint may include misuse and abuse testing to ensure system resilience and cyber survivability.

The T&E community should develop and tailor evaluation metrics for each capability release, then build a data collection, analysis, and reduction plan. While each development iteration may not lead to a capability release, data from each development iteration should support evaluation of the capability release. Ideally, the plan for the capability release will use already existing testing plans and frameworks, tailored to current needs. The T&E community should provide an assessment or evaluation to decision authorities to contribute to decisions and the shared body of evidence.

Within and across capability releases, epics and their features should trace to the identified user capability needs. The T&E community should understand and confirm the traceability among the epics, features, and user capability needs. The T&E community will collect information on how they fit within the larger system of systems for the program. Finally, the T&E community should observe the testing and approval of the features to understand the context of the test environment.

---

[46] Adapted from DAU Course ACQ 1700: Agile for DoD Acquisition Team Members

[47] This document uses the term Capability Release to refer to software to be delivered to users for operational use.

The T&E community will need to understand and participate in the process for developing user stories[48] from the features and validating that the user stories reflect user expectations. Confirming the process for user story traceability to capability needs and decomposition into software tasks informs the evaluation and understanding of test results. The T&E community should review processes, understand the traceability, and assist where needed. A key consideration when reviewing the user stories is to ensure inclusion of all relevant user personas present in the deployed operational environment.

In iterative software development during the Execution Phase, testing is a part of the continuous process that requires integration between testing and development and users to achieve high product quality. Test teams should be involved up front to ensure they get the data they need from the process.

A primary example of early test involvement is Test Driven Development (TDD), in which Dev Team testers develop test scripts, derived from the user stories, that provide details of what the software should do to be declared "done." Dev Teams then develop software to pass the test scripts. Some programs have experimented by placing government testers in these roles to better understand the software development process and participate in early verification. In these cases, TDD places a lot of responsibility on government testers to work closely with the capability owner, learn how to write and execute test scripts, and define and declare the "definition of done." If resources are limited, government testers may not be able to be embedded in the Dev Teams, but they should still understand how to write and read automated test scripts to monitor and collect test data from vendor or government Dev Team testing.

Two variations of TDD include Behavior Driven Development (BDD) and Acceptance Test Driven Development (ATDD). This is where government testers should have significant involvement. BDD looks at a class of user stories (e.g., a scenario) and tests to "the specifications of the behavior of the class" that produces an outcome valuable to a user.[49] Rather than using the "as a role-I-want-so-that" format of a user story, BDD uses a "given-when-then" format.

GIVEN: The preconditions of the test (e.g., my system is connected to all necessary external sources).

WHEN: An action is taken (e.g., I request a status of friendly forces).

THEN: The following results should occur (e.g., the location and status of friendly forces are displayed).

Note, it is also possible to add "**AND**" statements to better define the behavior (e.g., AND I specify the information I need).

ATDD derives from both TDD and BDD, but at a higher level, looking at the overall customer experience. According to a Net Solutions blog, TDD asks "are we building the thing right," BDD asks "if the thing is behaving as expected," and ATDD asks "are we building the right thing."[50]

---

[48] A user story is the smallest unit of requirements written from a user's perspective of how they will use the software. Refer to the Agile 101 document for additional information: https://www.dau.edu/cop/it/DAU%20Sponsored%20Documents/Agile%20101%20v1.0.pdf

[49] agilealliance.org/glossary/bdd

[50] Net Solutions blog

ATDD often uses the same given-when-then format of BDD, but at a higher level. Table 3 provides a comparison of TDD, BDD, and ATDD. As noted above, government testers should focus on BDD and ATDD.

Table 3. Comparison of TDD, BDD, and ATDD[51]

Parameters	TDD	BDD	ATDD
Definition	A development technique focused on individual units of a desired feature	A development technique focused on expected behavior	A development technique focused on meeting the needs of the user
Participants	Developer	Developers, Customer, testers	Developers, Customers, testers
Language Used	Written in programming language used for feature development (e.g., Java, Python, etc.)	Gherkin / Simple English	Gherkin / Simple English
Understanding Tests	Tests written by and for developers	Tests written for anyone to understand	Tests written for anyone to understand
Focus	Unit Tests	Understanding Requirements	Writing Acceptance Tests
Bugs	Reduced likelihood, easier to track down	Can be more difficult to track compared to TDD	Can be more difficult to track compared to TDD
Suitable For	Projects that do not involve end users (server, API, etc.)	Projects which are driven by user actions.	Projects where customer experiences are important and competition is high
Tools Used	JDave, Cucumber, JBehave, Spec Flow, BeanSpec, Gherkin Concordian, FitNesse, Junit, TestNG, NUnit frameworks, Selenium tool (any open source tools)	Gherkin, Dave, Cucumber, RSpec, Behat, Lettuce, JBehave, Specflow, BeanSpec, Concordian, MSpec, Cucumber with Selenium / Serenity	TestNG, FitNesse, EasyB, Spectacular, Concordian, Thucydides, Robot Framework, FIT

[51] Ibid.

Figure 5 and Table 4 summarize the different testing types during software development that may occur during the Execution Phase. These are each detailed further in Sections 3.2.1 through 3.2.6.

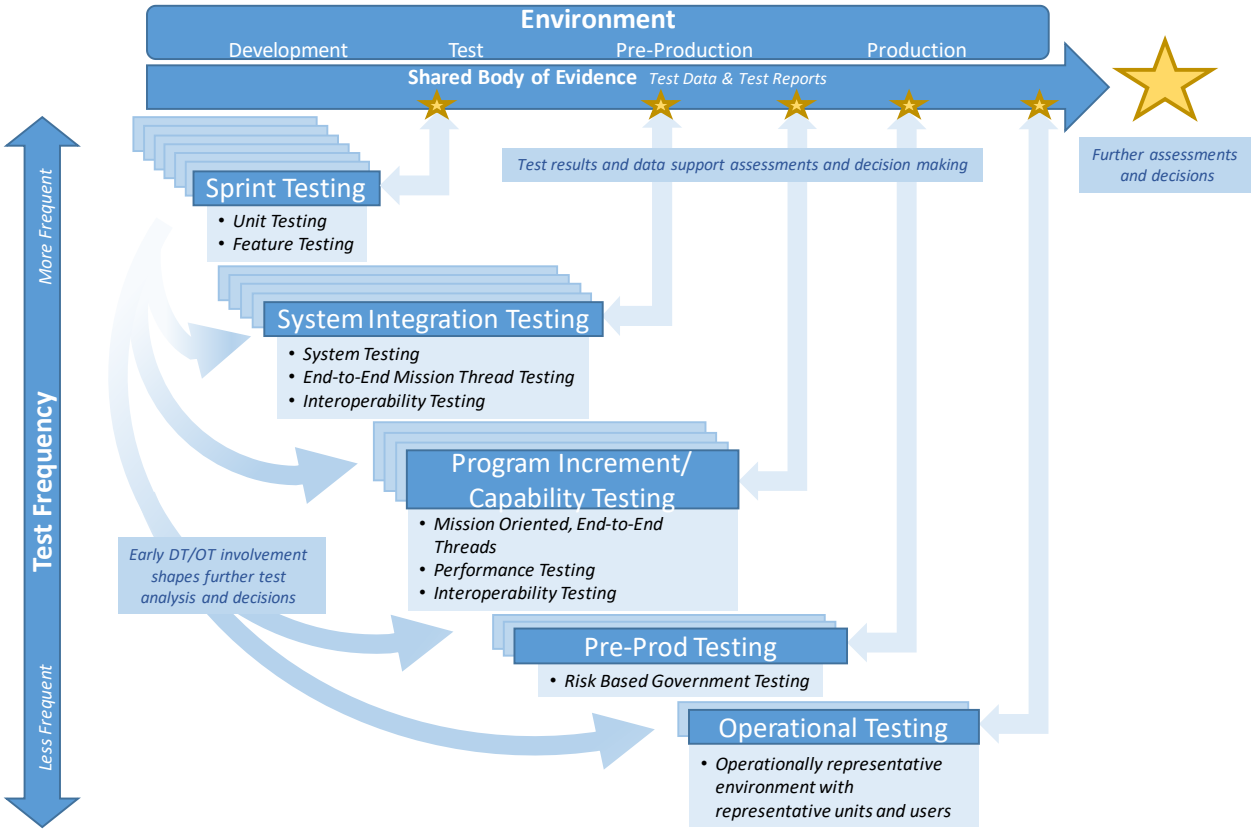


Figure 5. Continuum of Test Throughout the Development Lifecycle

Table 4. Summary of Testing Types During Iterative Development

Testing Type	Summary of T&E Guidance	
Agile/Sprint	• Developer runs unit tests and conducts demonstrations at the end of each sprint. OT Team can conduct user surveys during demonstrations to support early suitability assessments.	Cybersecurity Testing [a]
System Integration	• End-to-end mission thread testing (including cyber and interoperability testing). • Requires a secure test environment that closely resembles the production environment, a comprehensive build of the software, data that exercises the connections inside and outside the application, and a test plan.	
Program Increment or Capability Release	• Government-led DT event to verify that the system capability is ready for release to the operational user. • DT team should coordinate this testing with OT and interoperability (e.g., JITC) test teams to facilitate early collection of test data for independent OT&E and certification.	
Pre-production	• Operations team conducts testing to resolve any potential problems because of differences between the development environment(s) and production environment(s). • Lead DT&E Organization or OTA can conduct risk-based government DT or OT on the pre-production environment, and government cyber testers test with less risk of affecting actual operations.	
Operational (e.g., Operational Assessment, Initial Operational Test, Limited User Test)	• OTA conducts testing to evaluate the operational effectiveness, suitability, and survivability (including cyber) of the system, or progress toward, in an operationally representative environment with representative system users and units equipped with the system • OT&E should utilize data from contractor and developmental testing for system functions and focus on the software's ability to support end-to-end mission(s).	

[a] Cybersecurity testing should evaluate the system throughout development and all test phases to determine cyber posture and include in independent, government events to support evaluation

3.2.1 Agile /Sprint Testing

The developer runs unit tests upon implementing a user story, using pre-written test scripts to verify success or failure. The test scripts automate the testing and provide a repeatable process to verify software performance. Ideally, user stories combine into features that provide the user a better perspective of the operational value of the software. Just as the user stories are integrated to form features, the test scripts are integrated to automatically test software at this higher level.

This automated union of user story testing does not necessarily imply that the feature testing is complete. Early cyber testing at this level should include at a minimum static and dynamic analysis to identify known vulnerabilities.

The development team conducts demonstrations at the end of each sprint to show users how all the software units work together and to provide hands-on experience and gain user feedback. Actual user participation is important to get feedback, early acceptance, and buy in. The OT team can conduct user surveys during these demonstrations to support early suitability assessments.

3.2.2 System Integration Testing

Integration testing brings together the individual efforts and outputs of multiple development teams to test at the system level with end-to-end mission thread testing (including cyber and interoperability testing). This testing requires a secure test environment that closely resembles the target (or production) environment, a comprehensive build of the software, data that exercises the connections inside and outside the application, and a test plan (including test cases developed by the contractor and the government). The system integration testing should also include testing representative interfaces with external systems using representative data.

3.2.3 Program Increment or Capability Release Testing

The release test is a government-led DT event to verify that the system capability is ready for release to the operational user. Coordinating this testing with OT and interoperability test teams (as applicable) is encouraged to facilitate early collection of test data for their independent evaluation or certification. Capability Release testing is a key activity to support the decision authority in making informed release decisions. The Capability Release test should focus on mission-oriented DT with end-to-end mission threads and actual users. Testing should also include cyber DT, performance/load testing, and interoperability testing.

3.2.4 Pre-production Testing

The operations team[52] conducts testing to resolve any problems that might be caused by a difference in configurations between the development or test environment and the pre-production or production environment. Ideally, there should be no differences, but this may not always be the case. Testing on the pre-production environment is also where the Lead DT&E Organization or OTA can conduct risk-based government DT/OT and the government cyber tester can conduct testing with less risk of impacting actual operations.

3.2.5 Operational Testing

Operational testing and evaluation should be conducted to support MVP, MVCR, and subsequent capability releases.

These operational evaluations should utilize data from contractor and developmental testing for system functions (where feasible) and focus on the ability for the software to support the end-to-end mission of the users. Data for operational evaluations should include use by operators and

[52] The operations team includes system administrators, database managers, network managers, and cyber defenders.

units equipped with the system who were not involved in the development testing and software definition to evaluate whether the system will meet the needs of all users.

Software releases that are initial or greatly change capability require user- and unit-level training to develop tactics, techniques, and procedures for use. Often such training periods will include a culminating exercise or activity to ensure that the capability release is ready for operational use. Such activities present an opportunity for collecting data for OT&E.

If the culminating activity is conducted in a pre-production environment, that environment should be validated for operational representativeness and the OTA should note any limitations of the environment. This may include environments already developed for other testing within the DoD. For example, within the embedded path, these may be test environments, including digital representation of hardware, for the platform on which software will reside.

For programs on DOT&E oversight, DOT&E will approve the operational test plan(s) that describes how the OTA will execute operational testing, and which data will be used from the accumulated shared body of evidence to support the evaluation. An operational test plan should be written early in the execution phase, and referenced and updated as needed to support ongoing testing.

3.2.6 Cyber T&E

Cybersecurity testing should be conducted throughout all development and test phases to evaluate the system, including the software pipelines, and determine its cyber posture. Government testing should include both cooperative and adversarial testing.

The test teams should work with the Cyber Working Group, as a subset of the T&E WIPT, early to ensure a coordinated risk management framework and cyber test and evaluation process. Cyber T&E and software assurance will be integral to strategies, designs, development environments, processes, supply chains, architectures, enterprise services, tests, and operations.

The Cyber Working Group is responsible for designing and implementing automated cyber testing and continuous monitoring of operational software to support a continuous authority to operate (cATO) or an accelerated accreditation process to the maximum extent practicable. Results from Cyber T&E will support the cATO throughout the lifecycle.

Automated cyber testing should be augmented with additional testing where appropriate. Programs will also implement recurring cyber assessments of the development and test environments, processes, and tools.

Secured pipelines may improve software security, but additional steps are still required to verify the system itself is resilient to cyberattack. Software assurance and cyber testing activities within and beyond the software factory serve to evaluate that resilience. In addition, periodic assessment of the software factory components is necessary to assure their continued ability to provide a secure environment for software development.

To ensure secure code through the pipeline in the final application, cyber test teams should assess all aspects of the software pipeline. This includes the trusted development platform, tools, processes, and infrastructure. Cyber test teams should assess whether the operator of the development platform maintains trustworthiness through periodic assessments, implementing a

cyber threat intelligence program, regularly installing the latest security updates, and the use of an active defense capability that includes continuous monitoring and logging.

Continuous and automated cyber testing can identify vulnerabilities to help ensure software resilience in the evolving threat environment throughout each sprint and the entire lifecycle. Ensuring software security includes:

- Secure development (e.g., development environment, vetted personnel, coding, test, identity and access management, and supply chain risk management)
- Cyber and software assurance capabilities (e.g., software updates and patching, encryption, runtime monitoring, and logging)
- Secure lifecycle management (e.g., vulnerability management, rigorous and persistent cyber testing, and configuration control)

Testers should evaluate whether the software pipeline provides capabilities to enable iterative development to reduce the burden of full software stack testing and security. Test teams should evaluate, through automated and manual assessment methods, whether all platform, infrastructure, and application security requirements implemented by the development team or inherited by supporting services provide cyber resilience.

Cyber testing should also characterize the cybersecurity defensive status of a system. This includes evaluating the system with the cyber defense team in place.

Program offices adopting iterative development processes to develop and deliver code should incorporate the additional software assurance activities in the Cyber T&E Focus Area and Cyber T&E Companion Guide. The Cyber T&E Focus Area describes cooperative and adversarial cybersecurity testing throughout the lifecycle. The guidebook also offers insight and instruction for performing test activities to evaluate the security of the acquisition program.

3.3 Scoping T&E for MVP, MVCR, and Follow-on Capability Releases

While T&E is conducted throughout development, individual releases should be tested and evaluated as a whole to ensure they are meeting user needs, and are operationally effective, suitable, and survivable. The scope of independent government testing for each release should be determined using a risk-informed strategy.

3.3.1 T&E of the MVP

Government testers should assist the PM with test planning, execution, and data collection and with obtaining feedback from the users. Data collected during an MVP evaluation might be used later for an MVCR evaluation to determine readiness for operational deployment.

The MVP version of the software could become the MVCR if the sponsor determines it is sufficient to be fielded for operational use. In that case, the scope of T&E should increase so as to determine operational effectiveness, suitability, survivability (including cyber), and the risk of mission failure or personal injury in the event the MVP is defective in any manner. Refer to Section 3.3.2.

The scope of the MVP testing is guided by the specific capabilities available and the feedback that the PM and sponsor want to address. This may be limited to user surveys or it might include technical performance testing to help change the design. Since the MVP is essentially a

developmental evaluation, the PM or government developmental testers are prime candidates to lead any MVP testing. To maximize opportunities for integrated testing of the MVP, the PM or government developmental testers should coordinate with the OT&E community for this testing.

As the first version of the software exposed to users, the MVP presents the opportunity for early operational evaluation to assess progress toward operational effectiveness, suitability, and survivability. The OTA should evaluate the MVP in the context of the operational mission(s) the software will support and assess progress toward operational effectiveness, suitability, and survivability. The OTA should incorporate data from the shared body of evidence to support the evaluation.

If the data to support the evaluation will be generated in a virtual environment, the environment should go through VV&A as appropriate to support conclusions. OTAs should indicate any limitations for testing in a virtual environment in the assessment plan.

3.3.1.1 Cyber T&E for MVP

The scope of the cyber T&E of the MVP should be determined based on the maturity of the MVP and the representativeness of its attack surfaces' environment.

If the MVP is mature enough and the assessment is conducted in a quasi-production environment with attack surfaces similar to the production environment, then cybersecurity developmental test events, such as cooperative cyber assessments or adversarial cyber assessments, may be conducted.

Operational cybersecurity testing should also be conducted, as appropriate. At a minimum, the OTA should be gathering metrics and data from cybersecurity testing conducted within the development pipeline.

3.3.2 T&E of the MVCR

As the first capability fielded to support operational missions, the OTA should conduct an Initial OT&E for the MVCR to evaluate its operational effectiveness, suitability, and survivability (including cyber). The OTA should draw data from the shared body of evidence to support the evaluation and the scope the IOT&E. Data gathered during the IOT&E adds to the shared body of evidence supporting the system. DOT&E will independently report on testing of the MVCR for systems on DOT&E oversight.

3.3.2.1 Cyber T&E for MVCR

OT&E of the MVCR should include cooperative and adversarial cyber operational testing. Further details on conducting this testing is included in the Cyber T&E Focus Area and Cyber T&E Companion Guide. The cyber testing of the MVCR may include testing of the software pipeline itself, especially if the pre-production environments of the pipeline are directly connected to the fielded, production environment. This is part of the supply chain assessment.

For programs using the Embedded Pathway, testing the MVCR should be aligned with the IOT&E or other applicable OT&E for the platform on which the software resides.

3.3.3 Risk Informed OT&E for Follow-on Capability Releases after MVCR

OT&E of capability releases should be tailored using a risk-informed strategy. The MVCR testing provides a baseline for testing of future capability releases. Subsequent releases may require less dedicated OT&E based on the risk to mission of the new release being fielded (e.g., complexity of the release, amount of new capability and features included, number of new users involved). The OTA should determine the inclusion of previously tested capabilities in testing based on interactions with new capabilities added and the risk to the mission should they fail as part of the risk assessment.

Programs entering the Software Acquisition Pathway with a capability comparable to an MVCR should follow the risk-informed approach described for capability releases. If operational testing has not yet been conducted, a dedicated OT&E event may be needed to baseline the capabilities and support risk assessments for scoping of future testing.

OTAs should follow the latest DOT&E and Service guidance on conducting risk assessments to determine the level of operational testing. For programs on DOT&E oversight, DOT&E approves the operational test plan.

3.4 T&E Post-Release (Monitoring and Feedback)

To continually evaluate the system, the PMs should provide testers with data from monitoring and feedback of the production system once fielded. Examples of data sources that testers should be provided are:

- System uptime, downtime, and time to repair fixes (e.g., system logs)
- Error reports for specific node hardware, services, and applications
- Help Desk problem reports and their associated closure information
- Cybersecurity monitoring information

Testers should use these data to support ongoing, independent assessments. Monitoring data should be incorporated into the shared body of evidence, as applicable.

Operational testers should use these monitoring data to support independent evaluations of the systems. Use of monitoring data to support operational evaluations should be described in the T&E Strategy, as described in Section 2.1.1. Periodic assessment by operational test teams of the fielded baseline provides objective determination of capability improvement and continued security.

In addition to providing information on the suitability and supportability of the system, the user feedback should inform scoping of future independent testing.

3.5 T&E to Support Value Assessments

The sponsor and user community perform the value assessment annually, which assesses mission improvements and efficiencies realized from the delivered software capabilities, and determines whether the outcomes have been worth the investment.[53] The Value Assessment does not require

[53] DODI 5000.87, p23

separate T&E events, but may use data from T&E to support the assessment. How testing supports the value assessment should be included in the overall T&E Strategy.

The primary concern from the test perspective is: How does the program define "value," and how is it measured? The value assessment satisfies the requirement for a Post-Implementation review (PIR) for an IT system described in DoDI 5000.82, which states that the PIR will "report the degree to which doctrine, organization, training, materiel, leadership, education, personnel, facilities, and policy changes have achieved the established measures of effectiveness for the desired capability.[54]"

The Functional Sponsor should work with the PM to define "value" during the planning phase. Ideally, the definition of "value" and "measures of effectiveness[55]" should be included in the CNS. The USD(A&S) guidance supporting 5000.87 suggests the following examples:

- Increase in mission effectiveness
- Cost efficiencies
- User workload reduction
- User personnel reduction
- Equipment footprint reduction
- User adoption and user satisfaction.

If done properly, a value assessment requires capturing value data as a baseline before the implementation of the software to make the comparison post-implementation. This baseline data capture should be done on the legacy system (if one exists) before the development of the software system, as testers will be involved with testing the new system during development.

The OTA should work with the sponsor and user community to determine whether data they need to conduct the value assessment will need to be collected during operational testing, particularly in assessing the mission improvements and efficiencies realized. The OTA should incorporate collection of these data during OT&E events, as applicable. The OTA, sponsor, and user community should review the data collection needs at least annually to support the upcoming year's value assessment.

[54] DoDI 5000.82, page 8

[55] Note that these measures of effectiveness may not necessarily be the same as those developed by the OTA for OT&E. Value assessment measures of effectiveness may be more business related (e.g., cost reduction) than performance or mission effectiveness.