DEFENSE INFORMATION SYSTEMS AGENCY

JOINT INTEROPERABILITY TEST COMMAND
FORT HUACHUCA, ARIZONA

JITC GUIDEBOOK FOR DEVELOPMENT, SECURITY, AND OPERATIONS TEST AND EVALUATION

VERSION 1.0
JULY 2021

JITC GUIDEBOOK
FOR
DEVELOPMENT, SECURITY, AND OPERATIONS
TEST AND EVALUATION

VERSION 1.0
JULY 2021

Submitted by:

Lorie Sather
DevSecOps Service Area Lead

Richard Delgado Jr.
JITC Technical Advisor

Approved by:

MATTHIAS.ROBERT.DENNIS.1148250770

Digitally signed by MATTHIAS.ROBERT.DENNIS.1148250770
Date: 2021.08.06 15:18:17 -07'00'

ROBERT D. MATTHIAS
CAPTAIN, USN
Commander

Prepared Under the Direction of:

Ellen Preiss
Joint Interoperability Test Command
Fort Huachuca, Arizona

INTENTIONALLY BLANK

TABLE OF CONTENTS

Page

APPENDICES

LIST OF FIGURES

LIST OF TABLES

INTENTIONALLY BLANK

1. INTRODUCTION

The Development, Security, and Operations (DevSecOps, or DSO) process life cycle emphasizes fast delivery of functionality to stakeholders in smaller iterations to maximize the actualization of value and to reduce the complexity, size, and risk associated with each release. This causes an emphasis on reducing feedback loops through early detection, automation, continuous delivery pipelines, and self-service deployments from fully empowered and cross-functionally integrated teams. These integrated teams represent the development team and the operational team to ensure cybersecurity is an integral focus throughout. To support DSO, the Joint Interoperability Test Command (JITC) Action Officers (AOs) need to adapt testing methodologies and Standard Operating Procedures to accommodate the swift pace and continuous verification and validation of capabilities, compliance, and security.

The goal of DSO is to eliminate bottlenecks and optimize the value stream of delivery to facilitate shorter feedback cycles, which then facilitate even quicker delivery in a continuously improving cycle. DSO Program Management Offices (PMOs) pursue improving the time-to-delivery, reducing the size of delivery, and reducing the time to feedback for all features being developed in the system. Efficiency is the core defining feature of a DSO program. It will become imperative for AOs to learn the identifiable traits and culture of DSO. JITC's role as an independent test and evaluation (T&E) agency is to ensure that joint warfighting information technology capabilities are effective, suitable, interoperable, and secure while supporting mission needs. The JITC AO will need to adapt to the fast-paced environment of DSO and to support DSO programs in an integrated fashion while maintaining objectivity of mission.

Hint: Agile and DSO are not the same thing! Agile produces relatively larger and slower releases as compared to DSO. Whereas Agile releases are packaged as quickly as each week, DSO could be triggered daily (or hourly). The size of the release package is a major difference. DSO releases have been called “micro-releases” because they can often be as small as a single change. **Be careful not to conflate them!** Understanding the differences will enable the AO to determine the most appropriate support methodology, given the PMO’s specific lifecycle approach.

This guidebook provides JITC AOs guidance on conducting T&E of programs that are implementing a DSO software development approach within their acquisition pathway.

1.1 Purpose

The purpose of this T&E guidebook is to aid the JITC AOs and test teams in supporting DSO programs to objectively verify and validate compliance, security, and design testing methodologies. AOs will provide DSO T&E subject matter expert (SME) advice early and often to the PMO. This may include testing methodology design and validation, periodic testing methodology audits, and/or embedded testing support to the DSO cross-functional teams.

While Agile and DSO are not synonymous, this guidebook provides concepts and processes that are applicable to many Agile methodologies. Generally, the speed of Agile is slower than the speed of DSO, so the processes that are capable of supporting DSO can also support Agile.

1.2 Guidebook Organization

This guidebook contains sections that describe the general processes and steps applicable to DSO, and the JITC-specific application of DSO in test strategy development and test execution:

- Section 1: Introduction
- Section 2: High-Level AO Actions/Steps
  - Step 1: DSO Primer and Training
  - Step 2: Test Methodology Identification
  - Step 3: Engagement Agreement and Terms of Service
  - Step 4: Integrated Test and Evaluation
  - Step 5: Engagement Wrap-up/Closing
- Section 3: JITC Test Roles

The supplemental information in the appendices includes:

- Appendix D: Identifying DSO (Checklist)
- Appendix E: Early Test Involvement (ETI) and T&E Scorecard (for Defense Information Systems Agency (DISA) Projects)
- Appendix F: PMO Roles and Responsibilities
- Appendix G: Tools and Activities
- Appendix H: DSO Integrated Test Strategy Development
- Appendix I: Environments and Additional Tools

2. HIGH-LEVEL ACTION OFFICER ACTIONS/STEPS

2.1 DSO Identification

The first action an AO should take for potential DSO (and/or Agile) customers is to evaluate whether the DSO identification itself is true. DSO programs will have a multitude of differences, but there will be core similarities that should drive the AO to adopt DSO techniques. These similarities (traits) are as follows:

- The pace of releases is extremely fast and facilitated through automated processes.
- The PMO is focused on improving two primary Key Performance Indicators (KPIs): lead time (time-to-delivery) and process time (measures each discrete process).
  - Another KPI that is essential to DSO success is the time to feedback. The optimization of feedback, including testing and verification, is fundamental

to increasing the speed of delivery.

- Release packages are small (even tiny) and easily measurable.
  - Risk is measured and quantified more accurately and succinctly, which enables clear lanes of empowerment for deployment of software and configurations with low risk profiles (the majority of changes).
  - Smaller releases are easier to understand, manage, and rollback.
  - Hidden or unknown risks are minimized due to the simplicity of the releases.
- Artificial organizational boundaries are actively eliminated wherever possible in favor of empowered, cross-functional (including security) teams.
  - Development, Security, and Operations are not separate stove-piped teams. The professionals are all on the same team(s) with the same goal(s): Delivery of valuable and reliable software to the stakeholders.
  - Placing systems administration and security on the same team with the developers ensures the system and security impacts are considered throughout the process. This also ensures the developers utilize production-like environments, reducing deployment risk.
  - Shifting security left by certifying processes and teams, rather than each individual release, increases speed for deployment and ensures dependable cyber survivability. The goal is to automate cyber authorization, and continuously monitor cyber operations, for a continuous authorization.
- Time-to-feedback traits:
  - Testing is a primary form of direct feedback.
  - Developers develop automated test scripts to test their deliverables.
  - Developed code and new configurations are automatically tested (immediately triggered) upon check-in, ensuring the fastest possible feedback loop.
  - Developer machines (local or virtual) are configured as close to production configuration as possible. **This includes all security configurations**.
  - Production support feedback is integrated into development planning directly.
  - Operational scenarios are integrated into testing to verify operational performance thresholds are met.

Depending on the maturity of the customer's implementation of DSO, the AO may encounter different levels of adoption of the above traits. However, the focus on increasing speed and minimizing risk through automation and minimal release sizes are fundamental to DSO. Integrated teams that eliminate unnecessary organizational divisions (zones of authority that stifle speed of delivery) are fundamental. That is, a program may be maturing or

Hint: As an AO, it will be important to recognize when programs are DSO, Agile, planning adoption, or performing "cowboy development" (uncontrolled ad-hoc development) and calling it DSO.

This knowledge will help the AO determine which approach to utilize when engaging with the program.

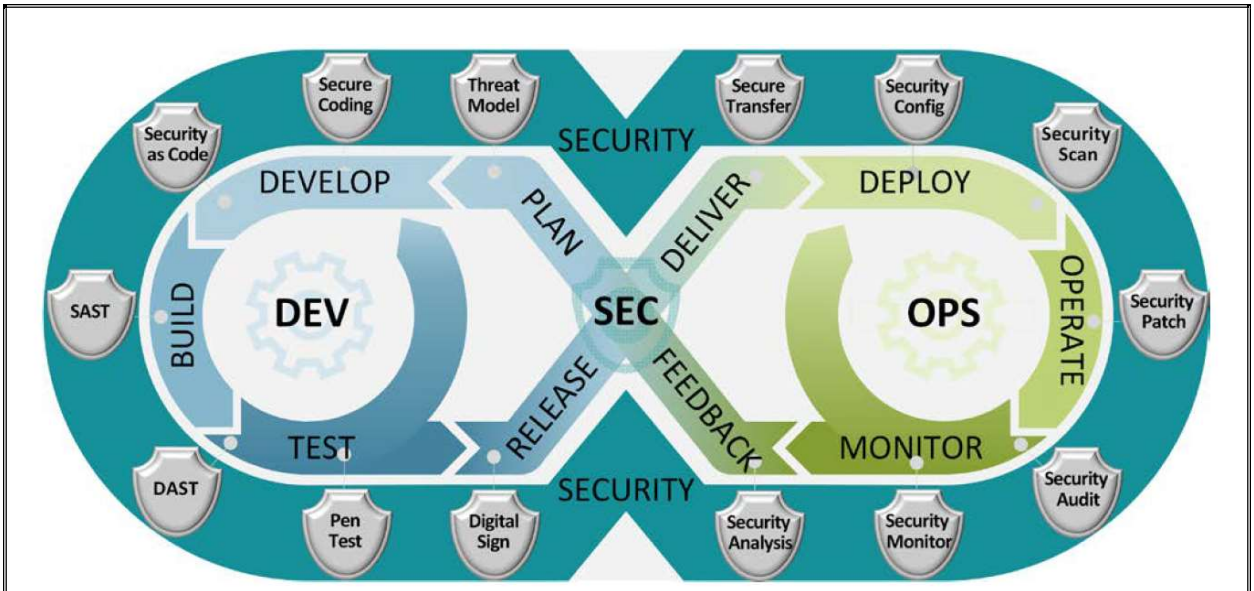
evolving towards DSO while embracing some DSO tenets but they are not DSO unless these traits become true.

As an AO, it will be important to recognize when programs are DSO, Agile, planning adoption of DSO, or performing “cowboy development” (uncontrolled ad-hoc development) and calling it DSO, to determine which approaches will and will not work for the program.

2.2 Step 1: DSO Primer and Training

Development Operations (DevOps), and the more recent DevSecOps, fundamentally change how systems or capabilities are delivered. AOs will not be able to rely on procedures of the past and JITC will not be able to shoehorn old, stage-gate focused processes and policies. Rather, JITC will need to adapt to cross-functionally capable teams and AOs will need to hone their skills in new techniques.

The Department of Defense (DoD) Chief Information Officer’s (CIO’s) “DoD Enterprise DevSecOps Reference Design,” Version 1.0, 12 August 2019, introduces the DSO software development life cycle (SDLC) as shown in Figure 1. It is important for the AO to remember that this is not a one-time development cycle. Each system release will undergo requirements development, testing, and delivery to production and operations continuously. JITC testing will be integrated throughout these cycles, for each release, with the exception of periodic testing being executed in parallel. Figure 2 provides another version of the DSO life cycle, focused on the continuous nature of integration and testing in DSO.



NOTE: Source: DoD CIO, "DoD Enterprise DevSecOps Reference Design," Version 1.0, 12 August 2019

LEGEND:

Config	Configuration	Pen	Penetration
DAST	Dynamic Application Security Test	SAST	Static Application Security Test
DEV	Development	SEC	Security
OPS	Operations		

Figure 1. DevSecOps Software Development Life Cycle

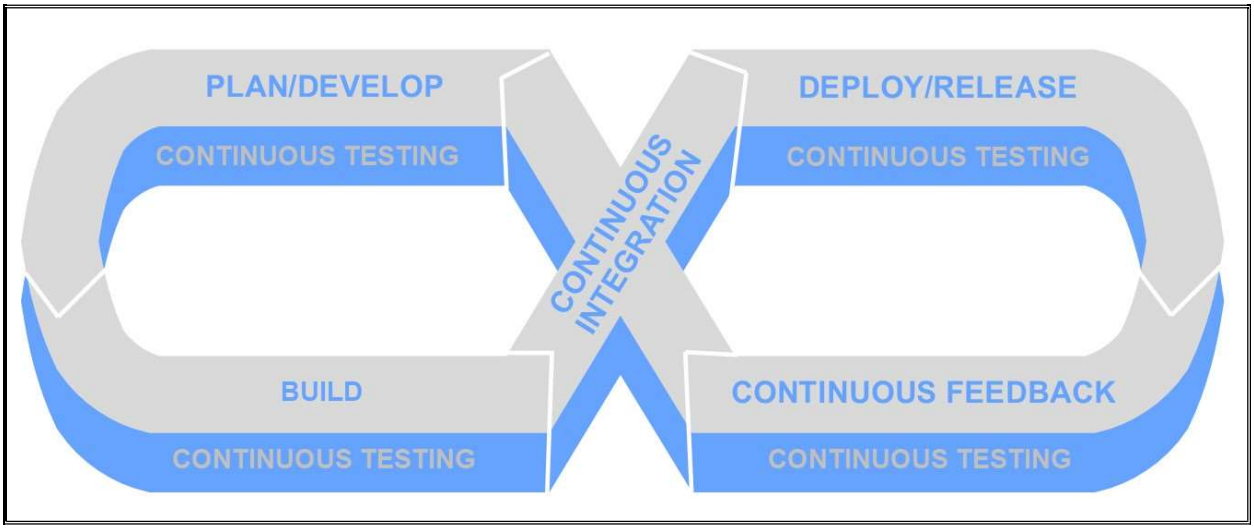


Figure 2. Continuous Integration and Testing

Historical organizational and team structures that create silos between professional skill sets are an impediment to speed. Because of this impediment, DSO revises the structure of systems development teams by incorporating the infrastructure and security professions onto the team focused on enhancing the system (historically called "Development Teams" and now called "DSO Teams"). These teams are

empowered to continuously move new configurations, code, and even architecture changes from the idea/inception stage through the live-in-production stage.

The DSO Teams' empowerment should not be carte-blanche. As a neutral third party, JITC AOs should engage with the PMO to ensure that proper limits have been put into place. These limits should prohibit any changes to the core system security and any policy-impacting changes. For example, a team could be empowered to change a web form so if the change does not impact privacy, system security, and/or compliance.

Limitations on the number of discrete changes per release can also facilitate greater speed of delivery. This is primarily due to the human variable. Humans are not adept at estimating the risk of large changes but are very good at estimating the risk of small ones. Because of this, small changes can be understood and evaluated much more quickly than larger or batch changes. Whereas Waterfall and Agile can push/deploy hundreds of changes at a time, it is not atypical for DSO to deploy (automated) one or two at a time (see analogy in paragraph 2.2.1) but with a greatly increased frequency.

Continuous delivery is an extension of continuous integration and test. This ensures the team can release software changes to production in a quick and sustainable way. Continuous delivery may be facilitated through Infrastructure as Code (IaC) and some controlled deployment methodologies that stem from IaC. IaC enables the team to build and tear down fully encapsulated "containers" that perform the operations/functions of the system without horizontal or vertical dependencies. In essence, this means that all infrastructural, commercial-off-the-shelf (COTS) products and/or development stack requirements are included within the single container being deployed. These requirements are included in the description and configuration of the container itself and are controlled with the same rigor as codebases (versioning/tracking).

Because of the greater flexibility IaC and containerization provides, the AO must be aware of the testing implications it enables. Feature flagging, blue/green deployment (see additional details in Appendix G), and the flexibility to stand up fully functional test environments on the fly (and then tear them down) enable the team(s) to move at a fast pace while keeping bottleneck-causing external testing (security audits/penetration tests for example) parallel to the value stream.

Agile and DSO do share some commonalities. The primary one is the culture of delivery and feedback-based refinement. Both Agile and DSO prioritize getting something of value into the users' hands as quickly as possible so that the users can drive the evolution of the product. This is a cultural shift from the typical top-down management philosophy of traditional systems and software acquisition and enhancement. Both Agile and DSO view testing, verification, and validation as primary types of feedback that help ensure acceptance and quality of products. DSO (and many Agile methodologies) also prioritizes automated testing. This prioritization is primarily to increase the speed of feedback to the developer.

Hint: Automated testing should be pushed as close to the individual developer/engineer as possible (push left). This ensures the developer/engineer still has the context of what they were working on when the test failed. Context switching causes overhead in human brains and that slows velocity. If the program can highlight a failure immediately, while the developer/engineer is still working on it, it has a much higher chance of being fixed as efficiently as possible and without anyone else having to get involved. The same tenet applies to system configurations and security settings. Developer/engineer virtual machines (VMs) and/or computers should mirror production's settings as much as possible at all times.

There are a myriad of tools emerging (see the DSO Adoption spreadsheet referenced in Appendix C) that help enable greater adoption of DSO. See appendices G and I for additional information on tools. As with any methodology, the tools are only as good as the culture and the knowledge of the organization/program. Some of the tools that JITC AOs will encounter are automated testing tools and software, dashboards, test data management, pipeline management software, deployment software, container software, and Agile lifecycle management software. These suites work together to empower the developer, security engineer, administrator, PMO, and testers to function as a cohesive unit while maintaining both lead time and process time (speed of delivery) KPIs. AOs will primarily interact with automated testing and Agile lifecycle management.

Developing and managing automated tests will become a primary skill set for the AO. Automated testing software is capable of performing unit, functional, security (triggered scans), smoke (includes confidence, build verification, acceptance), performance, regression, integration/Application Programming Interface (API) testing and more.

Finally, the emergence of artificial intelligence (AI) into the domain of testing is quickly increasing. Many of the automated tools have been working to incorporate AI into their capabilities for exploratory, user interface (UI), and functional testing. These AI bots are able to mimic user behavior before something is released into the wild and can represent significant risk-coverage capability increases when they are deployed.

2.2.1 Analogy: AgileWidge versus DSOWidge

To help when thinking about the differences between DSO and Agile, imagine two companies that make widgets. The first company: “AgileWidge”; makes weekly batches that it then sends to its distributors and stores. These batches are made and shipped quickly and adapt weekly to customer demand. The stores and distributors are partner companies and/or franchises that act through contractual relationships with AgileWidge.

The second company “DSOWidge” also makes widgets and they directly compete with AgileWidge. The difference is that DSOWidge provides made-to-order widgets that are direct-to-customer. DSOWidge has to incorporate all of the logistics, packaging, marketing, and sales into a single streamlined system in order to meet customer needs and demand ebb-and-flow. Because of customer customization, packaging and delivery are integrated into the flow. To receive and plan the orders, marketing and logistics are integrated into the flow. For customer delivery and satisfaction, tracking the order through assembly and delivery must be integrated. DSOWidge does all of this at speeds that compete with the in-store experience that AgileWidge provides.

There was also another company in the marketplace called “WaterfallWidge”. They planned an entire year’s worth of manufacturing based on sales forecasts and historical models. But when demand spiked or competitors innovated, they were unable to adapt and went out of business.

These examples are meant to illustrate a core difference between the three primary models of system development. Agile and DSO are often conflated but they are two different things. Whereas Agile is much faster at delivering value than Waterfall and it is more flexible than Waterfall, DSO is that much faster and more flexible than Agile.

2.2.2 Prerequisite Reading

Every AO should read the following books prior to attending any DSO training and prior to supporting any DSO program. These books are invaluable to increasing JITC and AO knowledge. The first three books cover awareness and knowledge of DSO practices, goals, and structures. The last two books are about continuous testing and automated testing.

- “DevOps Handbook” (available in DISA eLearning)
- “The Phoenix Project” (available in DISA eLearning)
- “Accelerate,” by Gene Kim, Jez Humble, and Nicole Forsgren, March 2018
- “Continuous Testing for DevOps Professionals,” by Eran Kinsbruner, September 2018
- “The Missing Link,” by Henri Bigot and Valentin Guerlesquin, March 2021

2.2.3 Training and Key Certifications

A DSO section is being added to the JITC Qualification Standard Catalog for workforce development. When applicable competencies, topics, and training venues are established, completion (qualification) will be tracked through the Command Workforce Development program (evolving). The competencies and training needed to conduct a successful DSO T&E effort are associated with these general areas:

- DSO Introduction and General Information
- DSO Test Strategy Development
- DSO Test Execution

Training courses are also available from the Defense Acquisition University (DAU). See Appendix C for the DAU link.

AOs are encouraged to request training and certifications to help expand their knowledge and cross-functional capabilities. The certification pathways will grant the AO real-world experience and credibility while also increasing JITC's credibility and capabilities. AOs should build training and certification pathways into their professional and individual development plans with their supervisors, and supervisors should request the funding on their behalf. AOs are encouraged to seek competitive training opportunities for these certifications and programs:

- Amazon Web Services Certified DevOps Engineer – Professional
- Microsoft Certified Expert Azure DevOps Engineer
- Udacity: Cloud DevOps Engineer Nano Degree

2.2.4 Continuous Testing Maturity Model

AOs and Integrated Test Leads (ITLs) will become familiar with the Continuous Testing (CT) Maturity Model. This six-step model is used as a measure of program CT adoption. The model has been adapted from the "Tricentis Continuous Testing Model" that has been in use for industry since 2017. The original model consisted of five stages, but level 0 has been added here to reflect the starting point most AOs will encounter in DoD (see the DSO Adoption spreadsheet referenced in appendix C):

- Level 0, Manual Testing: This level represents the starting point. Most programs begin here. They are attempting to test everything manually, perhaps without test cases and without official testing roles and responsibilities being developed. The AO engaged with a program at this level must guide them forward into higher levels of testing and acceptance maturity. Otherwise, the program itself is at risk.
- Level 1, Initial: Test design is based on tester intuition. Testing is mostly manual with a script-based approach (traditional). High rates of false positives are manually managed and test data is manually managed. Test environments are separate and manually managed. Developers do API

testing. Behavior-Driven-Development (BDD) with clear acceptance criteria (for example, the CucumberStudio BDD tool) is in place.

- Level 2, Aligned: Test-Driven-Development (TDD) and/or Acceptance-Test-Driven-Development (ATDD) is utilized by development. Risk-based test design is implemented, and risk coverage is tracked. Automation of UI-based functional testing implementation is complete, and the team is using Model-Based Test Automation (MBTA) to minimize false positive rates. Automated tests are primarily focused on emergent functionality (new code) versus system-wide capability.
- Level 3, Managed (minimum level of DSO success): API testing is performed by testers and is automated using mocking and API test tools. MBTA-based UI testing is utilized where automated API and unit testing cannot provide full coverage. Test data management is introduced to empower the team to build functional and unit tests based on managed test data.
- Level 4, Mature: Test data management is ubiquitous throughout the continuous integration (CI) pipeline and supports continuous testing. End-to-end testing is implemented throughout the CI pipeline. Service virtualization is implemented. Continuous testing that provides developers with instant feedback is in place.
- Level 5, Optimized: A comprehensive testing program with full end-to-end testing automation has been established and is actively enhanced. Metrics are continually adjusted (tightened) to ensure continual improvement. CT is fully integrated and enforced in the CI/continuous delivery (CD) pipelines.

2.3 Step 2: Test Methodology Identification

To support a DSO (and/or Agile) program, an AO must collaborate with the PMO to design a test methodology. These programs will typically require a more active role by JITC, and periodic assessments/certifications alone will not provide the DoD with stable and durable results because of the pace of change. There are three primary methods for engaging a DSO, or aspiring DSO/Agile, program.

2.3.1 Embedded Testing

This is service/agency testing with direct JITC participation. This method applies full-time JITC resources to the DSO teams to provide real-time third-party verification and validation of delivery and compliance. The JITC AO (and other resources) will align schedules, test planning, test authoring (automated and manual), test execution, and analysis. JITC testers would need to be embedded with the program and support

Hint: This is a great option for programs that are aspiring to be DSO but do not have the testing expertise required to make it a reality. As DSO evolves, the demand for expertise and SME engagement will also increase. JITC is uniquely positioned to provide cross-functional objective testing at speed to programs that the PMO can rely on, as opposed to the PMO developer that has a stake in the tests passing and the deployment proceeding.

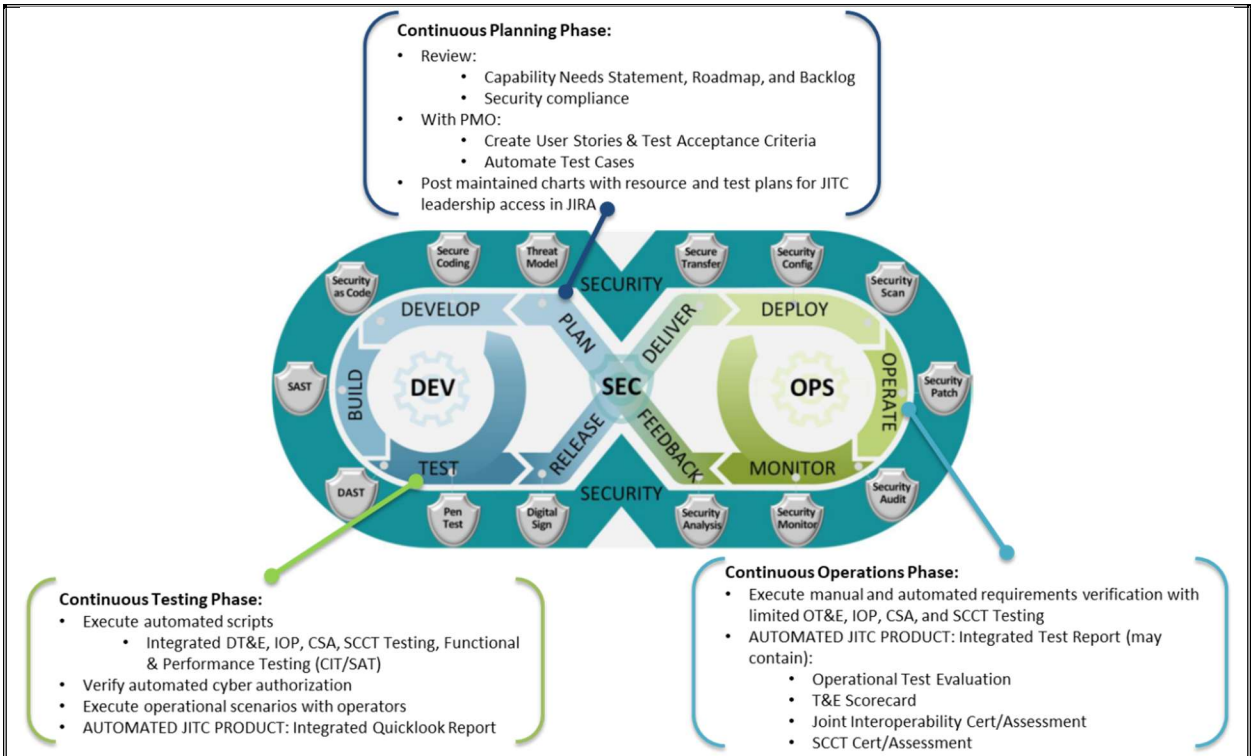
development of automated testing. This would require cross-functional skills related to multiple test disciplines such as Developmental Test and Evaluation (DT&E), Interoperability (IOP), Cybersecurity/Survivability Assessment (CSA), Standards Conformance/Compliance Testing (SCCT), and Operational Test and Evaluation (OT&E) and development of software to support testing and user-based test automation technologies. This embedded test support approach requires highly skilled support and involves constant work and coordination to support the overall software development life cycle.

Requirements: This requires a cross-functionally capable and empowered resource plan that can empower the customer to move at a fast pace. The resources must be able to provide the verification and validation testing within the program's process time and lead time requirements (that is, they cannot be a bottleneck). The program and JITC must agree to empower the embedded testing resources with the authority to approve or disapprove individual release packages and to determine the process for each.

Strengths: JITC provides customers with objective third-party verification and validation testing of acceptance criteria (functional, unit, regression, performance, etc.), compliance (DoD policy, regulations, laws (Section 508), etc.), DT&E, IOP, CSA, SCCT, and OT&E. PMOs can utilize JITC's containerized tools (under development) to facilitate testing and deployment.

Challenges: Introducing third-party oversight to existing modernization contracts could result in contract disputes. Secondarily, the cross-functional resourcing that is capable of performing at speed will take time to develop. Typically, this will require high levels of colocation with the program.

Figure 3 shows an overview of the inputs and outputs for each phase, and actions for the AO, for embedded testing.



LEGEND:

CIT	Contractor Integration Test	OT&E	Operational Test and Evaluation
CSA	Cybersecurity/Survivability Assessment	PMO	Program Management Office
DAST	Dynamic Application Security Testing	SAST	Static Application Security Testing
DEV	Developmental	SAT	System Acceptance Test
DT&E	Developmental Test and Evaluation	SCCT	Standards Conformance/Compliance Testing
IOP	Interoperability	SEC	Security
JITC	Joint Interoperability Test Command	T&E	Test and Evaluation
OPS	Operations		

Figure 3. Embedded Test Support

2.3.2 Periodic Audit and Evaluation (Test Event)

This is a blend of JITC-conducted periodic manual testing, with JITC participation for the active evaluation. This method closely resembles JITC's current periodic T&E events. The primary difference between JITC's current process and a periodic DSO Audit and Evaluation is the addition of an audit of the program's testing methodology, capabilities, efficiency, and risk.

Hint: This option is good for programs that have an objective third party performing continuous independent verification and validation testing but need JITC to perform T&E events. The resourcing and plan are similar to current ones with the addition of the audit. The audit simply ensures that the evaluation remains relevant between events because the DSO pace of change is far greater than traditional SDLC/Waterfall projects.

The audit will ensure the program's capability to discover and remedy compliance, acceptance criteria, and system/security failures in an effort to ensure the program office has designed verification into the process to minimize the risk of

noncompliance over time.

This approach focuses on the review of documents, artifacts, test scripts, and test results. Testers work with the program office to collect the information, analyze data, and help ensure that the automated testing for the test discipline in question happens early in the life cycle. This approach should also verify (audit) that any automated and manual testing adequately addresses the requirements. The approach also verifies that the development, test and any pre-production or deployment environment is representative of the target operational environment. When issues are found, it is recommended that changes and updates be provided back to the PMO to address test data gaps and discrepancies, reducing risk and limitations to testing. When using this approach, it is important to ensure that the user story acceptance criteria are testable and meet the information needs of the testing discipline (for example, for an IOP tester, ensuring a user story involving information transfer has acceptance criteria which address effective information exchanges with timeliness, accuracy, and completeness criteria).

Requirements: Resourcing is similar to current JITC models with a primary AO and supporting resources to accomplish the audit and testing event for compliance. Audit teams would need to be cross-functional (including DT&E, IOP, CSA, SCCT, and OT&E) and able to evaluate and analyze both the methodology in use and the real-world tests being utilized by the program to ensure continual compliance.

Strengths: JITC staffing and resourcing are similar to current processes. This is the lowest cost option for DSO and Agile programs. This introduces the opportunity for persistent compliance monitoring. It can be performed remotely, provided access to the repositories and artifacts is available.

Challenges: The pace of change in DSO programs could introduce methodological inconsistencies and/or discover methodology weaknesses between periods. This could cause disparity between the audited and evaluated versions of the program's methodology and systems architecture and the actual or emergent methodology and architecture. When testers are not included early in the process, additional risks and certification challenges can arise because testing requirements are not addressed early. This causes the PMO to force the testing requirements onto the team after the team has already established its rhythm or cadence. Having to stop work to test changes in requirements or architecture causes other outputs to pile up and requires a major release, drifting from DSO. This can subsequently cause major bottleneck problems that begin a spiraling effect of program failure. Efficiencies in the government requirements approval process are needed to avoid these types of slowdowns.

Figure 4 shows an overview of the inputs and outputs for each phase, and actions for the AO, for periodic audit and evaluation support.

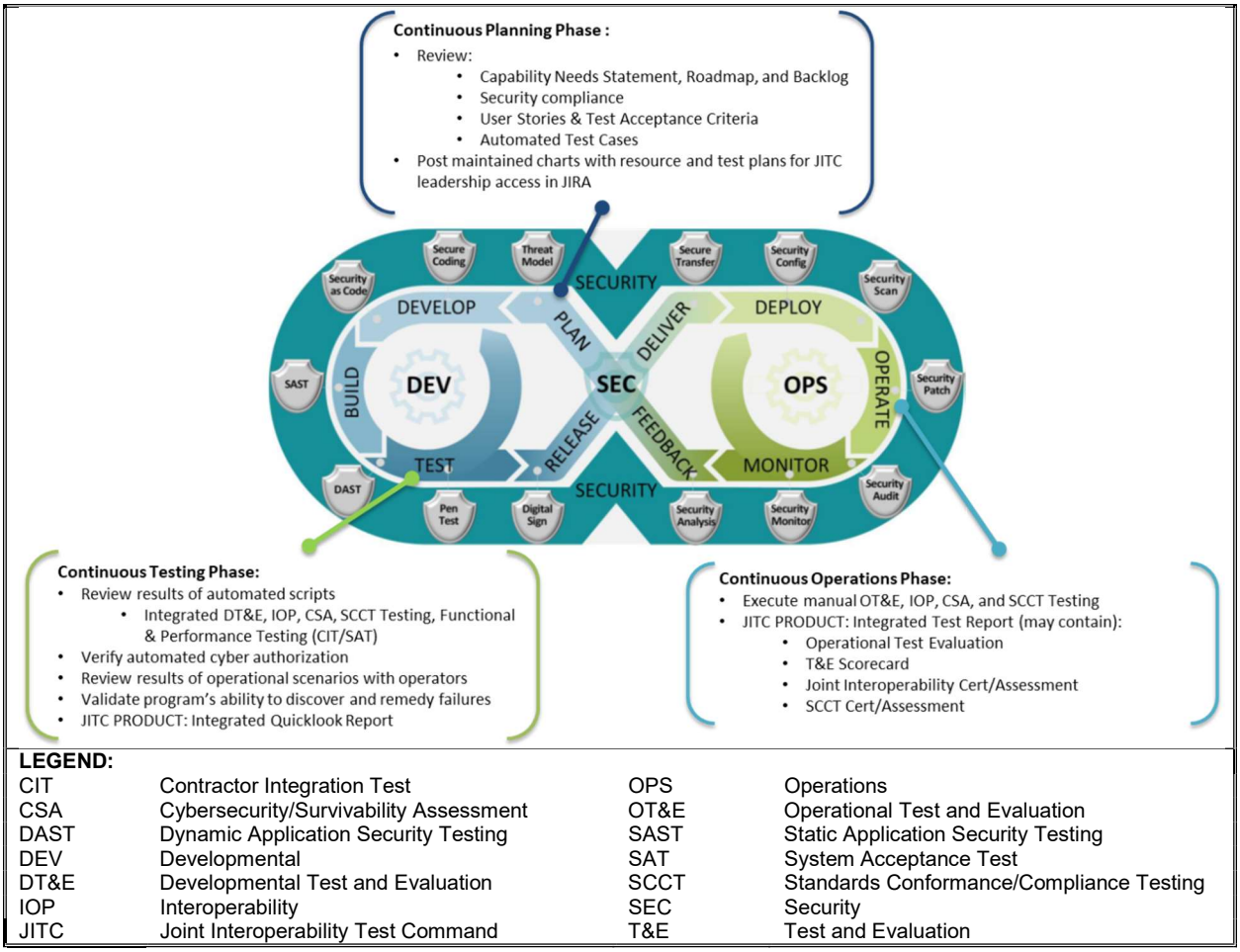


Figure 4. Periodic Audit and Evaluation Support

2.3.3 Hybrid of Embedded and Periodic Audit/Evaluation

Combining a periodic audit and evaluation with an embedded verification and validation methodology is the preferred method to ensure both snapshot compliance and ongoing compliance. This model provides customers with the highest level of assurance that JITC can provide.

Hint: This option is good for high-visibility and/or high-risk programs. Commander's Watch List and oversight programs should trigger the AO to think about this option. Programs with many development teams would also benefit from this option because the pace of change is extreme, and it is highly likely that the PMO would need assistance to stay abreast of, and ensure the quality of, the changes.

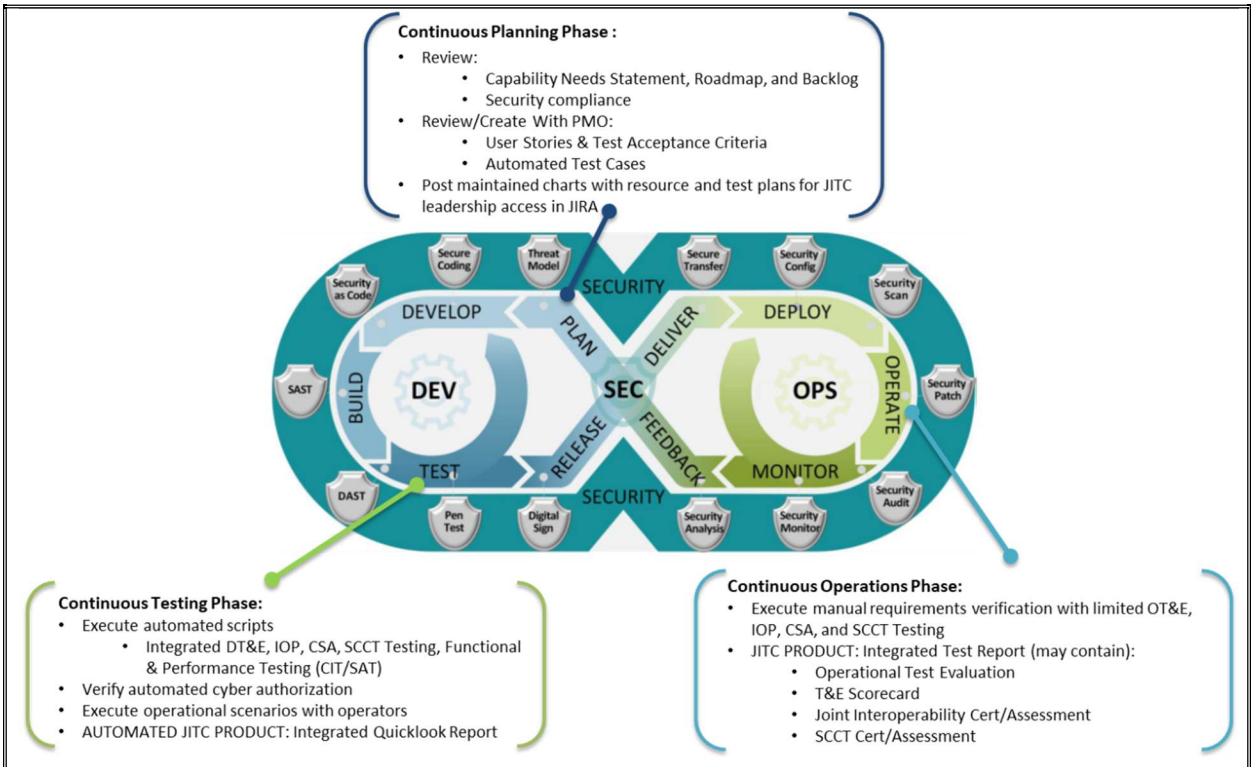
This method embeds third-party JITC verification and validation resources directly into the program's process while also providing periodic audit/evaluation capabilities. Mature DSO programs that have regulatory compliance for Joint IOP Certification (JIC), CSA, SCCT, and/or OT&E are good candidates for this model because it ensures continual compliance and periodic certification.

Requirements: Embedded resources, similar to the first method above and an additional resourcing of the periodic audit and evaluation event (every two years for example), would be required. The audit/evaluation requires different personnel than the embedded team to ensure objectivity.

Strengths: This methodology provides programs the expertise to deliver at a fast pace while also ensuring compliance in real time. It also provides for policy/regulatory compliance for periodic certifications and reporting, in addition to the periodic reflection of methodological audits. PMOs can utilize JITC's containerized tools (under development) to facilitate testing and deployment.

Challenges: This is the highest cost option, but it also provides the best level of service. This will typically require high levels of colocation with the program.

Figure 5 shows an overview of the inputs and outputs for each phase, and actions for the AO, for the hybrid of Embedded and Periodic Audit/Evaluation support.



LEGEND:

CIT	Contractor Integration Test	OT&E	Operational Test and Evaluation
CSA	Cybersecurity/Survivability Assessment	PMO	Program Management Office
DAST	Dynamic Application Security Testing	SAST	Static Application Security Testing
DEV	Developmental	SAT	System Acceptance Test
DT&E	Developmental Test and Evaluation	SCCT	Standards Conformance/Compliance Testing
IOP	Interoperability	SEC	Security
JITC	Joint Interoperability Test Command	T&E	Test and Evaluation
OPS	Operations		

Figure 5. Hybrid of Embedded and Periodic Audit/Evaluation Support

2.3.4 Methodology Overview

The above are the three primary methods for supporting DSO programs. These methods can be modified to fit the needs of a program and/or PMO. For example, the embedded option can be modified from being embedded on every development team to being embedded in the PMO, but still in process, for the deployment pipeline. This situation could be more effective for programs that have multiple development teams. However, AOs must be cautious about introducing bottlenecks when designing a solution. All solutions must move at the lead-time and process-time expectations of the program. JITC/program agreements must include the lead-time and process-time expectations, and JITC approvers (of such agreements) should seek to optimize these through critical analysis of the proposed solution.

Table 1 lists the differences between inputs and outputs of these three primary methodologies.

Table 1. Differences between Test Methods

	Embedded Testing	Periodic Audit and Evaluation	Hybrid
	Service/agency testing with direct JITC participation	Blend of JITC-conducted periodic manual testing with JITC participation for the active evaluation	Combination of direct JITC participation and JITC-conducted test events
Continuous Planning			
System Requirements • Capability Needs Statement • Roadmap • Backlog • Security Compliance	Review from PMO		
Test Requirements • User Stories • Test Acceptance Criteria • Automated Test Cases	Create with PMO	Review from PMO	Review and create with PMO
Command Reporting	Post maintained charts with resource and test plans for JITC leadership access in Jira		
Continuous Testing			
Integrated DT&E, IOP, SCCT, Functional and Performance Testing (CIT/SAT)	Execute automated test scripts	Review results of automated scripts	Execute and review results of automated test scripts
CSA	Verify automated cyber authorization		
Operational Scenarios with Operators	Execute	Review results, validate program's ability to discover and remedy failures	Execute
JITC Product: Integrated Quick Look Report	Automated	Manual	Automated
Continuous Operations			
OT&E, IOP, CSA, and SCCT	Execute limited manual and automated requirements verification	Execute manual testing	Execute limited manual requirements verification
JITC Product: Integrated Test Report (may contain): • Operational Test Evaluation • T&E Scorecard • Joint Interoperability Certification/Assessment • SCCT Certification/Assessment	Automated	Manual	Manual

LEGEND:

CIT	Contractor Integration Test	OT&E	Operational Test and Evaluation
CSA	Cybersecurity/Survivability Assessment	PMO	Program Management Office
DT&E	Developmental Test and Evaluation	SAT	System Acceptance Test
IOP	Interoperability	SCCT	Standards Conformance/Compliance Testing
JITC	Joint Interoperability Test Command	T&E	Test and Evaluation

2.3.5 Alternative Engagement Versions

Table 1 lists the primary methods (Embedded, Periodic, and Hybrid) of engaging a DSO program and form the basis of approach for JITC. Programs also have the following options when exploring T&E approaches:

2.3.5.1 Service/Agency Testing without JITC Participation (Not Recommended)

JITC leverages, to the extent possible, service/agency test results from automated and manual test events in which JITC had no participation and may or may

not have had input. The applicability of such results may be limited because service/agency testing often does not capture JITC's requirements when no coordination has occurred. When JITC is not included in the process, additional risks and certification challenges can arise that will inevitably impact the DSO program's ability to maintain velocity. This happens because testing requirements have not been addressed early, which causes the program office to take reactionary steps to comply with them. This approach is not recommended because it can cause program failures and/or delays. Programs should be encouraged to avoid this risk by addressing T&E requirements early with JITC.

2.3.5.2 JITC Periodic Manual (Only) Testing

This approach is the least desired among PMOs following DSO life cycles. DSO PMOs do not want to stop the flow of software delivery as it is against the core principles of DSO. This is the most antiquated of the options available to PMOs and should not be recommended. Instead, AOs should guide PMOs to baseline testing efforts with the Periodic Audit and Evaluation model.

PMOs utilizing this model will request that JITC perform evaluation events, typically in support of a required certification, and will provide JITC AOs with manual labor-intensive data to analyze and review. This data will be provided in support of a test plan that is manually written and agreed to between JITC and the PMO. The evaluation event itself will happen out-of-cycle from the actual development efforts and runs the risk of introducing unidentified failures that must be mitigated to the program.

Note that a program that insists on this model is probably not a true DSO program. See the DSO identification checklist in Appendix D.

2.3.6 ITL/AO Advice

In many cases, the Auditing approach is the easiest to start with, as it will be the most palatable from the customers' cost perspective while still allowing JITC to accomplish its mission. However, the Embedded and Hybrid approaches provide both JITC and the customer with the most value. A PMO that utilizes JITC with the Embedded approach will be able to enjoy third-party acceptance support and compliance partnership and leverage the ever-expanding JITC testing knowledge base.

To choose between the Hybrid and just the Embedded approaches, the ITL/AO should review program requirements and applicable policy such as the DoD Instruction (DoDI) 8330.01 for joint IOP (triggers the Hybrid model) or for any other periodic JITC certification. The main advantage, outside of periodic certifications, of the Hybrid model is that a different set of "eyes" will provide the PMO with advice on how to improve after each event. This could be invaluable to programs that are on watch lists, oversight, and/or are high-risk/visibility efforts.

2.3.7 Test Teaming

In order to support the DSO life cycle, it is critical to ensure the stakeholder team includes oversight, joint requirements reviewers, user representatives, operators, and testers from the beginning. All these stakeholders must be involved in reviewing the user stories and the process of creating the backlog. They must be available throughout testing to ensure readiness for releases and the verification process during operations. This is a shift in the DoD from verifying outputs to inputs, to find efficiencies and deliver warfighter capabilities at the speed of operations.

2.4 Step 3: Engagement Agreement and Terms of Service

2.4.1 Cost Estimate

Now that the program and JITC (through the primary AO) have agreed in principle on the engagement type, the AO must design a resourcing plan that will determine the ongoing costs. This resourcing plan is part of the Cost Estimation process that already exists at JITC. The AO should estimate the number of contracted, government, and other direct cost (ODC) resources that the engagement will require.

Hint: The recurring theme that should be noted by AOs supporting DSO programs is: "Velocity, velocity, velocity!" or: Anything that negatively impacts, or might negatively impact, throughput should be addressed aggressively and all unnecessary hindrances must be removed.

This is why the traditional silo-based stove-piped organizational structures fail at DSO. There are too many unnecessary hurdles inserted into the value stream, usually in the form of committees, approvals, and interdivisional boundaries.

For Embedded and/or Hybrid efforts, the AO should estimate full-time equivalent (FTE) activity from the JITC-provided resources. Embedded testing requires FTE resourcing because the testing cannot become a bottleneck to the program. If JITC-provided resources are split among multiple programs, the risk of resource overburden is high. DSO programs cannot slow down for JITC; therefore, JITC AOs should include this level of resourcing, and communicate that expectation with the program, from the beginning. Hybrid models will also incur the periodic cost of evaluation events in addition to the embedded resourcing costs. Periodic evaluations cannot utilize the embedded resources because the evaluations must remain objective and neutral.

Cost estimation of stand-alone periodic audit/evaluations will not require any changes from the current cost estimation process.

2.4.2 Customer Agreement

The customer agreement which may consist of the 7600 and the Plan of Action and Milestones (POA&M) document should include the cost estimate, resourcing plan,

and deliverables and should explicitly state the terms of service. These terms should include the program's expectations for process time of the testing services that will bind JITC to meeting the program's pace of delivery. For example, if a DSO program is releasing daily trunk (baseline) changes through a pipeline management tool that creates an A/B production environment, then JITC must apply the appropriate resources to test and validate those deployments in accordance with (IAW) the agreed service structure (Embedded or Hybrid).

Example deliverables for an embedded team are as follows:

- JITC will provide the PMO with automatic security testing/scanning on the Development, Staging, and Production Environments and immediate analysis of results by cybersecurity professionals. Failure notice to the development team(s) and PMO personnel will be automatically provided upon vulnerability detection. Analysis will be provided within 1 hour of detection during normal operating hours. This service will be provided every business day between the hours of 0800 and 1630 Arizona Time.
- JITC will provide the PMO with daily updates to the automated testing scripts IAW the acceptance criteria and regulatory compliance. These automated test scripts will cover integration and interoperability to ensure the program's continuous compliance with DoDI 8330.01. Testing results shall be provided in real-time to the program through the PMO's selected dashboard.
- JITC will provide the PMO with updates to the automated testing scripts IAW the operational evaluation framework. These automated test scripts will cover the critical operational issues. Testing results impacting the Operational Effectiveness, Suitability, and Survivability (OESS) shall be provided in real-time to the program and the Office of the Director, OT&E (DOT&E) through the dashboard.
- The dashboard will automatically roll up results of the automated testing scripts and provide the PMO with daily updates according to the developmental test requirements.
- JITC will provide the PMO with updates to the automated testing scrips IAW the system requirements for developmental, compliance, joint interoperability, and operational testing. Any failures will be reported in real-time to the program through the dashboard. A Quick Look Report or Certification will be automatically generated upon meeting the test requirements, and staffed for JITC review and approval within 10 business days.
- Test results in the dashboard may be utilized to support an Interim Certificate to Operate (ICTO) request and/or an OESS determination.

The agreement period of performance should match the period of performance of the enhancement effort. It should not be piecemealed each year because disruptions to the team can cause bottlenecks that will be unrecoverable for the program. AOs should seek to create these agreements early in the development process (prior to it whenever possible) and extend it for the entirety of the DSO effort.

It is also important that the boundaries of authority for the testing engagement and team are clearly defined in the agreement between JITC and the program. Clearly defining the scope of change that will be allowed at any given time, as well as the authority that the testing team will possess, will ensure higher velocities of delivery for both the program and the JITC supporting testers. One example is as follows:

JITC and the PMO agree that the embedded testing resources shall directly assist the PMO with the acceptance and rejection of releases to the deployment pipeline. The integrated team can approve any releases that are error-free and that do not possess any known or newly detected compliance errors. The testing team can also accept and send to deployment releases with minor issues that do not impact regulatory or security compliance. The team cannot approve, and must disapprove, any releases that introduce IOP or security issues, regressions, vulnerabilities, and/or errors. DoD oversight will be embedded along with the team when necessary.

2.4.3 Funding

There is no significant change to the mechanism of funding for DSO. The same process applies. AOs should advise the PMO and JITC finance office that the Fiscal Service Form 7600 should cover the extended term/period of performance (POP) of the enhancement and testing effort rather than single-year funding plans. Any delays to funding will negatively affect JITC's ability to resource the effort and could result in long-term program disruption.

2.4.4 Resourcing

Now that the agreement and the funding are finalized, it is time for JITC to assign resources to the engagement IAW the resourcing plan. JITC management in collaboration with the AO and the Program Manager of the program should assign resources (and/or contract them). Once assigned, JITC should refrain from changing or modifying the assignments for the entirety of the POP in the POA&M because the resources will get better at providing the service over time. Any replacement also causes storming on the team writ large, which negatively affects velocity.

Hint: Team stability builds trust. Trust builds empowerment. Empowerment enables speed.

Team stability is hugely important to maintaining and then improving the velocity of the team. As relationships build, both internally and externally to JITC, they become more effective. In-process introduction of new resources can cause both bottlenecking (as they learn/adapt) and loss of trust, which leads to program disruption.

2.4.5 Empowering

For embedded teams to succeed, it will become imperative that JITC and the PMO both actively support the acceptance and compliance role of the testing resources.

Developers, administrators, and security engineers will adapt to having a third-party compliance and quality oversight function if the oversight is not undermined or bypassed by the PMO. JITC must ensure the team is keeping pace so the PMO does not lose trust in it. The PMO must ensure that they do not override any negative results which then result in releases being rejected and/or rolled back, but utilize the information to address issues and develop fixes. JITC's value will be highlighted by early defect detection (Mean Time to Detection) and the PMO's follow-on correction.

2.5 Step 4: Integrated Test and Evaluation

2.5.1 Test Requirements

Through the modernization of the DoD software acquisition process, program requirements are being streamlined down to just the Capability Needs Statement, roadmap, and backlog. The AO will need to work with the PMO to review the program requirements documentation to determine the level of integrated test that will be required before and after each release goes operational. Long review cycles will be replaced by DSO stakeholder teams available to review test requirements as the backlog is worked. This DoD guidance calls for transformed processes and an upskilled workforce to manage the enterprise portfolio of capabilities in the cloud and the DSO software factories. Table 2 lists the differences in inputs and outputs between traditional and integrated DSO testing.

Table 2. Differences between Traditional and Integrated DSO Testing

<table>
<tr><th></th><th colspan="2">Traditional Testing</th><th>Integrated DSO Testing</th></tr>
<tr><td rowspan="4">Test Inputs and Outputs</td><td>DT&E</td><td>Test Plan
Test Report</td><td rowspan="4">• User Stories
• Test Acceptance Criteria
• Automated Test Cases
• JIES, Jira, or similar repository for requirements analysis
• Post maintained charts with resource and test plans for JITC leadership access in Jira
• Verify automated cyber authorization
• Automated Quick Look Report
• Data Authentication Group
• Automated Integrated Test Report (may contain):
o Operational Test Evaluation
o T&E Scorecard
o Joint Interoperability Certification/ Assessment
o SCCT Certification/Assessment

*Note: Utilize Dashboard for real-time test results analysis and reporting</td></tr>
<tr><td>IOP</td><td>Joint IOP Evaluation Plan
Test Concept Brief
Test Plan
Test Readiness Review
Quick Look Report
Certification/Assessment Report</td></tr>
<tr><td>SCCT</td><td>Test Plan
Test Report</td></tr>
<tr><td>OT&E</td><td>Evaluation Framework
Data Source Matrix
Test Concept Brief
Test Plan
Test Readiness Review
Data Authentication Group
Test Report</td></tr>
<tr><td>Timeline</td><td colspan="2">8 –12 months</td><td>With each release cycle</td></tr>
<tr><td colspan="4">LEGEND:
DSO Development, Security, and Operations
DT&E Developmental Test and Evaluation
IOP Interoperability
JIES Joint Interoperability Evaluation System
JITC Joint Interoperability Test Command
OT&E Operational Test and Evaluation
SCCT Standards Conformance/Compliance Testing
T&E Test and Evaluation</td></tr>
</table>

2.5.2 Test Phase Activities

Test Phase activities can consist of DT&E (including Functional, Performance, System Integration, and System Acceptance Testing (SAT)), IOP, CSA, SCCT, and initial OT&E. Dependent on JITC's role, testers could be embedded as part of the development team during DT&E execution.

2.5.2.1 Developmental Test and Evaluation

Traditional DT&E is designed to mitigate design risk and ensure compliance with system requirements. The DT&E will also evaluate compliance with operational requirements to minimize risk and certify the system is ready for initial OT&E. In DSO, DT&E must occur more often and at the pace of delivery due to the mechanics of system design and evolution involved in DSO-derived products. The system itself, and the underlying design thereof, are subject to frequent change and adaptation. The intermingling of IaC, container configuration changes, and code changes result in fast-paced changes to system design. JITC will need to provide analysis and testing capabilities in support of DT&E that match program pace and process time requirements. This goal will only be achieved through a combination of proper resourcing and automation (unit testing).

2.5.2.2 Standards Conformance/Compliance Testing

Standards conformance testing serves as a foundation for overall joint IOP and should be conducted prior to IOP test, evaluation, and certification testing in periodic audit and evaluation scenarios.

These tests and test scripts can be automated if the standard is clear. Positioning automated standards conformance and compliance tests close to the developers will help ensure new products are compliant and that new versions do not introduce problems. JITC will need to ensure new requirements utilize or reuse the applicable standards testing as the system evolves.

2.5.2.3 Interoperability Testing/Assessment

IOP testing can fit well into a DSO CI/CD pipeline, especially when automation is used. The JITC role supporting IOP in DSO will include validating test scripts and data outputs, and developing or utilizing automated testing and data collection. The steps to validate a test script include reviewing to ensure the requirement is addressed, confirming the tool is configured properly, and comparing the output against a sample of manual test data. The Joint IOP Evaluation System (JIES) or similar requirements analysis tool like Jira will be necessary for the test requirements review. The intent is to produce a similar output to JIES in a streamlined or automated fashion. Continuous validation can be provided through automated test scripts that ensure the criteria/thresholds for data integrity, timeliness, and periodicity are not negatively impacted by new development and that new integration points meet the IOP criterion.

Hint: Traditional DoD architecture framework products and requirements documents may not exist depending on the acquisition pathway used by the PMO. If they do exist, they may lose their relevance and accuracy very quickly. However, these requirements may be encompassed in historical and emergent user stories, design drawings, and acceptance criteria. The AO, and the embedded team, should follow development planning events closely to ensure tests are covering new/emergent systems integrations and IOP.

For Periodic audit and evaluations (including Hybrid ones), see the JITC Interoperability Process Guide, Version 2, Incorporating Change 1, 30 October 2018, which outlines the procedures and documentation required for joint IOP test and certification, waiver processing, and associated processes and procedures. It addresses IOP test and certification based on the Net-Ready Key Performance Parameter attributes. Also see the IOP Service Area SOP, Version 1.0, July 2020, which establishes standard processes and procedures to conduct T&E in support of JITC's JIC mission. A recertification (think "periodic") may be conducted at a determined interval or based on the program roadmap with a focus on only testing critical interfaces impacted by the upgrades. This will be a program-by-program determination as to when a full certification is needed. The ICTO should be integrated into the fielding strategy to cover gaps between fielding capability and issuance of the JIC. AOs should also stay abreast of JITC policy and Process Guide changes that may incorporate more DSO-friendly certification mechanisms than the current guides.

2.5.2.4 System Integration Testing

The purpose of the System Integration Test (SIT) is to ensure all separate functions of a system are working together. Systems are designed and configured on multiple disparate subsystems (for example, hardware, operating system, and software) that all work together to accomplish a goal. SIT ensures the interactions between these pieces all function. For example, consider the container's integration with the software. Typically, unit testing is performed prior to integration testing. The SIT environment is where the complete system is integrated, integration points are verified, and integrated

functional testing/user testing is performed. In DSO, each developer should have access to a SIT-like environment that mirrors all the configurations, integration points, and unit testing capabilities of the larger test/preproduction environment. Unit tests and automated integration tests should be triggered upon check-in and then again at each subsequent deployment stage in the pipeline.

Most DSO programs adopt a microservice application design. Microservice architectures focus on single-function, self-contained, and stand-alone pieces of software that are independent of one another. This results in simplicity for deployment, risk minimization for changes, and better clarity when making changes. However, nothing in life is free. Microservice architectures introduce complexity to SIT. JITC AOs will need to assist programs in managing the expansive and ever-increasing test scripts to validate both existing functionalities and emergent ones. This makes SIT a very important aspect of DSO success.

Microservices – also known as the microservice architecture – is an architectural style that structures an application as a collection of services that are:

- Highly maintainable and testable
- Loosely coupled
- Independently deployable
- Organized around business capabilities
- Owned by a small team

The microservice architecture enables the rapid, frequent, and reliable delivery of large, complex applications. It also enables an organization to evolve its technology stack.

Note that larger scale PMOs may also adopt a service-oriented architecture framework that covers multiple systems' interactions and integrations. These larger architecture frameworks influence both SIT and IOP testing.

JITC can help ensure that integration test scripts are authored, valid, and provide neutral verification of them for the program. AOs and the engagement team can provide both periodic reviews/evaluations and continuous integration testing services.

2.5.2.5 System Acceptance Testing

SAT determines whether a system satisfies stakeholder needs. Traditionally, the tests are repeated every time there is a new delivery to the government or the user community. In DSO, and some Agile methodologies, these tests are defined prior to solution development. The standard process is that the stakeholders/team define acceptance criteria, the development/enhancement team then configures tests to confirm said acceptance criteria, and finally the enhancement team configures/codes the

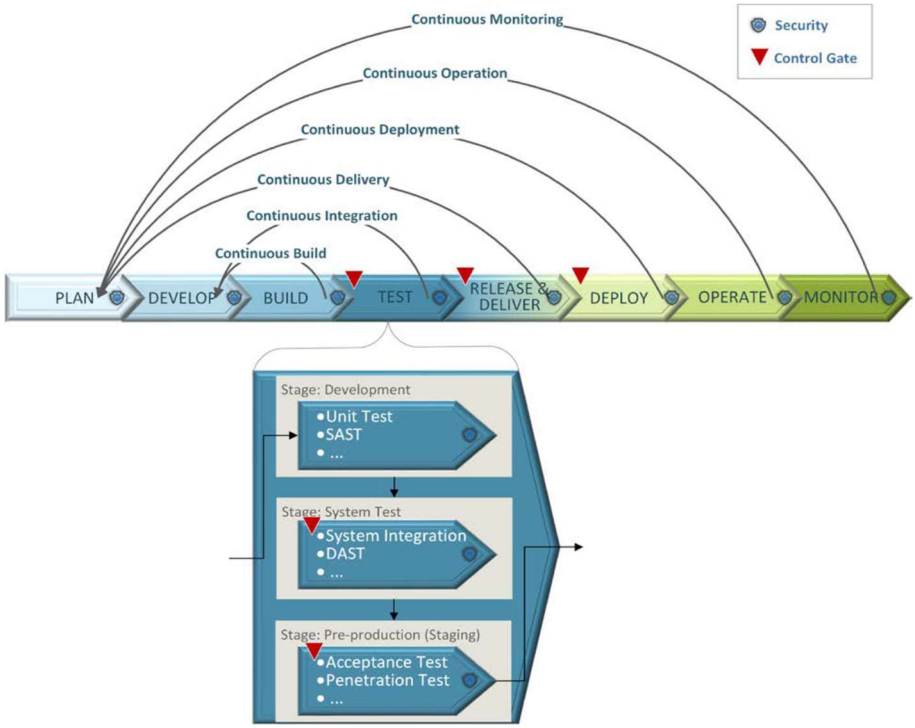
Hint: AOs should notice that their interactions will no longer be "after" something is done. Acceptance Testing is incorporated into the process before development in DSO (and some Agile). The "throw it over the fence" mentality of the past will no longer work. JITC will need to adapt to continuous flow and early integration with the program's enhancement teams.

solution to pass the tests. This model of TDD was used in the Agile XP (eXtreme Programming) methodology for two decades prior to being adopted by DSO. It is one of the more difficult processes to start and enforce for most PMOs that are in the beginning stages of DSO adoption. JITC AOs, as testing experts, should seek to become SMEs in this model.

2.5.2.6 Operational Assessment

Operational scenarios must shift left to incorporate testing with operators before release. In order to enable the operational testing as early as possible, developed capabilities would need to be ready for execution of operational tasks. The development environment must mirror the production environment as well. Acceptance criteria in DSO includes performance, suitability, and cyber requirements. Operators and stakeholders must be included in the DSO test team to ensure testing is accomplished early to support release schedules in the program roadmap. The readiness of the released capability will drive when the operational assessment may occur, which could be as early as the test phase, or later during operations.

Figure 6 is an example of the DSO processes for an application.



NOTE: Source: DoD CIO, "DoD Enterprise DevSecOps Reference Design," Version 1.0, 12 August 2019

LEGEND:

DAST	Dynamic Application Security Test	SAST	Static Application Security Test

Figure 6. Application DevSecOps Processes

2.5.3 Operations Phase Activities

Testing during the Operations Phase will focus on collecting data on any remaining IOP, CSA, SCCT, and OT&E requirements that were not yet verified during previous testing in a production environment. This would include any releases that complete a certain capability that is now ready for operational testing. Operational users must support the remaining verification activities in the operational environment. Continuous monitoring on the production environment should include non-invasive tests, smoke checks, security monitoring, functional monitoring, and performance monitoring.

OT&E test results support OESS determinations or progress toward OESS. See the JITC Operational Test and Evaluation Guidebook, Version 2.0, 5 October 2017, for more details on executing OT&E.

2.5.4 Cybersecurity Activities

AOs/combined test teams need to work with the PMO to get access or set up the appropriate tools in the pipeline and determine when and where data collection will occur. Data may be collected from automated or manual tests. Testing can and should occur at each step in a DSO pipeline. Figure 1 shows the DSO test integration points. See Appendix H for additional information on cyber related activities. Additionally, the DoD's "DoD Enterprise DevSecOps Fundamentals Guidebook: DevSecOps Tools and Activities," Version 2.0, March 2021, outlines the security activities and where they fit into each phase of DSO.

2.6 Step 5: Engagement Wrap-up/Closing

Ideally, JITC's engagement will begin at the program's inception and close when the system reaches maturity. Throughout the engagement, the ITL will need to track resource utilization and delivery of test products in accordance with the overarching agreement (7600/POA&M) to advise program management of upcoming milestones, funding levels, and potential engagement close-out dates.

In the DSO environment, the program definition of done does not just mean it is coded and tested; it means it will work in production and that the developer can receive as much of that feedback (from testing) as quickly as possible. When a system reaches maturity, the DSO framework may not be the optimal operation and maintenance (O&M) framework going forward. As such, the AO should work with the PMO to schedule closure and deallocation of dedicated personnel, ODCs, and contracted resources at least 6 months prior to a deallocation. Engagement close-out that occurs prior to final delivery of the product or system must be closely coordinated with the PMO to ensure minimal pipeline and team velocity impacts. Additionally, the AO should engage with the PMO to establish periodic compliance evaluations and new agreements for the O&M stage of the system. These agreements will then realign with traditional JITC processes.

AOs should also engage in debriefs with the PMO at DSO engagement closure. Lessons learned and retrospective out-briefs are valuable knowledge for JITC's continued improvement over time.

3. JITC TEST ROLES

JITC AOs will provide support in the following test roles, depending on the expertise and test methodology required:

- Integrated Test Lead
- Supporting Testers
- Data Lead
- Data Collectors

3.1 Integrated Test Lead

The ITL is the lead JITC AO assigned to, and responsible for, the direction of the testing. The ITL is responsible for initial customer/program engagement, defining the test engagement parameters, cost, and period of performance, as well as ensuring all resources assigned are delivering IAW agreements. This includes planning for all test activities (DT&E, IOP, CSA, SCCT, and OT&E), establishing the data management process, and addressing instrumentation and tool requirements. Ensuring cross-functional testing is imperative to DSO program success. As such, the ITL should seek to resource a testing engagement with the skill sets to build, maintain, enhance, and continuously improve a testing regimen at all levels of the DSO process (individual developer VM/container, development environment, test/stage, and production).

3.2 Supporting Tester

Multiple testers representing different service areas can make up a combined test team supporting the overall effort to automate testing and continuously support the DSO program. Supporting testers can be resourced to provide specific testing skill sets and/or to augment the testing capabilities within a skill set to meet the program's needs. For example, a DSO program that needs an embedded testing team would resource IOP, SIT, SAT, CSA, and other skills to the team, and pull from any other test organizations participating. They may also anticipate needing extensive cyber testing so the team could be augmented with multiple cyber/information security testers.

3.3 Data Lead

Traditionally, a Data Manager and Data Collectors would be required to collect, manage, analyze, and report on the data collected. In DSO, the Data Lead should work with, as a part of, the test team to determine the necessary automated capabilities that support testing and transparency of data to all levels of the DSO program's team. This will include automated tools for data management. The team will need access to such data as automated test scripts/cases, test results (including immediate access to

failures), acceptance criteria, historical failures, deprecated tests, and system statuses. The test tools should collect the data and store that information in a common area where automated scripts can process the information to provide the reports necessary, including publishing the results immediately after any test to information radiators (for example, dashboards) for the team to process. For most engagements, this will be a secondary role of the ITL/AO.

3.4 Data Collectors

Most data should be collected and most problems caught by the automated tests. This includes emergent problems. Once they are detected and measurable, new automated test scripts should be written to capture any reoccurrence. Data Collectors can support any manual data collection required beyond the automated testing. This may include administering surveys and preparing Test Incident Problem Reports on system problems, deficiencies, and anomalies in accordance with the test plan (which would then create new automated tests). Manual and automated test results should be accessible by the team as quickly as possible. The Data Collector's role is to publish these results after capturing them. For most engagements, this will be a secondary role performed by every member of the test team.

INTENTIONALLY BLANK

APPENDIX A

ACRONYMS

A&S	Acquisition and Sustainment
AA	Adversarial Assessment
AI	Artificial Intelligence
API	Application Programming Interface
AO	Action Officer
ATDD	Acceptance-Test-Driven-Development
ATO	Authority to Operate
BDD	Behavior-Driven-Development
CD	Continuous Delivery
CEVA	Cyber Economics Vulnerability Assessment
CI	Continuous Integration
CIO	Chief Information Officer
CNCF	Cloud Native Computing Foundation
COTS	Commercial-off-the-shelf
CS	Cybersecurity
CSA	Cybersecurity Assessment (DT&E)
CSA	Cyber Survivability Assessment (OT&E)
CT	Continuous Testing
CVPA	Cooperative Vulnerability and Penetration Assessment
DAU	Defense Acquisition University
Dev	Development
DevOps	Development Operations
DevSecOps	Development, Security, and Operations
DISA	Defense Information Systems Agency
DoD	Department of Defense
DoDI	DoD Instruction
DOT&E	Director, Operational Test and Evaluation
DSO	Development, Security, and Operations
DT	Developmental Test
DT&E	Developmental Test and Evaluation
EF	Evaluation Framework
ETI	Early Test Involvement
FTE	Full-time-equivalent

IaaS	Infrastructure as a Service
IaC	Infrastructure as Code
IAW	In Accordance With
ICTO	Interim Certificate to Operate
IOP	Interoperability
IT	Information Technology
ITL	Integrated Test Lead
JIC	Joint Interoperability Certification
JIES	Joint Interoperability Evaluation System
JITC	Joint Interoperability Test Command
KPI	Key Performance Indicator
MBTA	Model-Based Test Automation
MVP	Minimal Viable Product
MVCR	Minimum Viable Capability Release
NetOps	Network Operations
NRRB	NetOps Readiness Review Board
OCI	Oracle Cloud Infrastructure
ODC	Other Direct Cost
OESS	Operational Effectiveness, Suitability, Survivability
O&M	Operations and Maintenance
Ops	Operations
OT	Operational Test
OT&E	Operational Test and Evaluation
OTA	Operational Test Agency
OUSD	Office of the Under Secretary of Defense
PaaS	Platform as a Service
PM	Program Manager
PMO	Program Management Office
POA&M	Plan of Action and Milestones
PoP	Period of Performance
SaaS	Software as a Service
SaC	Security as Code
SAS	Scorecard Assessment Strategy
SAT	System Acceptance Test
SCCT	Standards Conformance/Compliance Test

SDLC	Software Development Life Cycle
Sec	Security
SIT	System Integration Test
SME	Subject Matter Expert
T&E	Test and Evaluation
TDD	Test-Driven-Development
UI	User Interface
V	Version

INTENTIONALLY BLANK

APPENDIX B

GLOSSARY OF TERMS

Term	Definition
Agile	Agile is a software development process that emphasizes short, iterative planning and development cycles to provide better control and predictability and to support changed requirements as projects evolve.
Artifact	The documentation or any deliverable associated with a project that helps to describe the functions, architecture, and design of the software being developed.
Backlog	A list of requirements/functionality intended for releases and broken out by user stories.
Continuous Integration/ Continuous Delivery (CI/CD) Orchestrator	CI/CD orchestrator is a tool that enables fully or semi-automated short-duration software development cycles through integration of build, test, secure, and store artifacts tools. CI/CD orchestrator is the central automation engine of the CI/CD pipeline.
Continuous Integration/ Continuous Delivery (CI/CD) Pipeline	CI/CD pipeline is the set of tools and the associated process workflows to achieve continuous integration and continuous delivery with build, test, security, and release delivery activities, which are steered by a CI/CD orchestrator and automated as much as practice allows.
Cloud Native Computing Foundation	According to the CNCF website, CNCF is an open-source software foundation dedicated to making cloud native computing universal and sustainable. Cloud native computing uses an open-source software stack to deploy applications as microservices, packaging each part into its own container, and dynamically orchestrating those containers to optimize resource utilization. Cloud native technologies enable software developers to build great products faster (See https://www.cncf.io/)
Cloud Native Computing Foundation (CNCF)-certified Kubernetes	CNCF has created a Certified Kubernetes Conformance Program. Software conformance ensures that every vendor's version of Kubernetes supports the required APIs. Conformance guarantees interoperability between Kubernetes from different vendors. Most of the world's leading vendors and cloud computing providers have CNCF-certified Kubernetes offerings.
Container	A standard unit of software that packages code and all its dependencies down to, but not including, the Operating System (OS). It is a lightweight, standalone, executable package of software that includes everything needed to run an application except the OS code, runtime, system tools, system libraries, and settings.
Continuous Build	Continuous build is an automated process to compile and build software source code into artifacts. The common activities in the continuous build process include compiling code, running static code analysis such as code style checking, binary linking (in the case of languages such as C++), and executing unit tests. The outputs from the continuous build process are build results, build reports (for example, the unit test report and a static code analysis report), and artifacts stored into an artifact repository. The trigger to this process could be a developer code commit or a code merge of a branch into the main trunk.
Continuous Delivery	Continuous delivery is an extension of continuous integration to ensure that a team can release the software changes to production quickly and in a sustainable way. The additional activities involved in continuous integration include release control gate validation and storing the artifacts in the artifact repository, which may be different from the build artifact repository. The trigger to these additional activities is successful integration, which means all automation tests and security scans have passed. The human input from the manual test and security activities should be included in the release control gate. The outputs of continuous delivery are a release go/no-go decision and released artifacts, if the decision is to release.
Continuous Deployment	Continuous deployment is an extension of continuous delivery. It is triggered by a successful delivery of released artifacts to the artifact repository. The additional activities for continuous deployment include, but are not limited to, deploying a new release to the production environment, running a smoke test to make sure essential functionality is working, and a security scan. The output of continuous deployment includes the deployment status. In the case of a successful deployment, it also provides a new software release running in production. On the other hand, a failed deployment causes a rollback to the previous release.

Term	Definition
Continuous Integration	Continuous integration goes one step further than continuous build. It extends continuous build with more automated tests and security scans. Any test or security activities that require human intervention can be managed by separate process flows. The automated tests include, but are not limited to, integration tests, a system test, and regression tests. The security scans include, but are not limited to, dynamic code analysis, test coverage, dependency/Bill of Materials checking, and compliance checking. The outputs from continuous integration include the continuous build outputs plus automation test results and security scan results. The trigger to the automated tests and security scan is a successful build.
Continuous Monitoring	Continuous monitoring is an extension of continuous operation. It continuously monitors and inventories all system components, monitors the performance and security of all the components, and audits and logs the system events.
Continuous Operation	Continuous operation is an extension of continuous deployment. It is triggered by a successful deployment. The production environment operates continuously with the latest stable software release. The activities of continuous operation include, but are not limited to, system patching, compliance scanning, data backup, and resource optimization with load balancing and scaling (both horizontal and vertical).
Definition of Done	In software development, a shared understanding of what it means for work to be complete.
Development, Security, Operations (DevSecOps, DSO)	DevSecOps is a software engineering culture and practice that aims at unifying software development (Dev), security, (Sec) and operations (Ops). The main characteristic of DevSecOps is to automate, monitor, and apply security at all phases of software development: plan, develop, build, test, release, deliver, deploy, operate, and monitor.
DSO Ecosystem	A collection of tools and process workflows created and executed on the tools to support all the activities throughout the full DSO life cycle. The process workflows may be fully automated, semi-automated, or manual.
DSO Pipeline	The DSO pipeline is a collection of DSO tools, upon which the DSO process workflows can be created and executed.
Factory, Software Factory	A software assembly plant that contains multiple pipelines, which are equipped with a set of 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. The software factory supports multi-tenancy.
Infrastructure as Code	The management of infrastructure (networks, virtual machines, load balancers, and connection topology) in a descriptive model, using the same versioning that the DSO team uses for source code. Infrastructure as Code evolved to solve the problem of environment drift in the release pipeline.
Infrastructure as a Service (IaaS)	IaaS is an instant computing infrastructure, provisioned and managed over the internet.
Minimum Viable Capability Release (MVCR)	MVCR is the first software version containing sufficient capability to be fielded for use.
Minimum Viable Product (MVP)	MVP is an early version of the software with just enough features to meet basic minimum functional capabilities.
Open Source	Refers to a program or application with source code that can be modified by anyone.
Orchestration Pipeline	Tools or products that enable the various automated tasks that make up a continuous delivery pipeline to be invoked at the right time.
Platform as a Service (PaaS)	PaaS is a cloud computing model where a third-party provider delivers hardware and software tools to users over the internet.
Pipeline	A sequence of orchestrated, automated tasks implementing the software delivery process for a new application version.
Repository	A central place in which data is aggregated and maintained in an organized way.
Roadmap	The roadmap is a high-level document designed to capture and communicate a product's strategic objectives, priorities, and plans.
Shifting Left	Shifting left refers to integrating risk assessments, security testing, and compliance evaluation processes earlier in the delivery pipeline.

Term	Definition
Software as a Service (SaaS)	SaaS is a method of software delivery and licensing in which software is accessed online via a subscription, rather than bought and installed on individual computers.
Test-Driven-Development	A software development process that relies on the repetition of a short development cycle where requirements in the form of test cases are used to improve software.
Toolchain	Use of an integrated set of task-specific tools to automate an end-to-end process.
Validation	According to the DAU glossary, the verification process provides the evidence that the system or system element performs its intended functions and meets all performance requirements listed in the system performance specification and functional and allocated baselines. Verification answers the question, "Did you build the system correctly?" Verification is a key risk reduction activity in the implementation and integration of a system and enables the program to catch defects in system elements before integration at the next level, thereby preventing costly troubleshooting and rework.
Verification	According to the DAU glossary, the validation process provides the objective evidence that the capability provided by the system complies with stakeholder performance requirements, achieving its use in its intended operational environment. Validation answers the question, "Is it the right solution to the problem?" Validation consists of evaluating the operational effectiveness, operational suitability, sustainability and survivability (including cybersecurity), or lethality of the system or system elements under operationally realistic conditions.
Virtual Machine	Software that emulates a physical device.
Waterfall	A software development methodology based on a phased approach to a project, from requirements gathering through development to operations, in a linear sequential flow.

INTENTIONALLY BLANK

APPENDIX C

REFERENCES

DEPARTMENT OF DEFENSE REFERENCES

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

Department of Defense (DOD) Chief Information Officer, "DoD Enterprise DevSecOps Reference Design," Version 1.0, 12 August 2019
https://dodcio.defense.gov/Portals/0/Documents/DoD%20Enterprise%20DevSecOps%20Reference%20Design%20v1.0_Public%20Release.pdf?ver=2019-09-26-115824-583

DoD, "DoD Enterprise DevSecOps Fundamentals Guidebook: DevSecOps Tools and Activities," Version 2.0, March 2021
https://software.af.mil/wp-content/uploads/2021/05/DoD-Enterprise-DevSecOps-2.0-Tools-and-Activities-Guidebook.pdf

DoD, "DoD Cloud Computing Security Requirements Guide," Version 1, Release 3, 6 March 2017
https://public.cyber.mil/dccs/dccs-documents/

DoD, "Cybersecurity Test and Evaluation Guidebook," Version 2.0, Change 1, 10 February 2020
https://www.dau.edu/cop/test/DAU%20Sponsored%20Documents/Cybersecurity-Test-and-Evaluation-Guidebook-Version2-change-1.pdf

"DoD Enterprise DevSecOps Initiative," Version 5.5, 15 September 2020
https://software.af.mil/dsop/documents/

DoDI 8330.01, "Interoperability of Information Technology (IT), Including National Security Systems (NSS)," 21 May 2014, Incorporating Change 2, 11 December 2019
https://www.esd.whs.mil/Portals/54/Documents/DD/issuances/dodi/833001p.pdf

DoD Office of the Under Secretary of Defense (OUSD) for Acquisition and Sustainment (A&S), "Minimum Viable Product (MVP) and Product Roadmap," Version 1.0, August 2019
https://www.dau.edu/cop/it/DAU%20Sponsored%20Documents/ProductRoadmapAndMVP_WhitePaper%20V1.0%20FINAL.pdf

DoD, "OSD DevSecOps Best Practice Guide," Version 1.0, 15 January 2020
https://www.dau.edu/cop/it/DAU%20Sponsored%20Documents/DevSecOps_Whitepaper_v1.0.pdf

DoD OUSD(A&S), “Software Acquisition Strategy: Agile Guidance,” Version 1.2, 15 December 2019
https://www.dau.edu/cop/it/DAU%20Sponsored%20Documents/WP-ASD-v1.2.1.docx

Joint Federated Assurance Center Software Assurance Technical Working Group PowerPoint Brief, “DevSecOps Lifecycle Charts v5,” 26 August 2019
https://intelshare.intelink.gov/sites/dodswawg/Shared%20Documents/2019-08-27/DevSecOps%20Lifecycle%20charts%20v5.pptx

JOINT INTEROPERABILITY TEST COMMAND REFERENCES

Defense Information System Agency Joint Interoperability Test Command (JITC), “JITC Service Catalog,” 6 April 2020
https://disa.deps.mil/ORG/JITC/StrategicPlanCollaborationSite/JITCServices/JITC%20Service%20Catalog%202020%20(002).pdf

JITC, “Development, Security, and Operations (DevSecOps) Cybersecurity Test & Evaluation (Cyber T&E) Tool Requirements,” Version 1.0, 31 July 2020
https://disa.deps.mil/org/JTD/JTD3/JTD3%20Document%20Library/Cyber%20T+E

JITC Instruction 380-50-02, “JITC Interoperability and Standards Conformance Test and Evaluation (T&E) and Certification Instruction,” 9 January 2017
https://jitc.fhu.disa.mil/jitcnet/instruct/380/3805002_Jan2017.pdf

JITC, “JITC Cyber Test and Evaluation Guidebook Version 1.02,” September 2019
https://disa.deps.mil/org/JTA/JITC%20Cyber%20Guidebook/JITC%20Cyber%20T-E%20Guidebook,%20v1-02%20(2019-11-06).pdf

JITC, “Joint Interoperability Evaluation Guidebook, Version 2,” June 2019
https://disa.deps.mil/org/JT4/JT4%20Public%20Shared%20Document%20Library/JT4A/Joint%20IOP%20Evaluation%20Guidebook%20Final-Signed.pdf

JITC, “JITC Interoperability Process Guide, Version 2.0 Change 1,” 30 October 2018
http://jitc.fhu.disa.mil/organization/references/publications/index.aspx
(Note: Access to the JITC Industry Toolkit (JIT) is required.)

JITC, “IOP Service Area SOP v1.0,” July 2020
https://disa.deps.mil/ORG/JITC/IPTS/IIPT/IIPT%20Products1/IOP%20Service%20Area%20SOP%20v1.0.pdf

JITC, “Notional Guide for Action Officers, Version 2.1,” June 2019
https://disa.deps.mil/org/JT4/JT4%20Public%20Shared%20Document%20Library/JT4A/Notional%20Guide%20-%20Architectures%20for%20TEC%20v2.1_Final.pdf

JITC, “JITC Operational Test and Evaluation Guidebook, Version 2.0,” 5 October 2017
https://disa.deps.mil/ORG/JITC/JITC%20Collaboration/JITC%20OTE%20Guidebook/Forms/AllItems.aspx

JITC, “JITC Test and Evaluation Scorecard Guidebook, Version 3.0,” November 2020
https://disa.deps.mil/org/JTA/TE%20Scorecard/01%20-%20Test%20and%20Evaluation%20-%20Current%20Version%20(Ver.%203.0)

MILITARY SERVICES REFERENCES

Department of the Air Force PowerPoint Brief, “DoD Enterprise DevSecOps Initiative & Platform One,” Version 5.5, 15 September 2020
https://software.af.mil/dsop/documents/

United States Army, Defense Acquisition Guidebook, February 2017, Chapter 8, Test and Evaluation
https://www.dau.edu/guidebooks/Shared%20Documents%20HTML/Chapter%208%20Test%20and%20Evaluation.aspx

OTHER REFERENCES

Kessel Run Software Factory
https://software.af.mil/softwarefactory/kessel-run

Platform One: DoD Enterprise DevSecOps Services
https://software.af.mil/team/platformone/

Software Factories
https://software.af.mil/software-factories/

Digital AI, Periodic Table of DevOps Tools
https://digital.ai/periodic-table-of-devops-tools

CNCF
https://www.cncf.io

Defense Acquisition Acronyms and Terms
https://www.dau.edu/glossary/Pages/Glossary.aspx

Tricentis Continuous Testing (Maturity) Model
https://www.tricentis.com/products/what-is-continuous-testing/

Defense Acquisition University, “DAU Agile Software and DevSecOps Training,” 17 August 2020
https://www.dau.edu/training/career-development/logistics/blog/DAU-Agile-Software-and-DevSecOps-Training

General Services Administration DevSecOps Guide
https://tech.gsa.gov/guides/dev_sec_ops_guide/

DSO Adoption Spreadsheet
https://disa.deps.mil/ORG/JITC/IPTS/DSO/Shared%20Documents/LOE%203%20Processes%20and%20Methodology/Methodology/DSO%20Adoption.xlsx

APPENDIX D

IDENTIFYING DSO (CHECKLIST)

Identifying whether a program is Development, Security, Operations (DSO or DevSecOps), Agile, Waterfall, or even uncontrolled can be difficult. There are many traits to compare, and some of which are common to multiple methodologies. The Table D-1 matrix helps the Action Officer identify whether or not a program complies with DSO principles or fits into the mold of another methodology. Generally, a DSO program will comply with all of the green traits in the table. The yellow traits are common to multiple other methodologies (X = Yes or True, S = Sometimes). The red traits are disqualifiers for a program to be DSO. These are based on industry standards. The Department of Defense Enterprise DevSecOps Reference Guide (Version 1.0 12 August 2019, page v) lists the following goals:

- Reduced mean time to production: the average time it takes from when new software features are required until they are running in production.
- Increased deployment frequency: how often a new release can be deployed into the production environment.

Table D-1. Identifying DSO Traits

Traits	DSO	Agile	Waterfall	Uncontrolled
Integrated Teams				
Shared management	X	X		
Colocation	X	X		
Common meeting attendance	X	S	S	
Shared goals (delivery of working and valuable systems)	X	S		
Teams are cross-functional (include developers, system admins, and security)	X			
Teams are empowered to release and to roll back releases when necessary	X			
Systems Admins/Architects report to a different manager than developers		X	X	X
Security reports to a different manager		X	X	X
Security Controls and Testing				
Production STIGs enforced on Developer Machines	X	S		
Production STIGs Development Environment (Trunk)	X	S		
STIGs applied to UAT/Preproduction and Production	X	X	X	
Security scanning tools such as ACAS, NESSUS, Angry Scanner, etc. Triggered on code check-in at developer machine, development, and all higher environments.	X			
Security is written into Definition of Done	X	X		
New solution discussions always include security and systems professionals	X	S		
New solution's acceptance criteria include security and systems criterion	X	S		
Automation				
Developer Machine triggered on each check-in	X			
Development Environment triggered on trunk check-in	X	S		
UAT/Preproduction Environment part of formal deployment process	X	X	X	
Production Environment - smoke and nonintrusive testing	X	X	X	
Developers author test scripts for all acceptance criteria	X	S		
Deployment is automated within a defined pipeline (tool)	X	S		

Table D-1. Identifying DSO Traits (continued)

Traits	DSO	Agile	Waterfall	Uncontrolled
Results (from all environments) are shared automatically with the team(s)	X	S		
Infrastructure as Code (IaC) is used for deployments	X	S		
A/B and/or blue/green deployment schemes are utilized for efficient rollback	X			
Release Size				
Micro Releases (as small as a single change at a time)	X			X
Medium Releases (1 to 4 weeks of work by the whole team)		X		X
Big Bang Releases (1 large release after several months)			X	X
Release Pace				
Multiple Tiny Releases per Day	X			X
One per day	X			X
Multiple Per Week	X	S		X
1 Per Week		X		X
Biweekly		X		X
Monthly		X		X
> Monthly			X	X
Planning, Monitoring and Controlling				
Just in Time (JIT) planning	X	X		X
Heavy planning and requirements documentation up-front			X	
PMO Focuses on bottlenecks and the value stream's efficiency/efficacy	X	S		
Solutions are feedback driven	X	X		
Key Performance Indicators/Parameters				
Time to Delivery (Lead Time)	X	X		
Process Time (how long at each "station" in the value stream)	X			
Percent Complete and Accurate (%C/A)	X	X		
Time to Feedback/Resolution	X	X	S	

(The legend is on the next page.)

Table D-1. Identifying DSO Traits (continued)

KEY:			
Green	DSO trait	S	Sometimes
Yellow	Trait shared with another methodology	X	Yes/True
Red	Not DSO		
LEGEND:			
ACAS	Assured Compliance Assessment Solution	STIG	Security Technical Implementation Guides
DSO	Development, Security, and Operations	UAT	User Acceptance Test

APPENDIX E

ETI AND T&E SCORECARD FOR DISA PROJECTS

1. EARLY TEST INVOLVEMENT APPLICABILITY

Early Test Involvement (ETI) engages Program Management Offices (PMOs) for Defense Information Systems Agency (DISA) projects early in order to address additional requirements necessary to acquire a Network Operations (NetOps) Certification Letter from DISA's NetOps Readiness Review Board (NRRB). ETI assists Joint Interoperability Test Command (JITC) Action Officers (AOs) and PMOs with developing a Scorecard Assessment Strategy (SAS). Establishing a SAS will facilitate the successful execution of NRRB assessments by structuring and facilitating pre-test planning activities via the JITC-led Test and Evaluation (T&E) Working-level Integrated Product Team.

The SAS provides assurance to the NRRB that JITC and the PMO have a joint approach to T&E assessments and can agree on necessary requirements, risks, evaluation approaches, and resources.

ETI supports JITC's Integrated Test Strategy by engaging with the PMO as early as possible to ensure details such as requirements, risk, test criteria, and resource requirements are identified during project initiation.

2. DSO INTEGRATED TEST STRATEGY

Testing in DSO spans the entire software development, security, and operations life cycle. It is crucial to involve testers and test design early. Testers ensure the definitions of done (list of requirements) is testable (measurable) and clear. In the DSO environment, the definition of done does not just mean it is coded and tested; it means it will work in production and that the developer can receive as much of that feedback (from testing) as quickly as possible.

The testers should be involved in test design, verification design, and acceptance design. By engaging professional testers early, ideally prior to acquisition processes, the PMO can ensure testing feedback loops are built into the product and acceptance support is built into the solution. Early engagement includes user story development and acceptance criteria definition, functional testing, feature verification, operations testing, performance testing, security testing, as well as being able to monitor and analyze production data and logs.

The JITC Integrated Test Strategy should be planned in coordination with the PMO and in alignment with other applicable JITC test guidebooks (see references).

3. ETI OVERVIEW

ETI is preferably initiated during the project initiation and/or design/technical development phases of the project's acquisition life cycle. This initiates JITC's risk-based test approach for DISA Information Technology (IT) projects and provides Program Managers and other decision makers with an early view of requirements gaps, test limitations, risks, and mitigation strategies that align with the eight Areas of Evaluation identified in the T&E Scorecard Guidebook.

The ETI process consists of three key T&E activities:

- Requirements/Technical Analysis
- Risk Assessment
- Strategy Development

The three ETI deliverables include:

- Initial T&E Risk Assessment
- Initial T&E Scorecard
- SAS

4. T&E SCORECARD OVERVIEW

The T&E Scorecard provides a standard means of reporting the T&E status of DISA IT projects throughout the acquisition life cycle. The Evaluation Framework (EF) enables JITC AOs to collaborate with the PMO to determine the evaluation areas of emphasis such as requirements/design analysis, technical verification, and operational validation of IT projects. It guides the scorecard assessment towards specific test data and provides a foundation for T&E assessments.

For a project using DSO, an overarching EF is developed for the project/system with subsequent EFs developed for each Minimum Viable Product (MVP)/Minimum Viable Capability Release (MVCR). As part of the early test engagement, JITC assists the PMO in ensuring an adequate and complete EF early in the life cycle. This postures the PMO for successful test cycles.

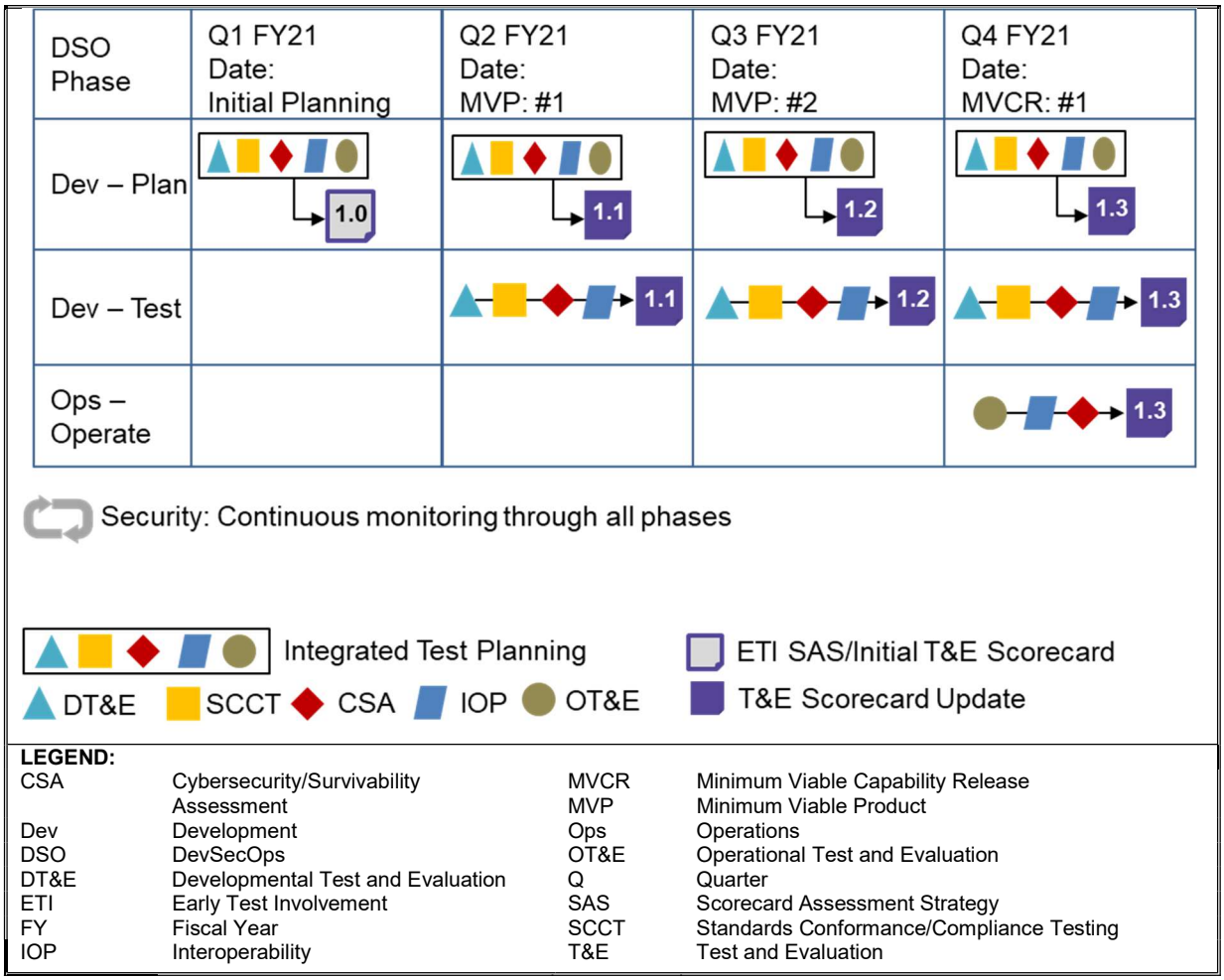
ETI Approach	AoE Risk Assessment	Scorecard Assessment Strategy Development	Scorecard Levels of Assessment
1. Requirements/ Technical Analysis 2. Risk Assessments 3. Strategy Development: Scorecard Assessment Strategy (SAS)	1. Capability 2. Interoperability 3. Availability 4. Capacity 5. Transition 6. Service Operations 7. User Experience 8. Cyber	1. Define what is being assessed 2. Define and analyze requirements 3. Identify/ Determine Area of Evaluation (AoE) risks 4. Identify high level test objectives based on identified risks 5. Define assessment approach 6. List relevant Test & Evaluation expectations 7. List required resources	1. Review (Requirements and Design Analysis) 2. Technical Verification 3. Operational Validation

Figure E-1. ETI and T&E Scorecard Summary

5. MVP/MVCR ROADMAP

JITC AOs should collaborate with the PMO to develop an MVP/MVCR Roadmap to track MVP/MVCR functionality-based milestones and T&E Scorecard updates. Creating a product roadmap helps the PMO communicate the direction and progress to internal teams and external stakeholders. It is a document showing the high-level functionality goals and initial prioritized listing of features supporting the functionality goals.

Updating and grooming the roadmap should be a continuous process throughout the life cycle to align emergent requirements with emergent program goals and prioritize them accordingly. JITC AOs should work closely with the PMO to periodically (as defined in the engagement agreement) identify any gaps in the program road map. The PMO and AO need to reach a common level of understanding relating to the level of effort needed to support each functionality milestone. The overall goal is to align the scorecard updates with MVP/MCVR milestones in the roadmap.



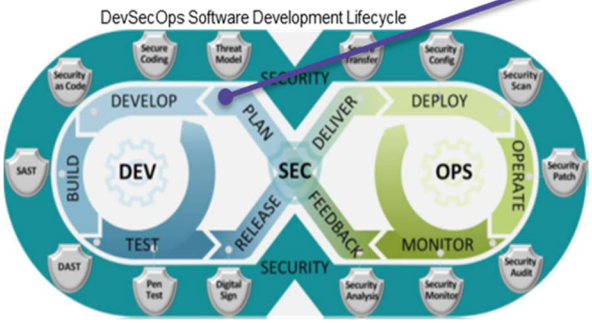
LEGEND:

CSA	Cybersecurity/Survivability Assessment	MVCR	Minimum Viable Capability Release
		MVP	Minimum Viable Product
Dev	Development	Ops	Operations
DSO	DevSecOps	OT&E	Operational Test and Evaluation
DT&E	Developmental Test and Evaluation	Q	Quarter
ETI	Early Test Involvement	SAS	Scorecard Assessment Strategy
FY	Fiscal Year	SCCT	Standards Conformance/Compliance Testing
IOP	Interoperability	T&E	Test and Evaluation

Figure E-2. MVP/MVCR Roadmap

ETI Approach for each MVP/MVCR is a collaborative effort between the AO, PMO, and Network Operations (NetOps) Integrators:

- Communicate requirements, strategies, and expectations early **(pre-planning)**
- Requirements/Technical Analysis: Analyze requirements and requirements gaps
- Risk Assessments: Assess test limitations, risk, and mitigation strategies
- Strategy Development: Establish Scorecard Assessment Strategy and utilize to generate agreement on the approach, cost, schedule, and performance expectations
- JITC PRODUCTS:
  - Initial Risk Assessment
  - Scorecard Assessment Strategy (SAS)
  - Initial/Updated Scorecard



LEGEND:

AO	Action Officer	MVP	Minimum Viable Product
DAST	Dynamic Application Security Testing	NetOps	Network Operations
Dev	Development	Ops	Operations
DSO	DevSecOps	PMO	Program Management Office
ETI	Early Test Involvement	SAS	Scorecard Assessment Strategy
JITC	Joint Interoperability Test Command	SAST	Static Application Security Tool
MVCR	Minimum Viable Capability Release	Sec	Security

Figure E-3. ETI in DSO

6. T&E SCORECARD TESTING IN DSO

The Review Assessment, Technical Verification, and Operational Validation should be executed for all T&E Scorecard assessments, for each MVP/MVCR.

INTENTIONALLY BLANK

APPENDIX F

PMO ROLES AND RESPONSIBILITIES

1. PROGRAM MANAGEMENT OFFICE

With the establishment of the Software Acquisition Pathway, adopting Agile/Development, Security, and Operations (DSO) practices for a Defense Acquisition Program requires an adjustment to the traditional roles and responsibilities Program Management Offices and Joint Interoperability Test Command Action Officers are familiar with under the traditional acquisition approaches.

1.1 Program Manager

The Program Manager is the capability developer and will:

- Develop and propose the acquisition strategy, timing, and scope of decision reviews, metrics, and required documentation to the Milestone Decision Authority.
- Work with the identified warfighter product owner to collaborate on a succinct capability needs statement that reflects the Minimum Viable Product (MVP) and Minimum Viable Capability Release (MVCR) as appropriate, along with a more substantial, but incremental delivery target.
- In conjunction with the Product Owner, develop a user agreement that identifies the desired level of user involvement and expectations for the collaborative method for evolving capability delivery timeliness.
- Ensure the team has a clear roadmap addressing the MVPs and MVCRs, as well as a definition of done.

Minimum Viable Product (MVP): An MVP is an early version of the software that has just enough working features to be useable to customers who provide feedback.

Minimum Viable Capability Release (MVCR): An MVCR is a set of features and/or capabilities suitable to be delivered to an operational environment.

1.2 Functional Lead/Product Owner

The Product Owner represents the user community and assists in the detailed capability needs statement development and user agreement during the DSO continuous planning phase. Throughout the life cycle of continuous development, integration, and testing, the Product Owner will act as the liaison to the user community and assist in prioritizing the backlog, performing periodic value assessments, and addressing user feedback on the developed solution.

INTENTIONALLY BLANK

APPENDIX G

TOOLS AND ACTIVITIES

1. DSO TOOLS AND ACTIVITIES

There are different tools and activities for each step of Development, Security, and Operations (DSO). Each program office will need to select the suitable tools for their system after processes, Key Performance Indicators, and activities have been defined. For detailed information on the standard tools and activities, see the "Department of Defense Enterprise DevSecOps Reference Design," Version 1.0. See Appendix C for the link to this document.

For Joint Interoperability Test Command (JITC) testing, the tools will need to be coordinated with the Program Management Office (PMO). The Action Officer (AO) will need to determine whether the program's or JITC's tools and/or pipeline will be used, or whether additional tools will be required. AOs that engage with programs early will have the opportunity to present options to the PMO that might otherwise be eliminated later in the process due to the complexity of adding tools and processes after throughput has started. It is not recommended that any DSO program change or modify their pipeline after it is in place and working, without first considering any bottlenecks and impacts to the pace of delivery.

Teams may also reference the JITC "Development, Security, and Operations (DevSecOps) Cybersecurity Test and Evaluation (Cyber T&E) Tools Requirement," Version 1.0, 31 July 2020.

2. SOFTWARE DEVELOPMENT PIPELINES/SOFTWARE FACTORIES

The DSO software factory and associated pipelines enable continuous integration/continuous delivery (CI/CD). See the DoD Enterprise DevSecOps Reference Design for the following definitions.

Software Factory: A software assembly plant that contains multiple pipelines, which are equipped with a set of 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. The software factory supports multi-tenancy.

CI/CD Pipeline: The set of tools and the associated process workflows to achieve continuous integration and continuous delivery with build, test, security, and release delivery activities, which are steered by a CI/CD orchestrator and automated as much as practice allows.

3. BLUE/GREEN DEPLOYMENT PROCESS

A blue/green deployment process will typically leverage infrastructure as code capabilities that are built upon containers (Kubernetes, Docker, OpenShift) to create and deploy to a new container (blue) in parallel to the in-production container (green). This new container is self-contained and independent from the original one. The new one (blue) inherits all the changes to be deployed and acts as a safe place to pre-test all the functionality, and/or slowly roll out changes to subsets of users, prior to promoting the container to production and deprecating the old container (green). Once the promotion is complete, the old one acts as a handy rollback option until the next deployment (and it can be backed up as a reference). When the next deployment is ready, the previous production container is deprecated, and the process repeats back and forth for every deployment. DSO programs benefit from this model in multiple ways:

1. The new deployment can be easily rolled back to the original state just by redirecting traffic back to the original one.
2. This allows for deployments to be implemented with more reliability (testing) and control (slow roll the deployment).
3. This enables clearer monitoring of the changes and the impact thereof.
4. This empowers the enhancement team (developers + systems + security) to configure, patch, apply Security Technical Implementation Guides (STIG), and deploy new functionality without large outages or maintenance windows.
5. Because of the above, it empowers deployment automation and the creation of multiple development effort software factories. Software factories are commonly used among many of the well-known technology giants in industry. They allow for multiple (upwards of thousands in some cases) teams to deploy to the primary asset (website/application or mobile application) independently without interdependence risk.

Pipeline management, auto-deployment, and container software help empower programs to automate the blue/green deployment process. Route-based deployments, which leverage different user access patterns, can be used to enable slower deployments when there is a higher risk of negative user impact. This strategy is recommended when large-scale security or systems changes are implemented because risk increases in tandem to the scale/size of the change. However, it may not be necessary for small deployments of functionality that have been thoroughly tested.

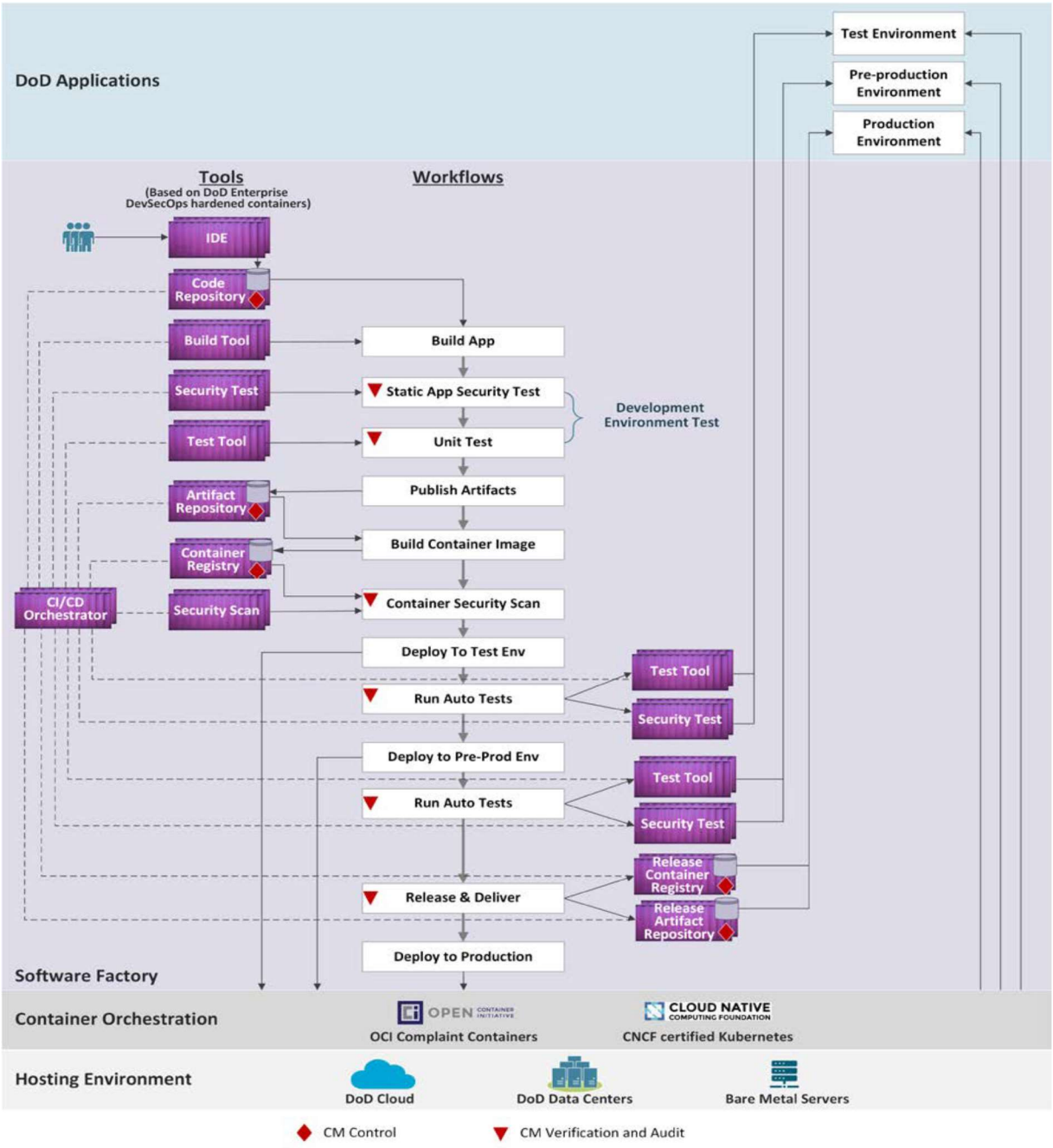
Teams should discuss where pipelines might be located and managed. If considering a test pipeline at JITC, the test team should discuss and consider the following:

- This solution would require the skill set of the embedded tester(s) role and the periodic manual tester role (Hybrid model).
- Determine what in-house skills are needed and what skills must be contracted and/or developed.
- Provide the program with subject matter expert-level advice on shifting all

configurations, designs, patches, and security settings left to the developer/enhancement team, which will minimize problems upstream. As a rule of thumb, the closer developer machines mirror production, the fewer problems will be experienced in deployment and testing upstream.

- Ensure programs are prepared for the overhead cost of running a DSO program and that they are aware of the benefits thereof. Benefits include cost avoidance (of failures later), return on investment beginning earlier, solution completeness, and user satisfaction.
- Pipelines and tools are great, but they are not DSO. DSO is a culture, a mentality, that has to be shared across the whole team(s). Without the culture, the tools will fail. AOs should advise newly adopting PMOs that the most important thing for them to focus on is Lead Time and Process Time. Aggressively pursuing the improvement of these two indicators will lead them to the right tools and the right processes. When choosing tools, PMOs and JITC should follow these simple rules:
    - Know and define the problem being solved. Do not anticipate or gold-plate for other problems. If the problem is not about Lead Time or Process Time, rethink it.
    - If it slows the team down, do not do it.
    - If it introduces complexity that isn't necessary, do not do it.
    - If it causes bottlenecks, fight it.
    - If it puts increased pressure on any process, resist it.

Figure G-1 shows a containerized software factory reference design.



NOTE: Source: DoD CIO, "DoD Enterprise DevSecOps Reference Design," Version 1.0, 12 August 2019

LEGEND:

App	Application	DoD	Department of Defense
CI/CD	Continuous Integration/Continuous Delivery	Env	Environment
CM	Configuration Management	IDE	Integrated Development Environment
CNCF	Cloud Native Computing Foundation	OCI	Oracle Cloud Infrastructure
DevSecOps	Development, Security, and Operations	Prod	Production

Figure G-1. Containerized Software Factory Reference Design

APPENDIX H

DSO INTEGRATED TEST STRATEGY DEVELOPMENT

Testing in Development, Security, and Operations (DevSecOps, or DSO) spans the entire software development, delivery, and operations life cycle. Testers should be involved early and ensure the definition of done is agreed upon, testable, and clear. This will ensure the validation of Program Management Office (PMO) testing and generation of additional testing needed to address the requirements.

Definition of Done: For testing in DSO, this is the minimum set of test criteria that must be met for each release. In the DSO environment, the definition of done does not just mean it is coded and tested; it means it will work in production.

Testers should be involved in the user story acceptance criteria definition as part of test-driven-development, functional testing, feature verification, operations testing, performance testing, security testing, as well as being able to monitor and analyze production data and logs.

JITC Test Strategy

AOs should plan the following integrated test strategy in coordination with the PMO and in alignment with other applicable Joint Interoperability Test Command (JITC) test guidebooks. Table H-1 contains an extract of test-related activities from the nine DSO phases in the "Department of Defense (DoD) Enterprise DevSecOps Reference Design," Version 1.0. It is provided as a starting point for JITC testers to assist them in their integration into the DSO life cycle. Note that the table is useful to the extent that the DSO product team is aligned with the "DoD Enterprise DevSecOps Reference Design."

Here are some notes on using the table:

- There are nine DSO phases in the table. (The DevSecOps Reference Design feedback phase is shown as inputs from other DevSecOps phase activities.)
- The first four columns are extracted directly from the DevSecOps Reference Design.
- Columns 1 and 2 are activities and associated descriptions from each DSO phase, respectively.
- Columns 3 and 4 are the inputs required to perform the activity and expected outputs from performing the activity, respectively.
- Column 5 suggests (when applicable) a JITC Test and Evaluation (T&E) activity to perform in conjunction with the DSO activity. The Cybersecurity T&E phase activities are based on the "JITC Cyber Test and Evaluation Guidebook," Version 1.02, September 2019. Note: DSO activities for JITC service areas will be updated as new information becomes available. Current

details focus largely on what is known for cybersecurity activities.

Note that all DSO decision making is performed inside the DSO life cycle based on inputs from the feedback phase. In the DoD, decision making is done by authorities outside the life cycle, and T&E activities inform these decisions. This is noted as a limitation in Table H-1.

See Figure I-3 for an example of tool types that can be used for each phase in Table H-1.

Table H-1. DSO Matrix

Activity	Description	Inputs	Outputs	JITC T&E Activities
DSO Continuous Planning				
Software requirement analysis	Gather the requirements from all stakeholders.	- Stakeholder inputs or feedback - Operation monitoring feedback - Test feedback	-Feature requirements -Performance requirements -Privacy requirements -Security requirements	Evaluate requirements for testability. Understand Cybersecurity Requirements (and plan for T&E). Check for Cyber Survivability Attributes (see the CJCS Cyber Survivability Endorsement Implementation Guide).
System design	Design the system based on the requirements.	Requirements documents	Documents: -System architecture -Functional design -Data flow diagrams -Test plan -Infrastructure configuration plan -Tool selections -Development tool -Test tool -Deployment platform	Begin Test Plan development as appropriate. Ensure the Test Plan includes adequate testing of the security requirements. NOTE: In the DSO paradigm, the Test Plan within the DSO life cycle is an ongoing, evolving Test Plan -- not a formal document. It is a derivative artifact from previous results and testing of new capabilities scheduled for the upcoming delivery. This will need to be determined in each case.
Threat Modeling	Identify potential threats, weaknesses, and vulnerabilities. Define the mitigation plan.	System Design	Potential Threats and Mitigation Plan	Characterize the Cyber-attack surface. Conduct Cyber Table Top exercise (Mission based cybersecurity risk assessment).
Project Planning	Project task management, Release planning		Task plan and schedule, Release plan and schedule	Ensure key test activities are in the IMS.
DSO Continuous Development				
Test development	Develop detailed test procedures, test scripts, test scenario configuration on the specific test tool.	Test plan	Test procedure document, Test data file, Test scripts	Test Procedure development IAW with Test Strategy or TEMP
DSO Continuous Integration (CI)				
Static application security test and scan	Perform SAST to the software system.	Source code, known vulnerabilities and weaknesses	Static code scan report and recommended mitigation	CVI. Blue team should participate in recommending mitigations or reviewing mitigations generated by the Tool.
Documentation	Detailed implementation documentation	User input, Source code	Documentation, Auto-generated Application Programming Interface (API) documentation	Review documentation and APIs for applicable standards and service agreements to inform tests and test planning.

Table H-1. DSO Matrix (continued)

Activity	Description	Inputs	Outputs	JITC T&E Activities
DSO Continuous Testing				
Unit test	Assist unit test script development and unit test execution. It is typically language specific.	Unit test script, individual software unit under test (a function, method, or an interface), test input data, and expected output data	Test report to determine whether the individual software unit performs as designed	Developmental testers are interested in adequacy of Unit Test and will review Test Plan/Results.
Dynamic application security test and scan	Perform DAST or IAST testing to the software system.	Running application and underlying OS; fuzz inputs	Vulnerability, static code weakness and/or dynamic code weakness report and recommended mitigation	Blue Team participation, continued
Integration test	Develop the integration test scripts and execute the scripts to test several software units as a group with the interaction between the units as the focus.	Integration test scripts, the software units under test, test input data, and expected output data	Test report about whether the integrated units performed as designed	Integration testing should be reviewed by IOP & OT testers. IOP should review for adequacy of the testing of interfaces and determination of the realism of the environment utilized. OT should review for developing test procedures, system familiarization, and collection of technical data.
System test	System test uses a set of tools to test the complete software system and its interaction with users or other external systems.	System test scripts, the software system and external dependencies, test input data, and expected output data	Test result about if the system performs as designed.	First opportunity for Integrated (DT/IOP/OT/SCCT) Testing.
Manual security test	This may include a penetration test, which uses a set of tools and procedures to evaluate the security of the system by injecting authorized simulated cyber-attacks to the system. CI/CD orchestrator does not automate the test, but the test results can be a control point in the pipeline.	Running application, underlying OS, and hosting environment	Vulnerability report and recommended mitigation**s**	DT Penetration Test (DTPT) Cooperative Penetration Assessment (CPA) Adversarial Cybersecurity DT&E (ACD) Adversarial Assessment (AA)
Performance test	Ensure applications will perform well under the expected workload. The test focus is on application response time, reliability, resource usage and scalability.	Test case, test data, and the software system	Performance metrics	Additional opportunity for Integrated (DT/IOP/OT/SCCT) Testing
Regression test	A type of software testing to confirm that a recent program or code change has not adversely affected existing features	Functional and non-functional regression test cases; the software system	Test report	Testers should monitor regression test reports/results to ensure changes do not invalidate prior reporting. Conduct verification and validation testing if Regression Test is done by a third party, as needed.

Table H-1. DSO Matrix (continued)

Activity	Description	Inputs	Outputs	JITC T&E Activities
DSO Continuous Testing (continued)				
Acceptance test	Conduct operational readiness test of the system. It generally includes the accessibility and usability test, failover and recovery test, performance, stress and volume test, security and penetration test, interoperability test, compatibility test, supportability, and maintainability.	The tested system, Supporting system, Test data	Test report	System Acceptance Testing OT testers should collect data for possible risk reduction to OT events (OA, UAT, etc.).
Container policy enforcement	Check developed containers to be sure they meet container policies.	Container, Policies in SCAP form	Container compliance report	Cooperative Vulnerability Assessment (CVA) (DT/OT) - A CVA can/should be conducted during the DT phase where Penetration and Adversarial events are scheduled. The CVA is preparatory analysis of Vulnerability based on system scans, Cyber and Security Controls and processes, procedures as related to RMF as required for achieving the ATO. - A CVA may or may not be necessary during OT if it was conducted during the DT phase. However, this depends on the outcome of the events conducted during DT, the program's/system's maturity level, plans, policies, and procedures across the program's spectrum (for example, development of PMP, CMP, COOP, IRP, SSP, SCG, etc.) Cyber Economic Vulnerability Assessment (CEVA) - For Financial Business Systems similar to CVA DTPT CPA
Compliance scan	Compliance audit	Artifacts, Software instances, System components	Compliance reports	CVA CEVA - The activity is the responsibility of the Program's Cybersecurity Personnel (ISSM/ISSO). - The CSAT will conduct analysis on compliance scans and reports during a Vulnerability Assessment. DTPT CPA The Red Team will run scans during a DTPT or CPA.

Table H-1. DSO Matrix (continued)

Activity	Description	Inputs	Outputs	JITC T&E Activities
DSO Continuous Testing (continued)				
Test audit	Test audit records who performs what test at what time and test results.	Test activity and test results	Test audit log	Testers review results. DTPT - Keep records during the event when scheduled. CPA - Keep records. AA - The Red Team will keep records of all of their activities and entries into the system tested.
Test deployment	Deploy application and set up testing environment using Infrastructure as Code.	Artifacts (application artifacts, test code), Infrastructure as Code	The environment ready to run tests	Testers participate in readiness review.
Database functional test	Perform unit test and functional test to database to verify the data definition, triggers, and constraints are implemented as expected.	Test data	Test results	Testers review results.
Database non-functional test	Conduct performance test, load test, and stress test. Conduct failover test	Test data, Test scenarios	Test results	Testers review results.
Database security test	Perform security scan, Security test.	Test data, Test scenarios	Test results	CVA CEVA ACD DTPT CPA - The activity in Column 1 is the responsibility of the Program's Cybersecurity Personnel (ISSM, ISSO). - The Red Team will run scans during a Penetration event.
Test configuration control and audit	Track test and security scan results. Generate action items. Make go/no-go decision to the next phase. (There may be several iterations for several tests across stages.)	Test results, Security scan and compliance scan report	Version controlled test results, Action items, Go/no-go decision	Testers review results and participate in readiness review. CVA CEVA - CSAT will conduct analysis of Configuration Management Policies and Procedures and scan logs and audit logs during a Vulnerability event. DTPT CPA - The Red Team will run security scans during a Penetration event.

Table H-1. DSO Matrix (continued)

Activity	Description	Inputs	Outputs	JITC T&E Activities
DSO Continuous Delivery (CD)				
Post-deployment security scan	System and infrastructure security scan	Access to system components and infrastructure components	Security vulnerability findings	Cybersecurity documentation and scan review during a CVA
Post-deployment checkout	Run test (automated when possible) to make sure the important functions of the system are working.	Smoke test scenarios and test scripts	Test results	Operational Assessment (OA) Risk Reduction Event
DSO Continuous Operations				
OT and Cyber Survivability Assessment	OT&E, Continued IOP and SCCT data collection, Cyber Survivability Assessment (CVPA and AA are subcomponents)	Data from prior test events	Operational Effectiveness, Suitability, and Cyber Survivability Determination and final report	Formal OT events including the Cyber Survivability Assessments are executed in the Operational/ Production environment. Collect additional IOP and SCCT data. JITC PRODUCT: Integrated test report (may contain): -Operational Test & Evaluation -T&E Scorecard -Joint Interoperability Certification/Assessment -SCCT Certification/Assessment -Quick Look brief
DSO Continuous Monitoring				
System performance monitoring	Monitor system hardware, software, database, and network performance; Baselining system performance; Detect anomalies	Running system	Performance KPI measures, Recommended actions, Warnings or alerts	Support continuous system monitoring and T&E reporting based on operational metrics/results.
System Security monitoring	Monitor security of all system components, Security vulnerability assessment, System security compliance scan	Running system	Vulnerabilities, Incompliance Findings, assessments and recommendations, Warnings and alerts	CVA CEVA The activity in Column 1 is the responsibility of the Program's Cybersecurity personnel (ISSM, ISSO). The CSAT will conduct analysis of these activities during a Vulnerability Assessment.

(The legend is on the next page.)

LEGEND:			
AA	Adversarial Assessment	IMS	Integrated Master Schedule
ACD	Adversarial Cybersecurity Developmental	IOP	Interoperability
API	Application Programming Interface	IRP	Incident Response Plan
ATO	Authority To Operate	ISSM	Information System Security Manager
CEVA	Cyber Economic Vulnerability Assessment	ISSO	Information System Security Officer
CI/CD	Continuous Integration/Continuous Delivery	JITC	Joint Interoperability Test Command
CJCS	Chairman of the Joint Chiefs of Staff	KPI	Key Performance Indicator
CMP	Configuration Management Plan	OA	Operational Assessment
COOP	Continuity of Operation Plan	OS	Operating System
CPA	Cooperative Penetration Assessment	OT	Operational Test
CS	Cybersecurity	OT&E	Operational Test and Evaluation
CSAT	Cybersecurity Assessment Team	PMP	Program Management Plan
CVA	Cooperative Vulnerability Assessment	RMF	Risk Management Framework
CVI	Cyber Vulnerability Investigation	SAST	Static Application Security Tool
CVPA	Cooperative Vulnerability and Penetration Assessment	SCAP	Security Content Automation Protocol
DAST	Dynamic Application Security Testing	SCCT	Standards Conformance/Compliance Test
DSO	Development, Security, Operations	SCG	Security Configuration Guide
DT	Developmental Test	SSP	System Security Plan
DTPT	DT Penetration Test	T&E	Test and Evaluation
IAST	Interactive Application Security Test	TEMP	Test and Evaluation Master Plan
IAW	In Accordance With	UAT	User Acceptance Test

APPENDIX I

ENVIRONMENTS AND ADDITIONAL TOOLS

1. DSO ENVIRONMENTS AND TOOLS

This section provides information on the general Development, Security, and Operations (DSO) environments and tools. The Joint Interoperability Test Command (JITC) Action Officer (AO) will need to work with the Program Manager to determine what environment (that is, Development, Test, Pre-production, or Production) is best for the test or tests which are being planned. The “Department of Defense (DoD) Enterprise Cloud Native Access Point Reference Design,” in development, will provide a high-level architecture and overview of the various environments, security zones, and planned access capabilities for DSO cloud-based implementations.

2. TEST ENVIRONMENTS

The test environment is primarily used for software builds and developmental testing. Typically, this testing is at the system level, although it may be at the system-of-systems level if the Program Management Office (PMO) owns multiple systems. It is normally the last opportunity for developmental testing prior to release to the operations team. The software that comes out of the test environment is the mark of quality from all the development and testing.

The test environment should represent the production environment as closely as possible. The only exception being that it may not connect to staging/pre-production or production environments of interfacing systems, but it may connect to test environments of interfacing systems. Testers should plan early to coordinate access to interfacing system test environments. These interfacing systems provide the basis for initial interoperability testing. When testing with these test environments, it is important to not only test the transmission and receipt of messages, but also test 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 the test environment. It is important to document the system, environment, and model or simulation configurations.

In DSO, the team should automate the test environment as much as possible for regression testing and testing of the routine and repeated human interfaces. The environments are set up using the software factory and should be orchestrated with scripts that include Infrastructure as Code (IaC) and Security as Code (SaC), which run on various tools.

3. STAGING/PRE-PRODUCTION ENVIRONMENTS

Much of the integration may need to be conducted in the staging/pre-production environment. In that case, some testing may also need to be conducted in this environment. The Operational Test Agency (OTA) may also conduct Operational Test and Evaluation (T&E) in the staging/pre-production environment in the Test Phase (such as risk reduction operational assessments).

4. OPERATIONAL/PRODUCTION ENVIRONMENTS

Deployment of a new software version to the operational/production environment occurs during the DSO Deploy Phase. This is also when real users and operators use the software in the operational/production environment to conduct their missions. The OTA also conducts formal operational testing in the operational/production environment as part of the DSO Operations Phase.

5. AUTOMATED TEST MANAGEMENT AND EXECUTION TOOLS

By automating manual processes and building tools into continuous integration/continuous delivery (CI/CD) pipelines, development and operations teams have increased workflow efficiencies and trust between groups, which is essential as these once-disparate teams now merge to tackle critical issues as a single new team. CI/CD gets rid of the manual gate and implements fully automated verification of the acceptance environment to determine whether the pipeline can continue on to production or not.

CI goes one step further than continuous build. It extends continuous build with more automated tests and security scans. Any test or security activities that require human intervention can be managed by separate process flows.

The automated tests include, but are not limited to, integration tests, a system test, and regression tests. The security scans include, but are not limited to, dynamic code analysis, test coverage, dependency/bill of materials checking, and compliance checking.

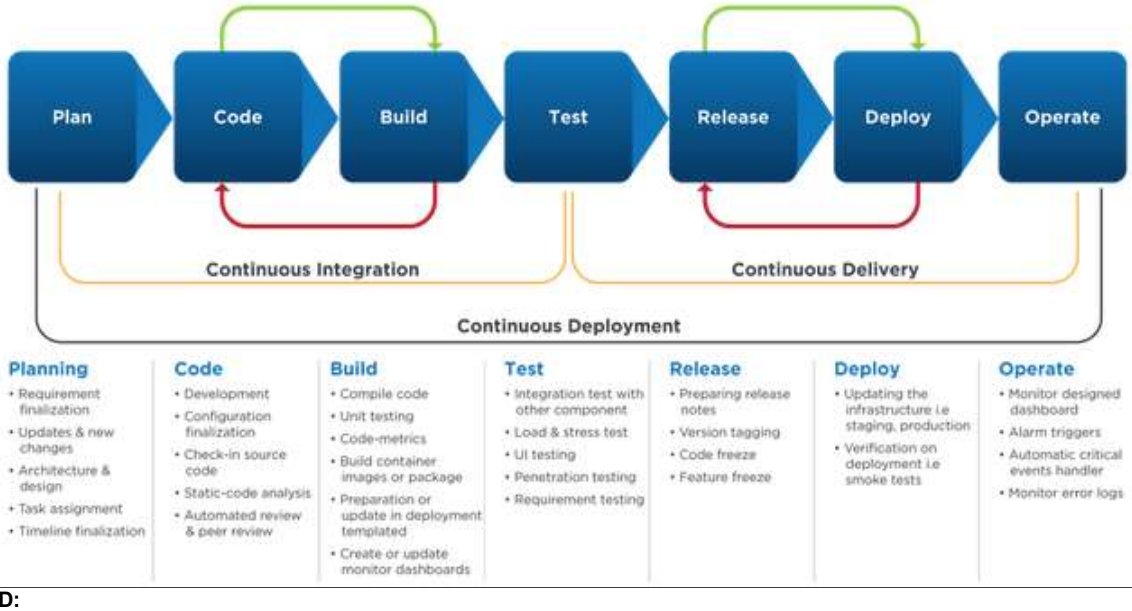
The outputs from CI include the continuous build outputs plus automated test results including security scan results. The trigger to the automated tests and security scan is a successful build.

CD is an extension of CI to ensure that a team can release the software changes to production quickly and in a sustainable way. The additional activities involved in CI include release control gate validation and storing the artifacts in the artifact repository, which may be different from the build artifact repository. The trigger to these additional activities is successful integration, which means all automated tests and security scans have been passed.

The human input from the manual test and security activities should be included in the release control gate. The outputs of CD are a release go/no-go decision and

released artifacts, if the decision is to release.

Figure I-1 shows the CI/CD in relation to the DSO phases.



LEGEND:

CD	Continuous Delivery	UI	User Interface
CI	Continuous Integration		

Figure I-1. CI/CD Pipeline

Based on JITC's experience with its internal software development and T&E pipeline, in support of external customers/programs seeking JITC's T&E support, JITC will make technical recommendations as to T&E tools, process, and techniques external customers should consider incorporating into their software development pipeline. JITC is also developing T&E tools for external customers, and the AO will need to determine for each system whether the PMO's or JITC's tools will be used or whether additional tools will be required. Customers may seek to use DoD Enterprise DSO offerings or build their own software factory. As more tools are being integrated in software factories, the AO will need to determine which tool(s) will best meet test requirements for each individual program.

There are various lists of tools published to include those available as part of the DoD Enterprise DSO Initiative:

- The Air Force Platform One has been designated as the first DoD Enterprise DSO Service provider. Platform One DSO-managed tools are available at https://software.af.mil/team/platformone/. Additional software factories are available at https://software.af.mil/software-factories/, including links to the following:
  - The Kessel Run software factory: https://software.af.mil/softwarefactory/kessel-run

- See the “Cybersecurity Test and Evaluation Guidebook,” Version 2.0, Change 1, 10 February 2020, for information on cybersecurity tools (https://www.dau.edu/cop/test/DAU%20Sponsored%20Documents/Cybersecurity-Test-and-Evaluation-Guidebook-Version2-change-1.pdf).
- Digital Artificial Intelligence (AI) (previously XebiaLabs) has created a Periodic Table of DevOps Tools, which is shown in Table I-2. The interactive version of this table is available at https://digital.ai/periodic-table-of-devops-tools. This can be a valuable reference tool when mapping a customer’s toolchain.

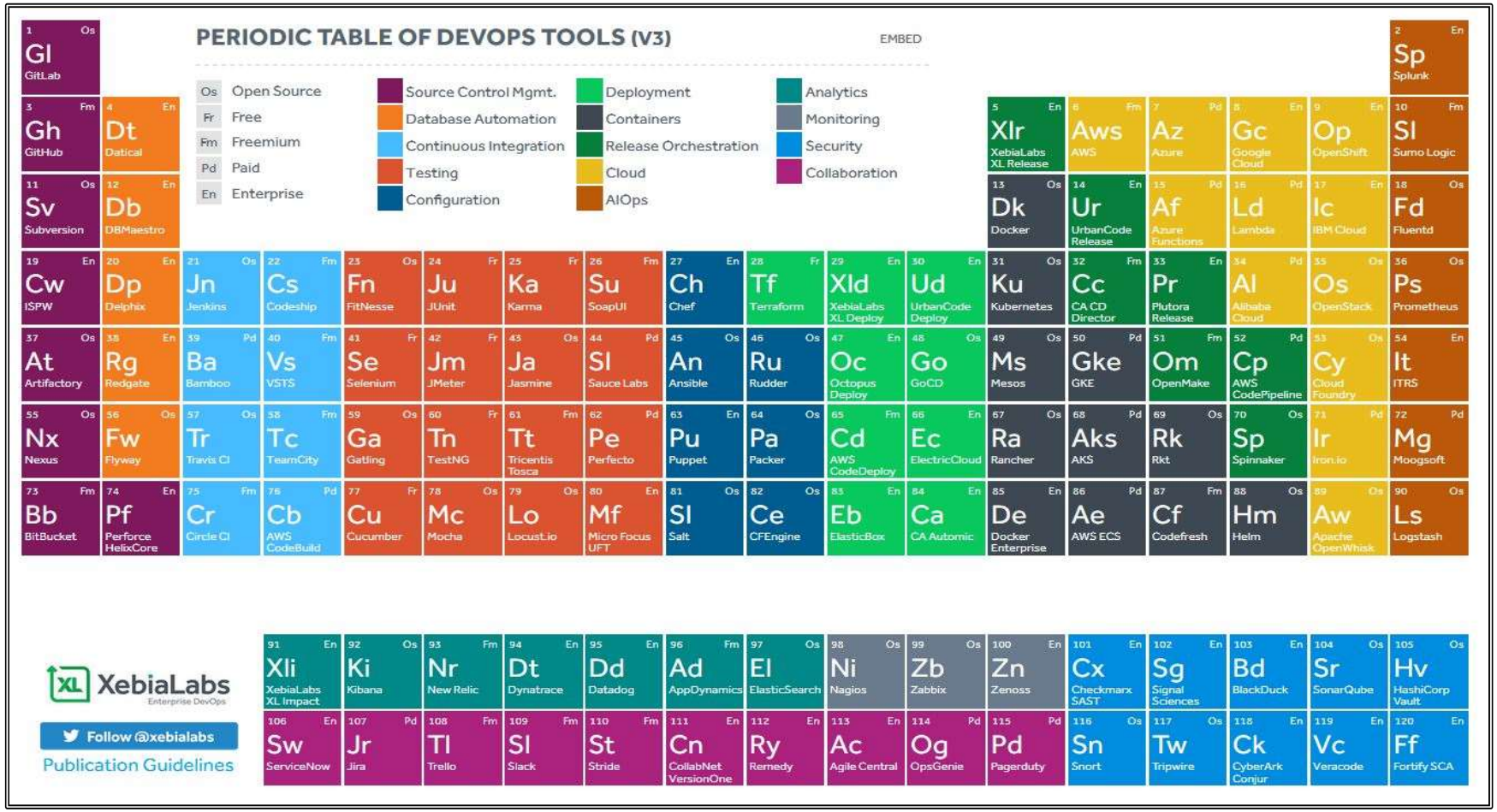
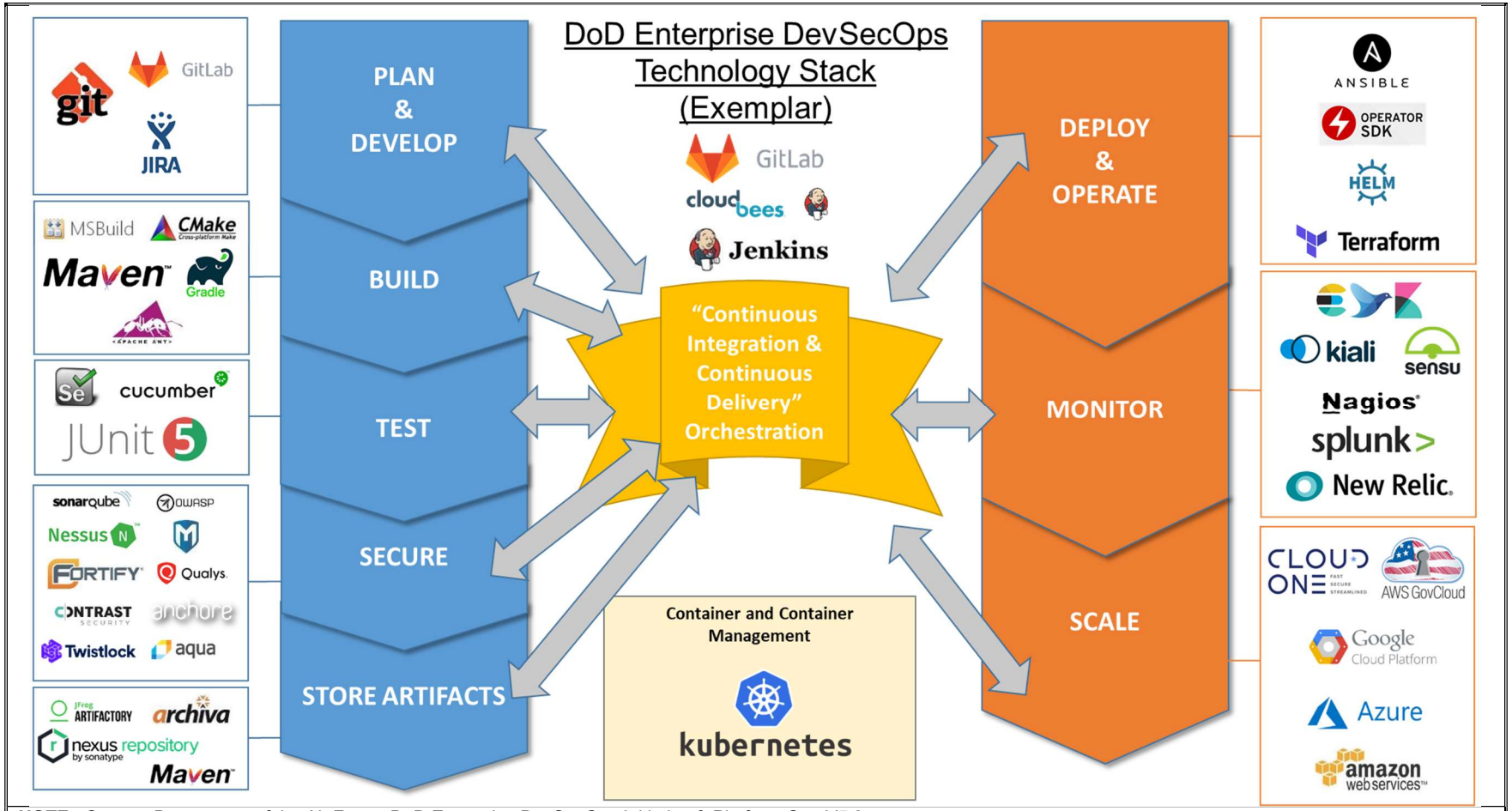


Figure I-2. Periodic Table of DevOps Tools

The Digital AI (previously XebiaLabs) interactive table contains a general explanation of tools and their role in the DSO process. Any specific tools listed in this section represent some of the more commonly used tools. Figure I-3 shows an example of a DSO technology stack and some of the tools as they relate to each phase of the DSO life cycle.



NOTE: Source: Department of the Air Force, DoD Enterprise DevSecOps Initiative & Platform One V5.2

LEGEND:

DevSecOps	Development, Security, Operations	DoD	Department of Defense

Figure I-3. DevSecOps Technology Stack