



CDAO

Systems Integration Test and Evaluation of Artificial Intelligence-Enabled Capabilities

What to Consider in a Test & Evaluation Strategy

April 2024

Table of Contents

This document provides a foundational overview for how leveraging Artificial Intelligence (AI) in DoD capabilities will influence Systems Integration (SI) Test and Evaluation (T&E) considerations in DoD T&E Strategies (TESs).

The SI T&E of AI-Enabled Capabilities (AIECs) is necessary to build justified confidence that an AIEC functions as a holistic unit.

01

T&E Strategies for AIECs

pp. 2–6

Provides an overview of the framework for the T&E of DoD AIECs and the role of this document within CDAO's T&E of AIECs Guidance and Best Practices product series.

02

SI T&E is important

pp. 7–12

Discusses the need to evaluate an AI component within its larger system to ensure that the AIEC functions as a holistic unit and document its limitations and risks.

03

Challenges of SI T&E of AIECs

pp. 13–20

Introduces five SI T&E activities and discusses their relevance to the T&E of DoD AIECs to provide non-SI experts with a high-level understanding.

04

SI T&E over the AIEC lifecycle

pp. 21–30

Highlights where SI T&E should be incorporated across the AIEC lifecycle, including during acquisition and sustainment.

05

Reflecting on SI T&E of AIECs

pp. 31–32

Summarizes recommended improvements for the SI T&E of AIECs, reflecting on the challenges posed by the inclusion of AI and how T&E changes vary over the AIEC lifecycle.



T&E Strategies for AIECs

This Section:

- + Provides an overview of the best practices for the test and evaluation of AI-enabled capabilities produced by CDAO Assessment and Assurance (A2); and
- + Specifies the role of the current document within the larger series of CDAO A2's T&E best practices document series.

01



This document is part of a framework for the T&E of AI-enabled capabilities

CDAO Assessment and Assurance is creating a framework to provide guidance on how to test and evaluate (T&E) AI-enabled capabilities (AIECs).

What is the framework?

The T&E of AIEC Framework provides best practices and guidance on how to test and evaluate AIEC.

The framework is organized into four focus areas of testing and provides different types of resources to AIEC developers and working-level testers.

Why is it needed?

The DoD community for the T&E of AIEC comes from a variety of backgrounds.

The T&E of AIEC Framework promotes a shared understanding between AIEC experts new to T&E and to T&E experts new to AIEC.

What is this document?

This document discusses what aspects of Systems Integration (SI) T&E to consider in a Test and Evaluation Strategy (TES) for an AIEC.

It is intended to help developers and working-level testers evaluate an AI component within a system to assure that the AIEC functions as a holistic unit.

This document provides:

- ✓ **Guidance and best practices**
- ✓ **A primer on SI T&E of AIECs**
- ✓ **Strategy-level T&E considerations**
- ✓ **T&E at the systems level**

This document does NOT provide:

- ✗ **Binding policy and requirements**
- ✗ **A comprehensive SI T&E guide**
- ✗ **Detailed T&E implementation**
- ✗ **T&E at the algorithm level**



CDAO's T&E of AIEC framework is organized into four T&E focus areas

While these T&E focus areas help break critical aspects of T&E into digestible pieces, they are neither mutually exclusive nor cleanly delineated in real testing.



Operational T&E (OT&E)

Evaluating an AIEC performing representative missions within an operationally realistic environment against a realistic adversary.



Human Systems Integration (HSI) T&E

Evaluating an AIEC's ability to help stakeholders observe and orient to their environment, make informed decisions, and carry out their missions.



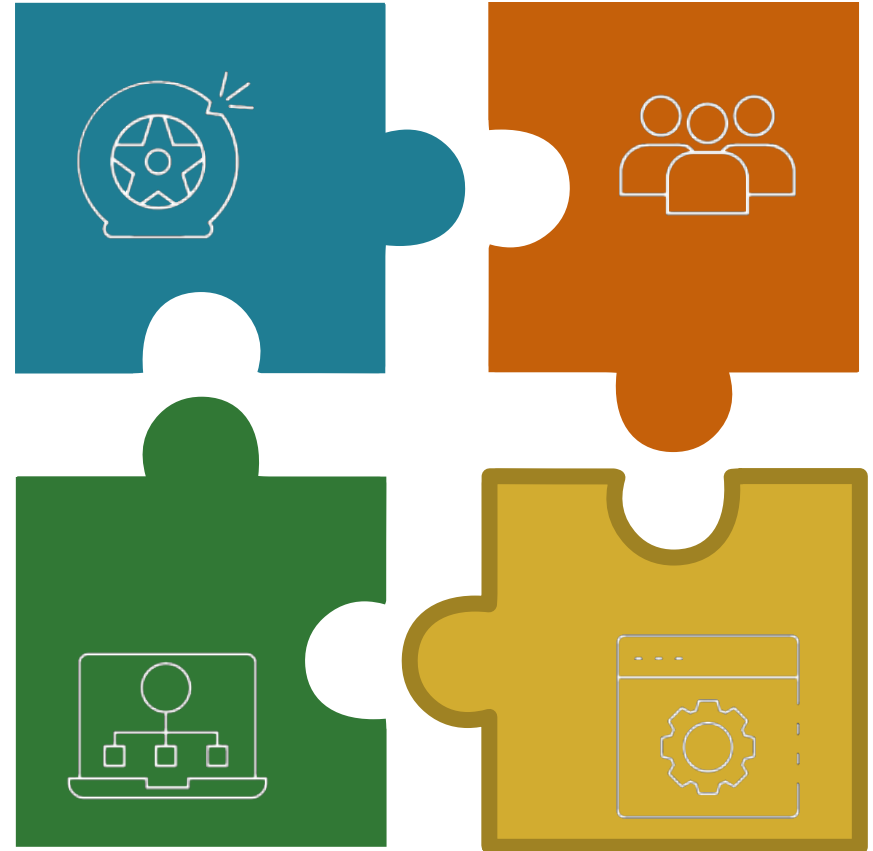
Systems Integration (SI) T&E

Evaluating an AI component within its larger system to ensure that the AIEC functions as a holistic unit and identify its limitations and risks.



AI Model T&E

Evaluating and documenting AI models and data across performance dimensions informed by system and mission constraints.



This document covers the SI T&E focus area



CDAO is developing a series of products that address critical T&E needs

Part 1 is designed to help testers understand core T&E concepts so that working-level testers can write and assess test and evaluation strategies for AI-enabled capabilities

This document focuses on Part 1



1 | Write and assess T&E Strategies

Provides a high-level overview of critical T&E concepts that will be influenced by the inclusion of AI models in the system under test.

Supports testers and developers as they write TESs and assess whether the TES is committed to the right evaluations.



2 | Write and assess Detailed Test Plans

Provides guidance for implementation of T&E concepts introduced in Part 1; highlights promising paths forward for unsolved challenges.

Supports testers and developers as they develop and implement detailed test plans that capture mission objectives.



3 | Engage with other DoD T&E stakeholders

Provides frameworks outlining how T&E is critical to fielding trustworthy AIECs across DoD acquisition pathways and mission applications.

Supports testers and developers as they advocate for policy and investments that address DoD T&E shortcomings.



4 | Execute tests and rigorously analyze results

Provides resources such as templates, validated measurement instruments, and automated analysis tools.

Supports testers and developers by streamlining and automating common T&E activities with tailorable tools.



What is a Test & Evaluation Strategy?

A high-level document in DoD acquisitions that guides test planning and execution.



Captures the mission(s) a capability is intended to perform and all hardware and interfacing systems in the test design.



Identifies and prioritizes T&E objectives to inform and justify data requirements necessary to support program decisions.



Specifies the resources required to conduct T&E and identifies shortfalls in resourcing that will require investments.



Describes the test activities necessary to evaluate the capability and inform acquisition, technical, and program decisions.



Learn More

You can read more about DoD TESs at
<https://www.test-evaluation.osd.mil/T-E-Enterprise-Guidebook/>



SI T&E is important

This Section:

- + Provides a brief overview of Systems Integration (SI) and why it is critical to do it well.
- + Describes the DoD ecosystem and the value of integration.

02



Systems integration is important!

“The integration process provides a framework to systematically assemble lower-level system elements into successively higher-level system elements, iterative with verification until the system itself emerges.”

See [DoD Systems Engineering Guidebook](#).

A well-integrated system can improve functionality with less additional overhead.

Systems integration is the structured approach to combine building block components, such as the relevant software and hardware, into functional and useful systems. This process can take many iterations.

Unlike standalone or federated systems, where the user may face the friction of shifting interfaces and data inconsistencies, a well-integrated system consolidates functionalities into a unified whole. The interconnectivity of an integrated system can avoid the redundancy and inefficiencies that often plague isolated systems, and lead to more streamlined operations and potentially lower costs.

Multiple standalone systems can increase the burden on people and platforms.

Standalone systems may require dedicated support to carry, power, and administer, resulting in an inefficient use of our warfighters' limited resources.

Manually deconflicting disparate systems adds another layer of complexity, increasing the risk of mishaps and reducing effectiveness.

Iterative systems integration is crucial for effective deployment of AI components.

Simultaneously integrating all system elements at once can be time consuming and error prone, and it can obscure the root cause of problems.

CDAO recommends iteratively integrating AI components into the larger system and testing the system to catch integration errors or deficiencies early and often.

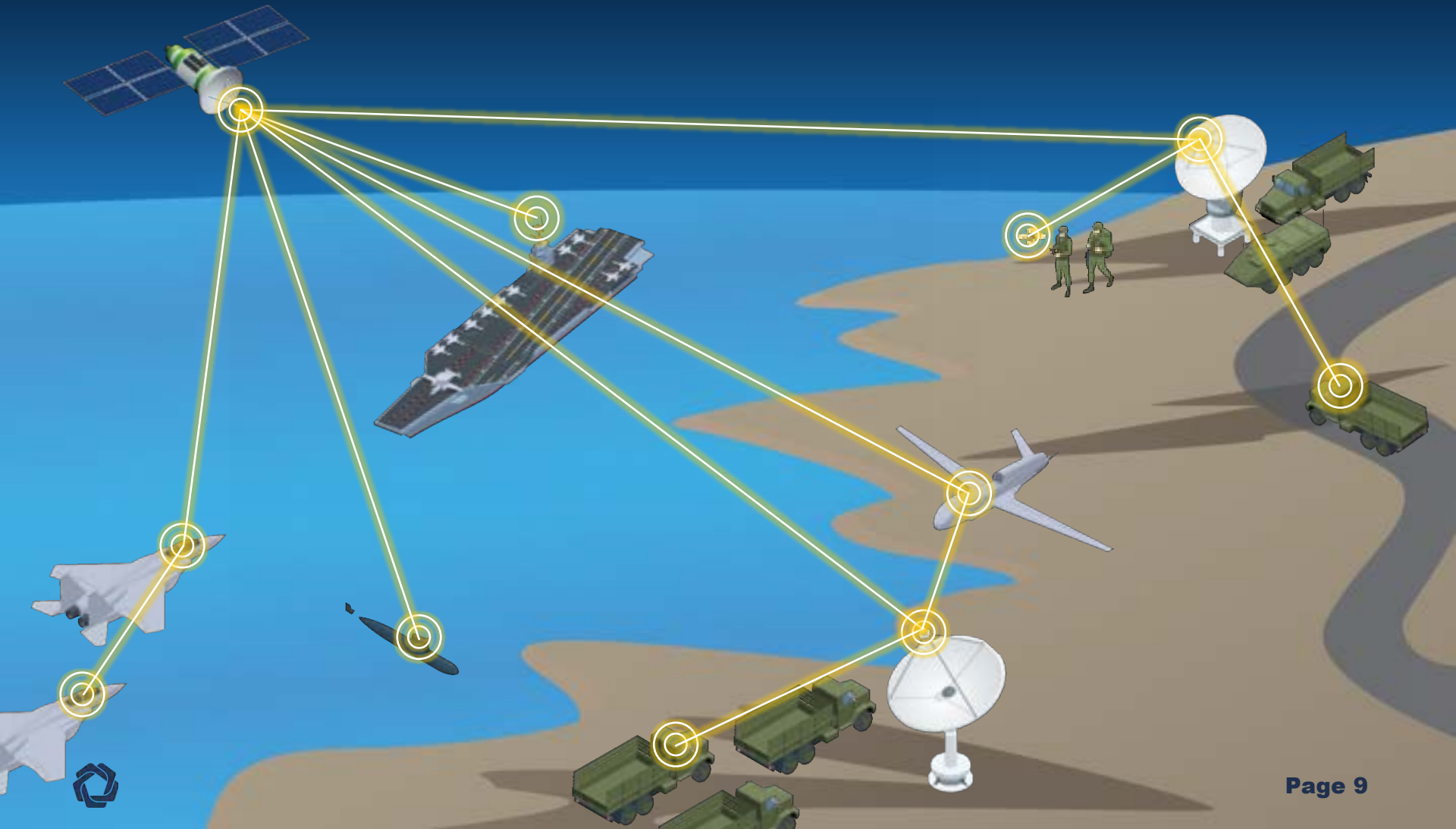
Example

Consider a smartphone app. It is designed to be integrated with the phone's operating system and the phone's hardware. This architecture allows the new app to take advantage of a Wi-Fi radio, a cellular radio, the processor's compute power, the touchscreen user interface, and the power supply.



DoD missions and systems are complex

Individual tools and widgets are not useful unless they fit into the broader mission context and vast ecosystem of hardware, software, and people.



The DoD integration ecosystem is very different from how it happens in industry

How to field an iOS app

How to field a DoD capability

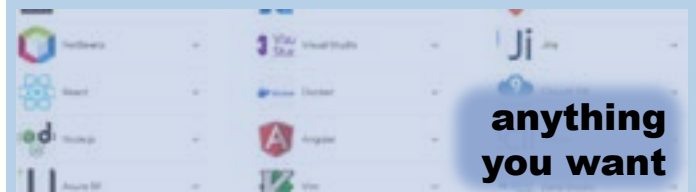
Build using ...



Build with Xcode 15

Deliver great experiences by seamlessly integrating with the latest iOS and iPadOS. Build your apps using the latest version of Xcode 15, which includes the latest iOS and iPadOS SDKs. Please note that as of April 2024 all iOS and iPadOS apps require a minimum of Xcode 15 and the iOS 17 SDK.

Apple's Tools



anything you want

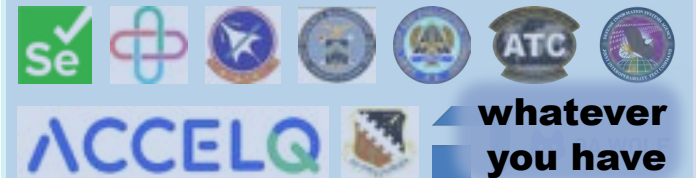
Test using ...



Test your apps

Make sure your apps work as expected on the latest released devices. All-screen support

Apple's Tools



whatever you have

Approved by...



Submit for review

Before submitting your app for review, make sure it's ready to be the most of your product page.

Apple



someone with authority

Field on...



devices sharing a operating system



disparate devices



Integrated systems leverage common information, scale, and data

“Integration is essential to increasing system maturity, reducing risk and preparing the system for transition ...” DoD Systems Engineering Guidebook

SI T&E must cover the complexity of DoD missions and AIEC use cases.

For large systems of systems, SI T&E can be a daunting effort. The scope can be kept manageable by designing and implementing robust and modular architectures, maintaining configuration management, testing subsystems in parallel, and leveraging efficient test designs and automated testing.

Below are some examples of what types of software and hardware AI components will need to successfully integrate with:

- User interfaces
- Datalinks
- Power supplies
- Computing resources
- Databases
- Operating systems
- Legacy software & hardware

Integration AI components can leverage SI principles used for agile software development.

All AIECs are types of software-intensive systems. In this way, the AI Model T&E Framework is analogous to component testing for traditional hardware or software elements. However, while AI models can be considered a type of software, testers should be aware that they introduce new T&E challenges or exacerbate existing ones.

In addition to SI considerations for software, integrating an AI component into a system requires additional care to understand how system performance is impacted by model and data drift, AI real-time learning, or interactions with new software components, hardware, or environments.

After performing sufficiently in model tests, the AI component is ready for integration.

The integration process is where the AI component is combined with its relevant hardware and software elements. SI T&E objectives include:

- Characterize the performance of the hardware, software, and AI component.
- Identify and document changes in model performance between integration iterations.
- Provide objective evidence that the system fulfills the requirements.
- Use the test results to create and maintain a system assurance case.



Systems Integration T&E is an iterative process conducted over three phases

When an AI component seems to be performing acceptably in standalone tests, it may be ready for integration into the larger system.

Integration T&E Prep

Before integration T&E activities, the TES should be current, system elements under test should be stable, and the AI component should have undergone standalone AI model T&E.

During “T&E prep,” testers should identify system elements and interfaces, configure the test environment, and integrate the AI component into the test environment.

Integration T&E Execution

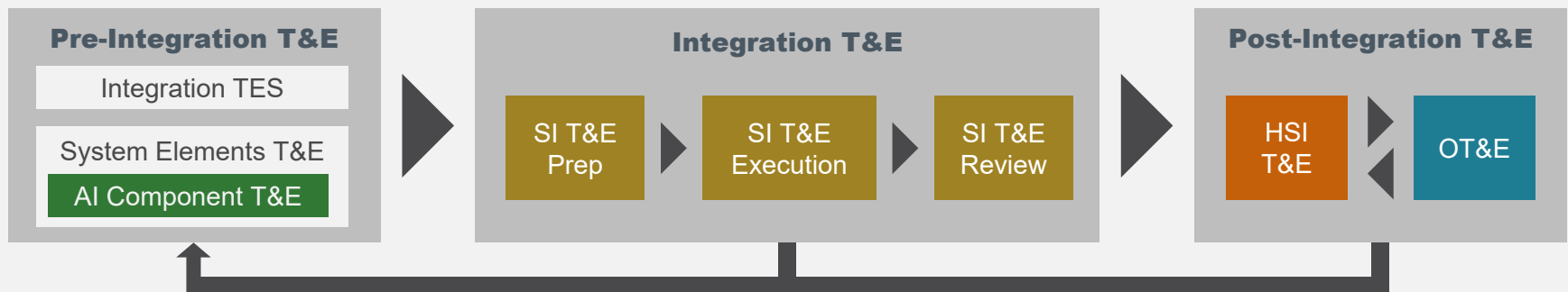
Perform the systems integration test in accordance with the TES to include functionality, reliability, security, compatibility, and interoperability test activities. Identify opportunities to combine test activities when appropriate.

Collect test data from each T&E activity and document any anomalies.

Integration T&E Review

During “T&E review,” testers should analyze their test results and update system documentation and the TES where appropriate.

Analysis may reveal undesirable behaviors that require the AI component to be retrained. If the system performs well, consider integrating humans and conducting HSI T&E and OT&E.



Important!

This process will be iterative, and for complex systems, nested as components are added to the system and tested. No step will happen just once.



Challenges of SI T&E of AIECs

This Section:

+ Introduces five Systems Integration T&E activities.

03



Systems Integration T&E Activities

T&E activities provide objective evidence and complement other non-test activities to build a body of evidence that the AIEC fulfills its requirements.



Functionality

The ability of the system to do the work for which it was intended.



Reliability

The probability that a system successfully performs a function under stated conditions for a stated period of time.



Security

The ability to identify and mitigate a system's vulnerabilities and weaknesses.



Compatibility

How well two or more system components interact in the same environment.



Interoperability

The ability of system components to connect and communicate with one another readily, even if they were developed by different manufacturers in different industries.

Important!

Identify opportunities to combine test activities when appropriate. For example, a single activity test could include the interoperability and performance of a sensor.

Important!

The SI T&E activities provided in this section are a great start, but this list is not exhaustive! Future updates to this document will better integrate the SI T&E considerations in DoDI 5000.89 and the NIST AI Risk Management Framework (e.g., Safety, Maintainability).



How to use this section

Each SI T&E activity is presented in a “one-pager.”

This framework identifies 5 different SI T&E activities that should be included in a TES.

Use this section to write or review a TES so that it includes core SI T&E activities relevant to AIECs.

How should I use this section?

Identify core concepts: In this product, we identify the critical SI activities to consider when testing and evaluating AIECs.

Find “Google-able” terms: For each concept, the one-pager includes its more formal name and definition. Beyond being informative, this provides the keywords needed to find the supplemental literature online.

Learn to interpret informal language: Because most TESs will not have input from AI experts, one-pagers provide overviews and AI-specific concerns so that testers can identify if the TES has included relevant SI concepts with different, informal language.

Understand the need to test: We explain how each SI T&E activity can either empower or undermine the effective, safe, or ethical employment of these novel systems.

What are the limitations of this section?

It is not an exhaustive product.

While the core SI T&E activities included in this product highlight key concepts that testers should focus on, please be aware that this list is not complete. While more nuanced concepts and implementation guidance will be discussed in future “guidebooks” and “deep dives,” no product in this series exhaustively lists all SI T&E activities.

Additionally, these summaries are limited to a single page, but in reality, most of these concepts span entire research communities.

Some SI T&E activities will be more important than others for a given AIEC application.

Every TES may not emphasize the 5 SI T&E activities in this framework equally. Some will have to prioritize resources, and some activities may be less relevant for some systems.





Functionality

The ability of the system to do the work for which it was intended.

What are SI T&E concerns?

Functionality testing provides objective evidence that system components meet the specified requirements and satisfy applicable standards, and regulations. Functionality testing begins at the component level. For an AI component, this maps to the testing guidance provided in CDAO’s “T&E of AI Models” framework.

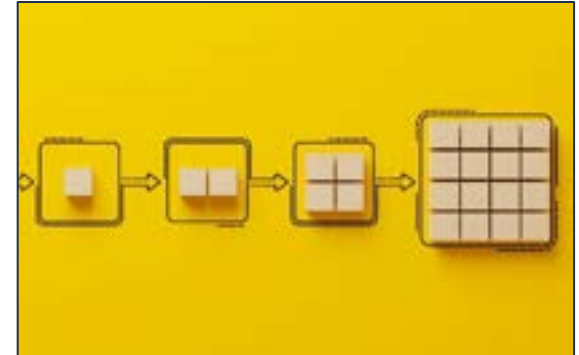
When system components are integrated together, the interaction between components may result in new, unanticipated behaviors not produced by any single component in isolation. This phenomenon is known as intra-system emergence. Jumping from element-level testing to full system testing will make it difficult, if not impossible, to identify the source of emergent behaviors. Incrementally integrating and testing the system is necessary to catch, troubleshoot, and mitigate undesired emergent behaviors.

How can AI make it harder?

While all complex systems will produce unexpected, emergent behaviors, the probabilistic and autonomous nature of many AIECs can make these systems particularly susceptible. Additionally, AI components often fail to generalize to novel inputs, making characterizing system performance.

Test design for SI should consider implicit parameters that are not explicitly labelled in the AI component’s training data, such as changes to other system hardware and software.

Beyond exacerbating concerns about undesirable emergent behaviors, AIECs are often tasked with work traditionally performed by human warfighters, for which there may not be established evaluation standards. Determining what constitutes adequate performance can be difficult, especially in scenarios where the task outcomes are not easily quantifiable.



What should testers do?

- ! For each level of integration, identify metrics and criteria that are relevant and at an appropriate fidelity.
- ! Use test sampling methods to identify edge cases that might give rise to emergent behaviors.
- ! Collaborate on the design of system instrumentation to assure data is sufficient to interpret test results.
- ! When needed, retrain the AI component with data from the integrated system.





Check out the “*DoD Guide for Achieving Reliability, Availability, and Maintainability*” to learn more about reliability in the DoD.

Reliability

The probability that a system successfully performs a function under stated conditions for a stated period of time.

What are SI T&E concerns?

Reliability testing provides objective evidence that system components meet the specified reliability requirements, and satisfy applicable standards, and regulations.

Multiple components introduce various failure modes, complicating the diagnosis and rectification of reliability issues. Small errors can propagate and magnify, resulting in significant impacts on overall system reliability.

How can AI make it harder?

In the same way that emergence and overfitting can make it difficult to characterize system performance, it can be difficult, if not impossible, to evaluate which operational conditions are likely to cause failures.

AI components will not “wear out” over time like physical components. AI component performance may change over time though in ways that degrade performance, either through explicit learning or changes in operational inputs. Input change over time is known as drift.

For AIECs, traditional tools like reliability growth curves and defect elimination trackers may not be sufficient to characterize the reliability of a system.



What should testers do?

- ! Use test sampling methods to identify edge cases that might give rise to emergent behaviors.
- ! Research how your AI components may introduce novel failure mechanisms and account for them in the test design.





Check out the CDAO A2's "*Red Teaming Handbook*" and "Adversarial Machine Learning Literature Review " to learn more.

Security

The ability to identify and mitigate a system's vulnerabilities and weaknesses.

What are SI T&E concerns?

Security testing assesses how vulnerabilities impact performance and provides evidence that security requirements have been met. While security testing is critical throughout development, SI introduces unique challenges.

Understanding and securing all interactions, interdependencies, and data flows between components is more difficult compared with simpler, standalone systems. Variations in security levels across different components can create weak links, providing easier opportunities for attackers to exploit these inconsistencies. A vulnerability in one component can lead to cascading failures, compromising the entire system. Component interactions may also produce unpredictable, emergent behaviors that result in vulnerabilities would not be present in any single isolated component.

How can AI make it harder?

Techniques that attack and defend against capabilities enabled by AI by exploit the data-driven nature of AI are commonly referred to as "Adversarial AI." Our taxonomy divides attacks into three categories – poisoning, evasion, and privacy – based on the goals of the attack and when they occur during the lifecycle of the AIEC.

- Poisoning attacks attempt to introduce a vulnerability into the model while it is being developed.
- Evasion attacks occur during the fielded operation of the system, with the introduction of inputs designed to subvert the intended functioning of the model.
- Privacy attacks attempt to extract information from a fielded model.



What should testers do?

- ! Develop a system threat model to identify potential threats posed by adversaries.
- ! Develop test cases to verify the security requirements and determine if the system vulnerabilities have been sufficiently mitigated.
- ! Incorporate automated testing to identify new vulnerabilities and assess their impact on mission performance.



Compatibility

How well two or more system components interact in the same environment.

What are SI T&E concerns?

At a component level, compatibility testing provides evidence that a system component can coexist and successfully function with other system components. At a system level, compatibility testing should demonstrate how the overall system functions in an environment that contains other systems.

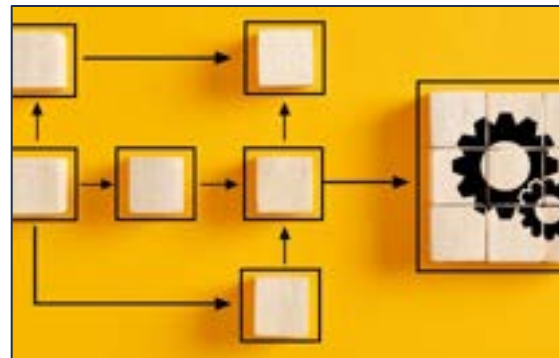
Compatibility testing also considers the system can successfully function in the specified operating environment. The goal of environment compatibility testing is to assess if an AIEC can function consistently across various operating system and deployment environments (e.g., cloud, on-premise, and edge device deployments.)

How can AI make it harder?

As with other types of software system compatibility, AIECs should also be focused on compatibility related to new version updates, hardware changes, operating system changes, and other system component changes.

Similar to functionality testing at higher levels of integration, interoperability and compatibility testing may uncover gaps in an AI component's training data. This could require retraining and revisiting model testing and functional testing to achieve desired interoperability and compatibility. The more complex the system-of-systems environment becomes, the more likely the initial training data will not be sufficient.

A new system may interact negatively with an existing system, reducing the performance of one or both. Compatibility between multiple AIECs may require training them together.



What should testers do?

Assess how the following impact mission performance:

- ! Hardware specifications, security policies, and network architecture.
- ! Virtualization, scalability, and network latency.
- ! Resource and connection constraints (e.g., limited memory, processing power, or bandwidth)
- ! Updates of software, hardware, operating system, and other system components.



Interoperability

System elements should connect and communicate with one another readily, even if they were developed by different manufacturers in different industries.

What are SI T&E concerns?

Interoperability testing goes beyond compatibility testing, and considers the interactions between system components within a system as well as interactions with external systems.

Testing objectives should include verifying independent functionality and seamless communication. Independent functionality means that each system component can perform its allocated tasks independently. For seamless communication, ensure that system components communicate as expected without compromising their individual functionality.

How can AI make it harder?

AI introduces a layer of complexity to interoperability testing. More than traditional software, AI systems exhibit dynamic behavior, and interoperability testing must account for this dynamic nature. Ensuring consistent behavior across different contexts is challenging.

Many AI components heavily rely on data. Their performance hinges on the quality, diversity, and relevance of training data. The output of AI components will shift as new versions are trained with updated data or improved algorithms. Ensuring compatibility between different component versions is crucial.

Because some AI systems operate as “black boxes” there can be challenges in understanding their processes. Interoperability testing must consider how other systems interact with these black-box components. Can they interpret AI outputs correctly?



What should testers do?

- ! Test how the AI component adjusts its behavior when integrated with other system components and systems.
- ! Consider how the AI component handles variations in input data and unexpected inputs from other system components and other systems.
- ! Test how updates to the AI component impacts overall system behavior and mission performance.



SI T&E over the AIEC lifecycle

This Section:

- + Introduces considerations to shift left for Systems Integration T&E.
- + Introduces considerations to shift right for Systems Integration T&E.

04



We must continue to “shift left”

An ounce of prevention is worth a pound of cure.

Integrating a AI component into a system of systems, especially in an environment with diverse components from different developers and under constraints like strict security or bandwidth limitations, requires careful coordination, planning, and execution.

AI components are complex, and their inclusion in any system creates a need for additional conversations between all stakeholders as they work toward a shared understanding of requirements, capabilities, and limitations.

Early-stage testing and assurance activities are crucial for identifying and mitigating integration risks.

Testers should focus on SI considerations throughout development, continuously collecting objective evidence about an AIEC’s interactions and behaviors to build the case for appropriately calibrated trust.



Shift Left



DoD “Valley of Death”

A common plight of DoD technology development efforts, where they fail to transition from the research and development phase into fielded, operational capabilities.

Collaborate with stakeholders to understand objectives and constraints

AIECs blur the line between development and testing, giving testers equity in design.

Integrating an AI component into a system requires compatibility with existing infrastructure and system components. Common challenges, such as proprietary restrictions and integrations across classification levels, are often exacerbated when components are developed by multiple stakeholders.

Testing must be done early and often to identify and correct integration flaws and bottlenecks. Testers should work with stakeholders to answer the following early in the AIEC’s lifecycle.

- What access will testers have to an AI component and its training data?
- How is the AI component and its training data documented?
- Will the system architecture allow for ongoing automated testing?

Cross-functional collaboration is the only way to overcome the “valley of death.”

While the development of all DoD systems requires coordination across stakeholders, including an AI component can increase the complexity of the communication. The integration of an AI component necessitates a closer collaboration between data scientists, AI engineers, software developers, and domain experts to address the unique challenges of an AI component, from data preparation to model deployment and monitoring. In cases when an AI component is embedded within a system of systems, the responsibility for updating and maintaining the AI component must be clearly delegated.

Collaboration provides a shared understanding of AIEC requirements, design features, anticipated risks, and performance expectations to inform test design and prioritization.



Shift Left



IP Strategy

A required artifact in the DoD's Software Acquisition Pathway that identifies and describes the licensing rights for all software and related materials necessary to meet a capability's operational, cybersecurity, and supportability requirements. There are similar requirements for the other acquisition pathways.

Understand how training data impact an AIEC and attain appropriate data access

The data-driven nature of AIECs makes the system performance very sensitive to data quality.

AIECs are often preferred over traditional technologies for their ability to rapidly identify trends across vast quantities of data that would be overwhelming or nonintuitive to humans. However, the advantage of AIECs—their ability to pick up on subtle trends and features in data—can also make it challenging to identify appropriate test factors and edge cases. Furthermore, an AIEC may appear misleadingly effective during testing if the same data are used to both train and test the model.

Testing an AI component within a system must consider data flows and interactions between system components. Testers must work with program stakeholders early to establish clear agreements on data usage, access, and security.

Testers need appropriate access to relevant, high-quality data to inform testing.

Beyond just tracking and managing code, version control for AI components must track datasets, model parameters, and training environments. Tracking datasets is a distinct difference from traditional software documentation. Datasets often are not static, and tracking training data evolution is crucial to ensure that the model remains valid and accurate.

Leveraging an AI component can obscure the causal relationships between a system's inputs and its performance. When integrating an AI component within a large system, clear documentation and version control is needed for testers to identify appropriate test factors and edge cases and interpret test results. This is especially true if the testers were not involved in the AI component's development.



Shift Left

Advocate for T&E needs in the design and resourcing of AIEC system interfaces

Design of system instrumentation should account for T&E needs.

System instrumentation is critical for testers to access the data needed to inform their evaluations. Beyond recording system inputs and outputs, “cognitive instrumentation” that captures internal system state data will help testers understand how and why an AIEC succeeds and fails. While adding data collection hardware later may be feasible, post hoc solutions often do not allow for the necessary data fidelity or structure.

Testers should advocate for instrumentation design that addresses their data requirements; however, there is no free lunch. More complex instrumentation will increase the networking bandwidth and computing needs for the system. Testers should confirm that testing instrumentation is balanced with the resource demands from other system functions.

Maintaining consistent interfaces in an AIEC is crucial for system performance.

Standardized system interfaces can enable consistent system performance and streamline communication between components. However, due to DoD security constraints, many developers design custom system interfaces and pipelines after struggling to access DoD digital infrastructure and tools common in industry AI development. Converging on a shared interface from standalone custom interfaces requires significant time and effort to bring all the system components in alignment.

Interfaces and system architecture cannot be easily modified late in development. Testers should work with programs to establish clear, consistent data structures and system interfaces that align with testing objectives and enable automated testing for issues like bias, drift, and data corruption.



Cognitive Instrumentation

Built-in infrastructure for recording the internal states of system components (e.g., inputs and outputs at each piece of a system’s modular design).



System Interface

The protocols and pathways for data exchange between system components.



Shift Left



Data Card

A concise, structured document that summarizes the acquisition of the data, the dataset distribution, and any data processing or transformations.



Model Card

A concise, structured document that summarizes the AI component's purpose, key predictors, performance metrics, and data sources.

Build transparency and version control into the documentation to benefit testing

Document AI components and training data to inform SI testing across stakeholders.

While AI components are touted as being exclusively “data-driven,” the reality is more nuanced. AI developers rely on their expertise to make many assumptions and decisions, such as:

- What data are used to train the AI?
- How should the data be processed?
- What does it mean to “succeed”?

Documentation of design decisions and assumptions, training data, and use cases informs test designs, ensures transparency across stakeholders, and facilitates troubleshooting and audits.

Data cards and models can be used to summarize key information about the AI component; additional documentation should provide the model architecture, hyperparameters, training process, evaluation reports, and a plan for ongoing maintenance.

Documentation and version control are critical for collaboration & reproducibility.

Beyond just tracking and managing code, version control for AI components must track versions of data, model parameters, and training environments. Tracking data is one of the more distinct differences from traditional software documentation. Datasets are often not static, and tracking training data evolution is crucial to ensure that an AI component remains valid and accurate.

Leveraging an AI component can also obscure the causal relationships between a system's inputs and its performance. When integrating an AI component within a system, clear documentation is needed for testers to identify appropriate test factors and edge cases and interpret test results. This especially true if the testers were not involved in the AI component development.



Shift Left

SI testing should happen early and often to identify errors and bottlenecks



Tech Debt

The technical debt or potential future costs incurred due to shortcuts or expedient decisions made during development, which may need to be corrected later at a significantly greater cost.



Edge Device

Hardware that processes, collects, and/or analyzes data at the "edge" of a network rather than fully relying on a centralized hub (e.g., wearable technology and autonomous vehicles.)

Integrate your AI component incrementally to buy down system "tech debt."

Over the past decade, the increasing complexity of software has pushed testing earlier in development to mitigate "tech debt." Leveraging AI can exacerbate tech debt, as correcting an AI component trained on poor quality or inappropriate data may be exceedingly challenging, if not impossible.

Testing AI components early enables early identification of issues and minimizes the risk of major problems arising from component interactions. It allows for the adjustment of AI components and system infrastructure in manageable phases, ensuring compatibility and performance at each step. Incremental testing not only makes problem-solving more efficient but also enhances the system's reliability and functionality, leading to a more robust final product.

Use scalability testing to stress test the AI component and identify bottlenecks.

Scalability testing assesses how well an AI component can handle increased data volumes or concurrent requests. Ensuring that the AI component performs efficiently and scales appropriately is crucial for its successful integration into the system of systems, especially in edge devices where computational resources, power, and/or network bandwidth may be severely limited.

While robust monitoring and logging are crucial, it's essential to balance these needs with bandwidth and security constraints. Efficient data compression techniques can minimize bandwidth usage, and sensitive data should be encrypted to maintain confidentiality and integrity.



We must continue to “shift right”

All AIECs are software-intensive systems that receive ongoing updates. Consequently, their capabilities, functionalities, and interactions with other system components evolve over time. These modifications can either enhance or diminish performance.

Post-fielding, additional testing of AIECs might be necessary due to "performance drift" caused by alterations in the AI component or training data. Performance may also be affected by changes in integrated components, such as new sensors, model updates, or environmental shifts.

When deploying an AIEC, testing every function in all possible environments is impractical, as it is with any complex software system. Instead, the DoD should monitor AIEC performance in the field to ensure the AIEC operates as expected. Ongoing data collection is essential, and project managers should share these data with testers.

Additional planning and resourcing will be necessary to implement monitoring tools, establish automated and ongoing data collection, develop sharing protocols to facilitate easy and secure data sharing between program managers and testers, and create a feedback loop for improvement.

T&E cannot stop at deployment.

We need a post-fielding TES.



Systems should be monitored and managed throughout deployment

The AI component may interact with other components in unanticipated ways.

Some emergence behaviors will stem from the interaction of multiple components within a single system that were not present or predictable from the individual components alone.

Testers should work with program managers to deploy monitoring tools that track the performance of a fielded AIEC in real time. These tools will help monitor system behavior, resource usage, and any deviations from expected norms.

The AI component can be overfit to environment and system components.

AI components can be brittle and fail to generalize to new novel inputs. Discussions of brittleness and overfitting often focus on factors explicitly captured in the AI component. However, implicit parameters that are not explicitly labelled in the AI component's training data, such as changes to other system hardware and software, can impact the AIEC's overall performance. Consider:

- A dirty camera lens disrupts the AIEC's ability since it is not trained for variations in dirt accumulation, or
- Upgrading to a lighter hardware component degrades an AIEC's performance that implicitly "assumes" the previous weight.

Shift Right



Emergent Behavior

Behavior stemming from the interaction of parts that the individual components cannot carry out in isolation.



Brittleness

An AI component's inability to maintain adequate performance when faced with novel environmental conditions; the AIEC may respond unpredictably to minor input perturbations. Brittleness is often the result of an AI component that was overfit to noise in its training data.



Important!

Identify & document "Shift Right" activities early and often!



Systems should be monitored and managed throughout deployment

Some performance degradations may be severe enough to require intervention.

Testers will need to work with other program stakeholders to develop processes and mechanisms to identify when AIEC performance deviations require some kind of intervention. This will be particularly challenging when a system-level performance degradation cannot be attributed to a specific component.

When components are developed by different contractors, these governance processes should establish conventions for who is responsible for addressing the issue.

Runtime assurance mechanisms may be useful, but they are not a panacea.

Automated governance mechanisms are commonly suggested as a way to prevent an AIEC from operating outside defined boundaries of acceptable behavior. For example, an autonomous car's navigation system might have constraints to prevent it from exceeding speed limits or crossing double lines.

However, a poorly implemented governance mechanism might lead to undesirable outcomes, particularly in off-nominal situations, such as failing to cross a double line to avoid an accident or driving at the speed limit despite icy road conditions. While automated governance mechanisms have the potential to limit unsafe behaviors, their performance must be rigorously assessed to minimize unanticipated, negative outcomes.

Shift Right



Runtime Assurance

An automated governance mechanism that monitors a capability and intervenes if it operates beyond technical limitations or prescribed unacceptable behavior.



Important!

Identify & document "Shift Right" activities early and often!



Reflecting on SI T&E of AIECs

This Section:

- + Discusses how successful SI T&E is critical for deploying trustworthy AIECs

05



Framework Recommendations



SI T&E builds justified confidence that DoD AIECs will function as holistic units.

Systems integration is the structured approach to combine building block components, such as the relevant software and hardware, into functional and useful systems.

CDAO recommends iteratively integrating AI components into the larger system and testing the system to catch integration errors or deficiencies early and often.



TESs must adapt SI T&E activities to account for novel challenges posed by DoD AIECs.

The TES should include the necessary information required to support systems integration evaluations.

Perform SI T&E in accordance with the TES; include functionality, reliability, security, compatibility, and interoperability test activities.



SI T&E must be incorporated across the AIEC lifecycle, from acquisition to sustainment.

Shift Left—Early-stage testing and assurance activities are essential for identifying and addressing integration risks.

Shift Right—Post-fielding testing of AIECs might be necessary (i.e., shift right) due to changes in integrated components, such as new sensors, model updates, or environmental shifts.

