

Verifying Security in a Safety Context

Airworthiness and DO-326A/DO-356A



Cary Bryczek

Jama Software

Director of A&D Solutions



Ehsan Salehi

LDRA

Senior Field Application Engineer

Sample airborne systems customers



NORTHROP GRUMMAN

BAE SYSTEMS

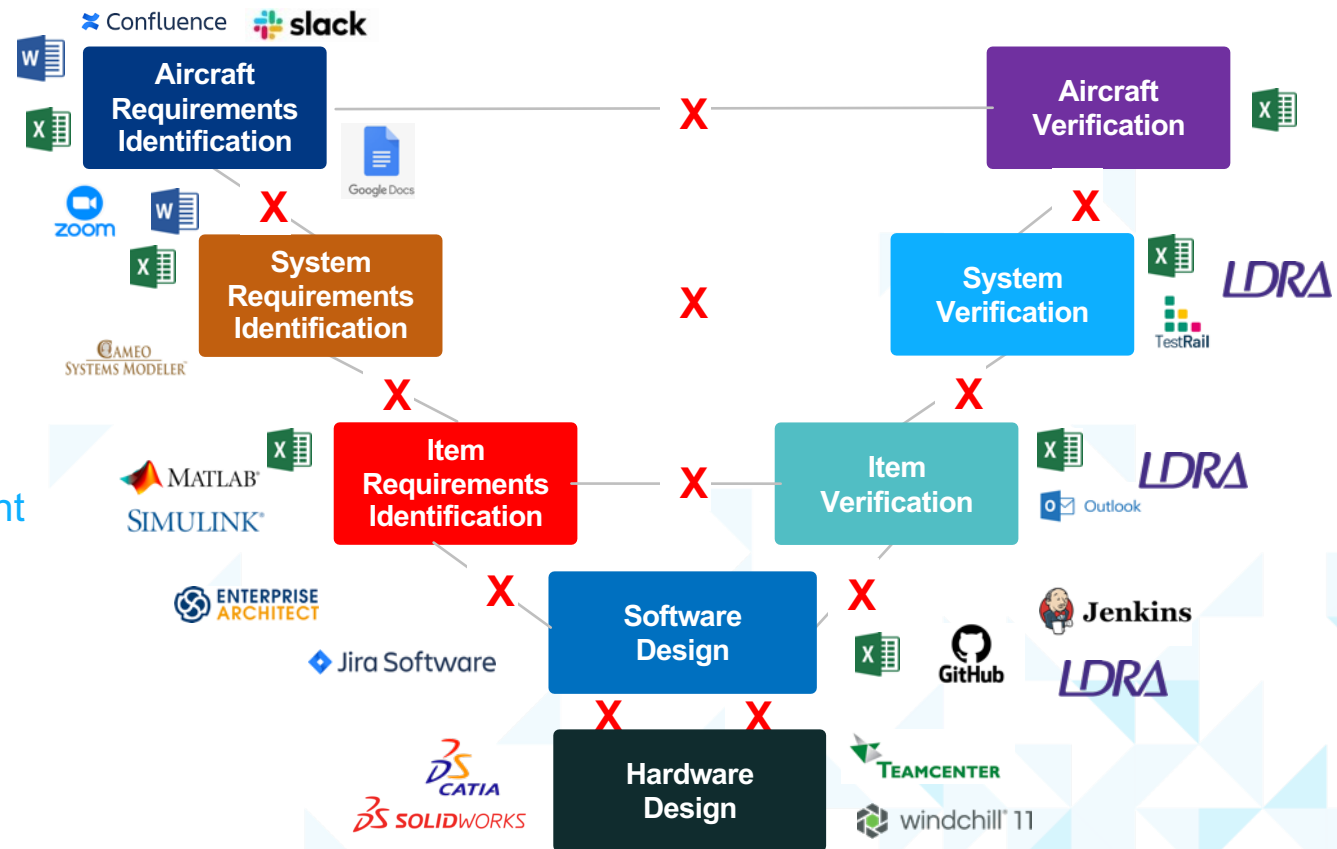


Partnering for Cybersecurity by Design

PREVENTING NEGATIVE PRODUCT OUTCOMES & COMPLIANCE ISSUES

#1 Cause of Product Delays, Defects, Cost Overruns, Audit Failures, Product Failures

- Late identification of defects/coverage gaps due to lack of visibility through the development process
- Poor requirement coordination and change management between hardware/software
- Lack of ongoing risk assessment and change management
- Highly manual, after the fact traceability effort to satisfy contract requirements and standards (ex: ARP4754, DO-178C, DO-254, DO-326/356, ARP4761)
- No way to integrate desktop tools and Excel files



Live Traceability™

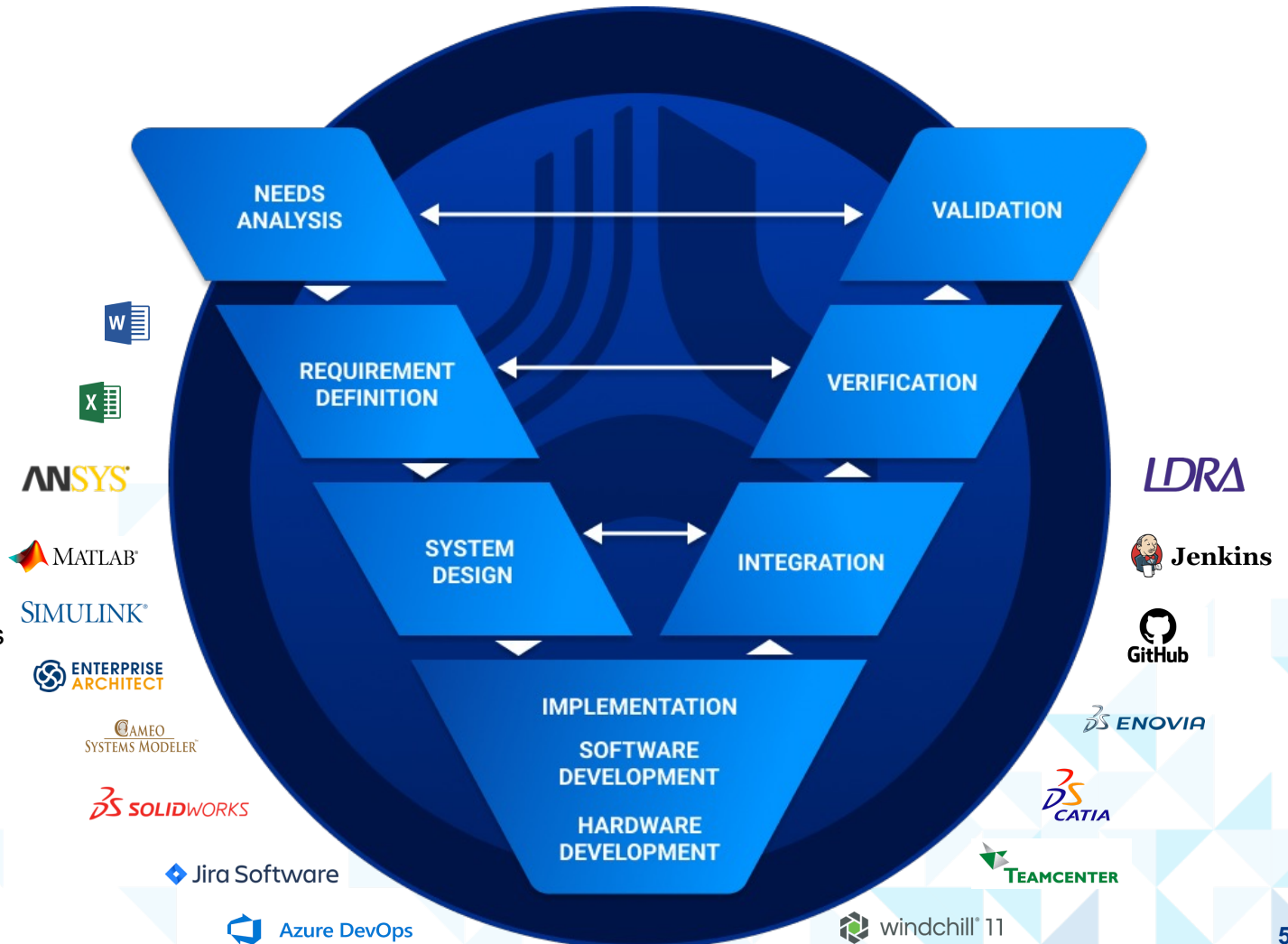
RESOLVES NEGATIVE PRODUCT OUTCOMES & COMPLIANCE ISSUES

WHAT IT IS

Live requirements traceability is the ability for any engineer at any time to see the most up to date and complete upstream and downstream information for any requirement—no matter the stage of systems development or how many siloed tools and teams it spans. This enables the engineering process to be managed through data, and its performance improved in real time.

WHY IT'S IMPORTANT

Live Traceability of system requirements is required by industry standards to ensure product safety and forms the foundation for digital engineering and model-based systems engineering. It delivers significant productivity improvements, and dramatically reduces the risk of product delays, cost overruns, defects, rework, and recalls resulting in faster time to market.



Advantages to Using Jama Connect & LDRA Together

- SINGLE METHOD TO MANAGE SECURITY ASSURANCE (NOT WORK IN OFFLINE SPREADSHEETS AND TOOLS)
- A single system to manage assets, architecture, vulnerabilities, security threats, security requirements and the verification tests.
- Relationships between cyber security items to security requirements and verification test cases
- Built-in Version control capabilities at the atomic item level and overall Security Assurance level
- Leverage Live Traceability to understand coverage, context, and impact
- Manage item lifecycle – Draft -> Review -> Approved
- Built-in and custom reports available for export need

The “DO-326/ED-202 Set”

AVIATION CYBERSECURITY

The DO-326/ED-202 set of standards, jointly developed by RTCA (U.S.) and EUROCAE (Europe) since 2006 includes, at its core, the following standards:

- DO-326A/ED-202A: “Airworthiness Security Process Specification,” with the original edition issued in 2010, and its revision A in 2014. This is the “header” standard, from which all the others evolve, and which defines the cyber-security process for aircraft and systems development, including the actions to be taken for certification per-se.
- DO-356A/ED-203A: “Airworthiness Security Methods & Considerations,” with the original editions issued in 2014 and 2015 (respectively), and its revision A in 2018. This is a detailed, practical, guide for the implementation of DO-326A/ED-202A, and could be as well considered its “part B.” DO-355/ED-204: “Information Security Guidance for Continuing Airworthiness,” issued in 2014. This is an “in-service” guide for airworthiness cybersecurity, versus the previous two standards, aiming at the development stage, and as such is already in relatively wide use by developers as well as operator

What is Airworthiness Security

AVIATION CYBERSECURITY

- RTCA DO-326A. *“Airworthiness security is the protection of the airworthiness of an aircraft from intentional unauthorized electronic interaction. Existing safety processes have not had to consider intentional disruption. Intentional unauthorized electronic interaction (also known as “unauthorized interaction”) is defined as human-initiated actions with the potential to affect the aircraft due to unauthorized access, use, disclosure, denial, disruption, modification, or destruction of electronic information or electronic aircraft system interfaces. This definition includes the effects of malware on infected devices and the logical effects of external systems on aircraft systems, but does not include physical attacks or electromagnetic jamming.”*
- DO-326A: “The purpose of the Airworthiness Security Process (AWSP) is to establish that, when subjected to unauthorized interaction, the aircraft will remain in a condition for safe operation (using the regulatory airworthiness criteria)”

Applying the Airworthiness Security Assurance Process

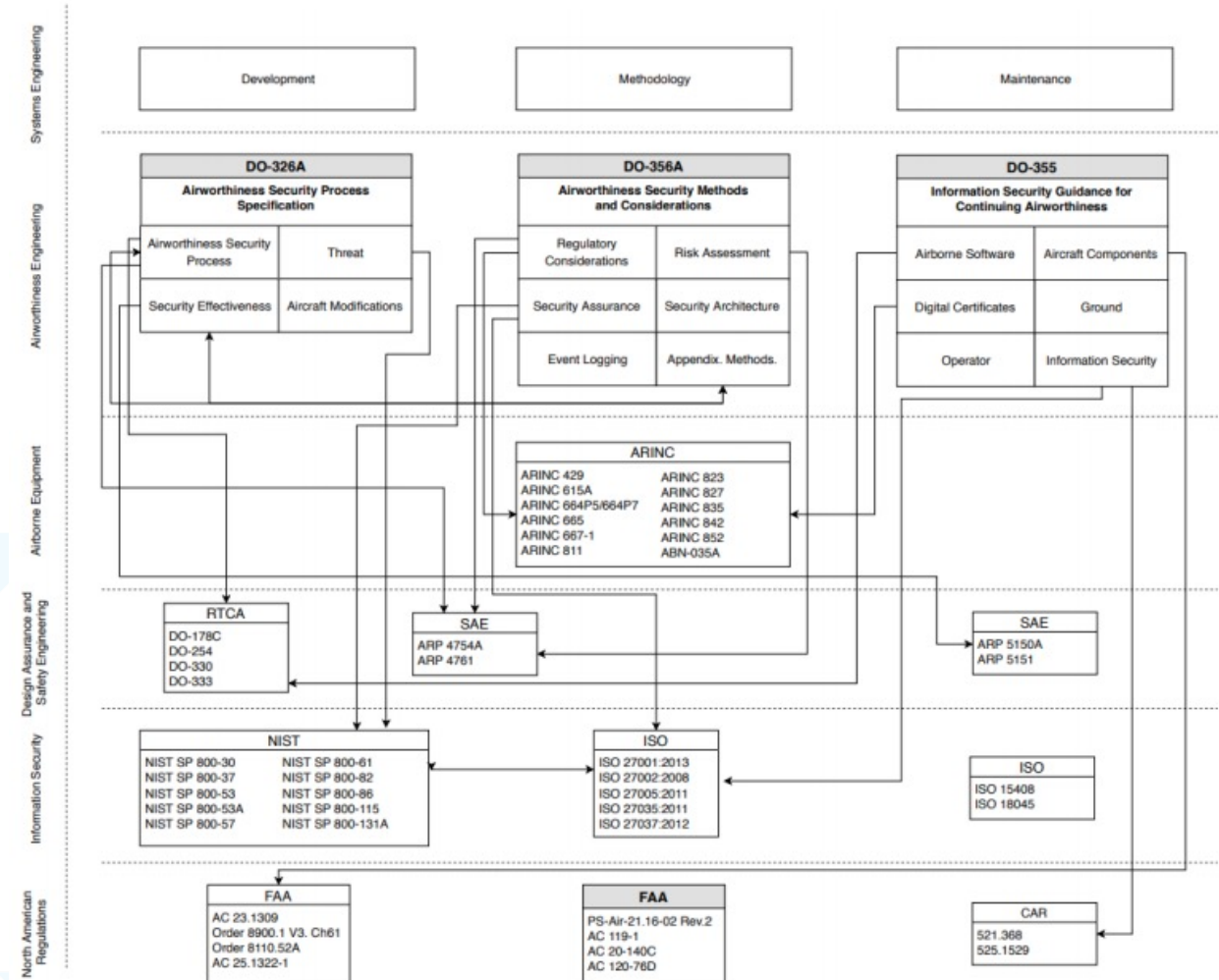
WHAT IT IS AND HOW YOU DO IT

The What

DO-326A	
Airworthiness Security Process Specification	
Airworthiness Security Process	Threat
Security Effectiveness	Aircraft Modifications

The How

DO-356A	
Airworthiness Security Methods and Considerations	
Regulatory Considerations	Risk Assessment
Security Assurance	Security Architecture
Event Logging	Appendix Methods



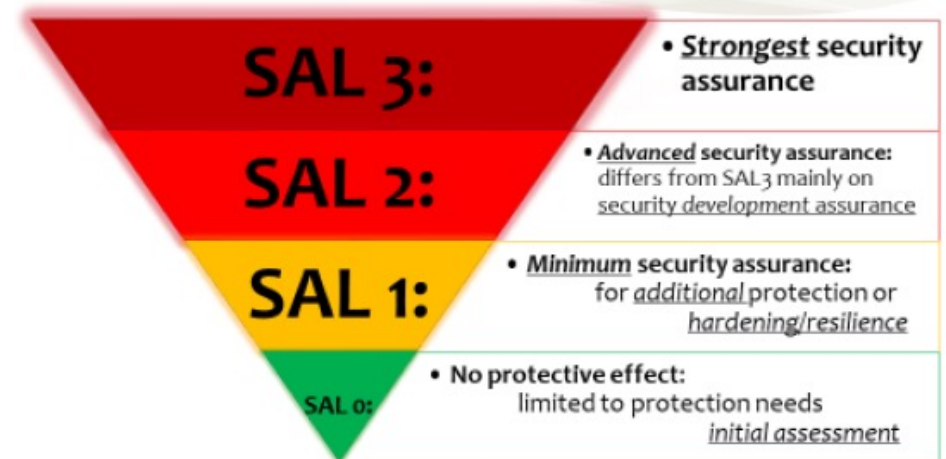
Airworthiness Security Process (AWSP)

DO-326A/ED-202A & DO-356A/ED-203A

Airworthiness security is the protection of the airworthiness of an aircraft from intentional unauthorized electronic interaction

Basic Process

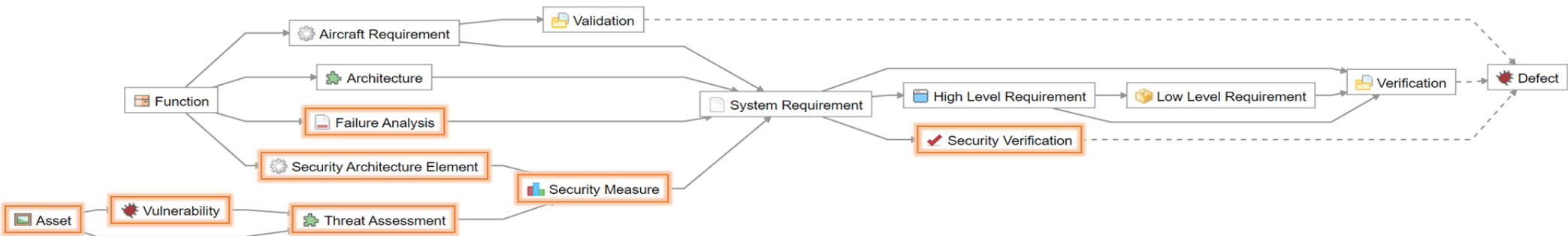
1. Identify System Assets & Perimeters
2. Identify Threats for each Asset
3. Identify Risks for each threat
4. Create Controls/Mitigations for each risk



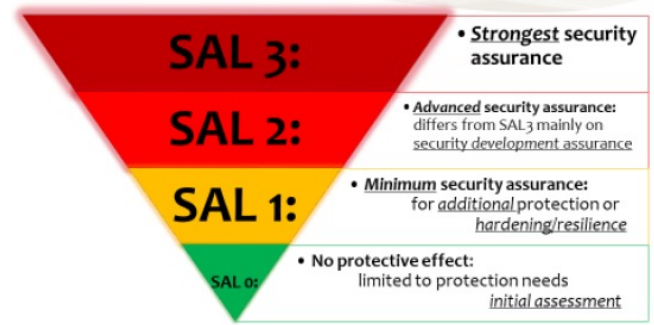
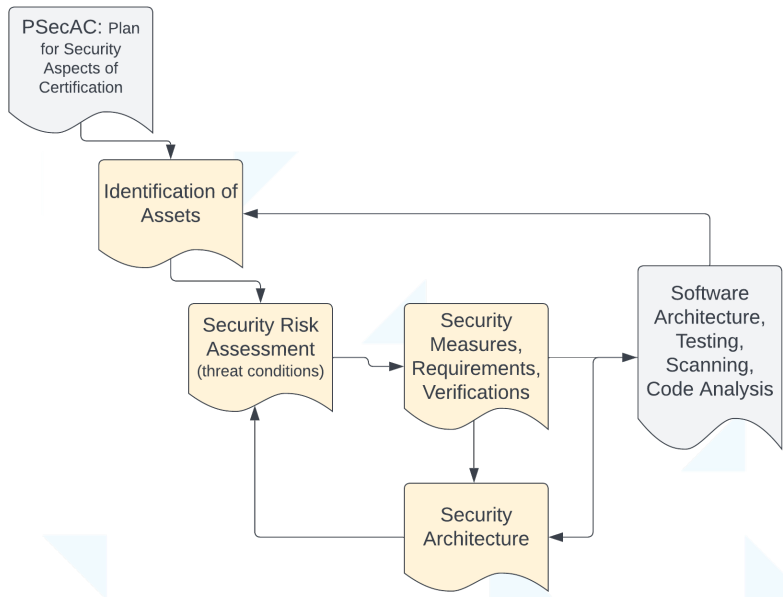
DO-326A Cybersecurity

AIRWORTHINESS SECURITY PROCESS

DO-326A



- Airworthiness Security Analysis
 - Cybersecurity Risk Assessment Process
 - Assets
 - Wired Devices
 - Wireless Devices
 - Data Entry Devices
 - Mechanical Devices
 - What is an Asset
 - Cyber Vulnerabilities
 - Security Architecture (Scope)
 - Threat Assessments
 - Security Measures
 - Security Verification Test Cases

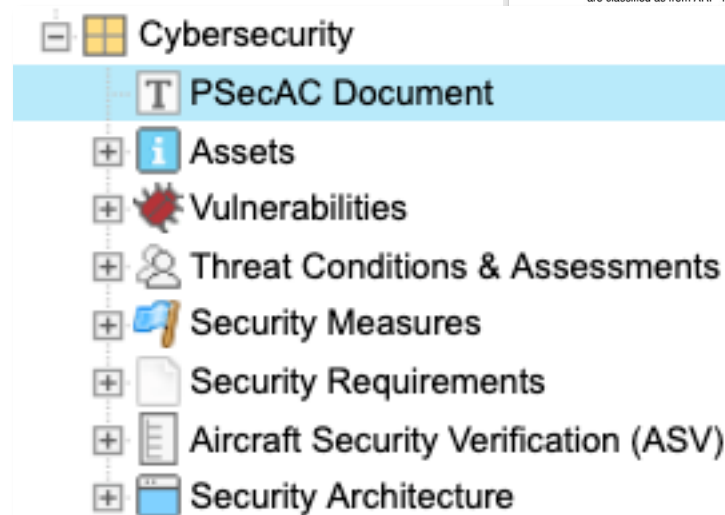


DO-326A Airworthiness Security Process - 7 Steps

2.0 Step 1: Plan for Security Aspects of Certification	Aircraft Level Planning / System Level Planning
2.0 Step 2: Security Scope Definition	Threat Assessment Process
2.0 Step 3: Security Risk Assessment	Threat Assessment Process
2.0 Step 4: Decision Gate	Threat Assessment Process
2.0 Step 5: Security Development	Definition of Security Measures & Requirements
2.0 Step 6: Security Effectiveness Assurance	Verification and Validation of Security Measures and Requirements
2.0 Step 7: Communication of Evidence	PSecAC Summary Reporting

Step 1 - Develop Plan for Security Aspects of Certification document (PSecAC)

- Use Plan item type in Jama Connect to author or import PSecAC document.
- Alternatively, use Plan item type to reference released documents in a document management system using Document Control ID or URL.



Plan for Security Aspects of Certification (PSecAC)

Introduction

1.1 Purpose

This document describes the processes that ACME Engineering will use in developing the software for the Aircraft X. This document also describes how ACME Engineering will demonstrate compliance to RTCA/DO-326A, criticality Level A. This plan is the primary means used by the certification authority for determining whether the proposed software life cycle is commensurate with the rigor required for the level of software being developed.

1.2 Scope

This document will guide the certification of the software for the Aircraft X. The required criticality level of the software is Level A. A Functional Hazard Assessment (FHA) has determined that all failure conditions are classified as from ARP 4754. Based on the results of the FHA, the Software Assurance (Criticality) is Level A. This document describes the system and software overviews, life cycle and software life-cycle data, schedule, additional documents referenced by this Plan.

BA Level A design assurance objectives. This document serves as the Aircraft X software component of the Aircraft X.

Documents

Documents provided by the Customer and referenced by this Plan.

Table 1-1 Documents	
Title	
ME-X, Plan-1 Aircraft Architecture Specification	

Documents referenced by this Plan.

Table 1-2 Documents	
Title	
Worthiness Security Process Specification	
Worthiness Security Methods and Considerations	
As applicable	
As applicable	
As applicable	

Step 2 - Identify Security Scope, Perimeter, and Assets

- Security scope and security perimeter will inform the creation of Asset items in Jama.
- It is often helpful to create diagrams using Jama's diagram editor for helpful reference of the scope and perimeter in relation to the assets.



Step 3 - Identify Threats for Each Asset

- For each Asset, create one or more Threat Assessment items.
- Create trace relationships between the Asset items and Threat Assessment items.
- For each threat record the analyzed threat conditions.
- For each threat assessment, record the threat scenarios.

The screenshot displays the Jama Software interface for a Threat Assessment. The top navigation bar includes tabs for 'Learn more', 'Dashboard: Airborne Ground Surve...', 'Cybersecurity', 'AGS-TA-2: If aircraft s...', and 'Threat Conditions & As...'. The main content area is titled 'AGS-TA-2' and shows a threat assessment for the project 'AGS-TA-2' (GLOBAL ID: GID-50656). The threat condition is 'If aircraft systems can not rely on attitude and heading values, th...' (V13). The threat scenario description states: 'If aircraft systems can not rely on attitude and heading values, the aircraft is declared in Degraded Mode, and will need to exit from duty and return to hangar for maintenance'. A flowchart illustrates the threat scenario: 'Airline maintenance laptop infected with malware' leads to 'Malware will find and manipulate NDB files', which leads to 'Laptop is connected to aircraft and infected NDB files are uploaded', which leads to 'Aircraft in flight faces elevated risk of crashing'. The threat source is 'Airline maintenance computers are infected with Malware', the threat agent is 'Malicious actor', and the attack vector is 'Malware placed on USB device manipulates NDB files'. The effects are listed as 'the Aircraft is available for duty', 'Field loadable software', 'Configuration files', 'Firmware', 'Data storage in Pressurization Controller', 'Digital certificates', 'Download Software PN prior installation', and 'Download Software HW prior installation'. The interface also shows a sidebar with 'Cybersecurity' and 'Vulnerabilities' sections, and a bottom panel with 'Assets' and 'Threat Conditions & Assessments'.

Step 4 - Identify Risks for Each Threat

For each threat assessment record the severity of threat condition and the level of the threat. Jama Connect will automatically generate the Acceptability (Unacceptable or Acceptable).

Threat Conditions & Assessments										Add		Trace view	Export
Set • View details													
3 items													
		ID	Threat Condition (Name)	Threat Scenario (Description)	Threat Agent	Threat Source	Attack Vector	Effects	Security Measures ...	Level of Threat	Severity of Threat C...		
☐		AGS-TA-2	If aircraft systems can not rely on attitude and headin...	If aircraft systems can not rely on attitude an...	Malicious actor	Airline mainte...	Malware place...	Files on the N...	Preventative	Moderate	Major		
☐		AGS-TA-3	If wrong attitude and heading values (erroneous signa...	—	Malicious Actor	Airline mainte...	Malware place...	Files on the N...	Preventative	Very High	Hazardous		

Step 5 - Create Controls/Mitigations for Each Risk

- For each Threat Assessment create a trace relationships to a Security Measure item. Create a new security measure if necessary.
- Perform Security Measure characterization documenting a control/mitigation by relating it to a System Requirement. Create a new requirement(s) at the System Requirement level as necessary.

The screenshot displays the LDRA Technology software interface. On the left, a project explorer shows a hierarchy for 'Airborne Ground Surveillance System', including 'Threat Conditions & Assessments'. The main window shows a table titled 'Threat Conditions & Assessments' with columns for Project ID, Threat Condition, Threat Scenario, and Level of Threat. Below this, a table titled 'Security Measure (1)' shows columns for Project ID, Name, and SAL. The table contains three rows of data.

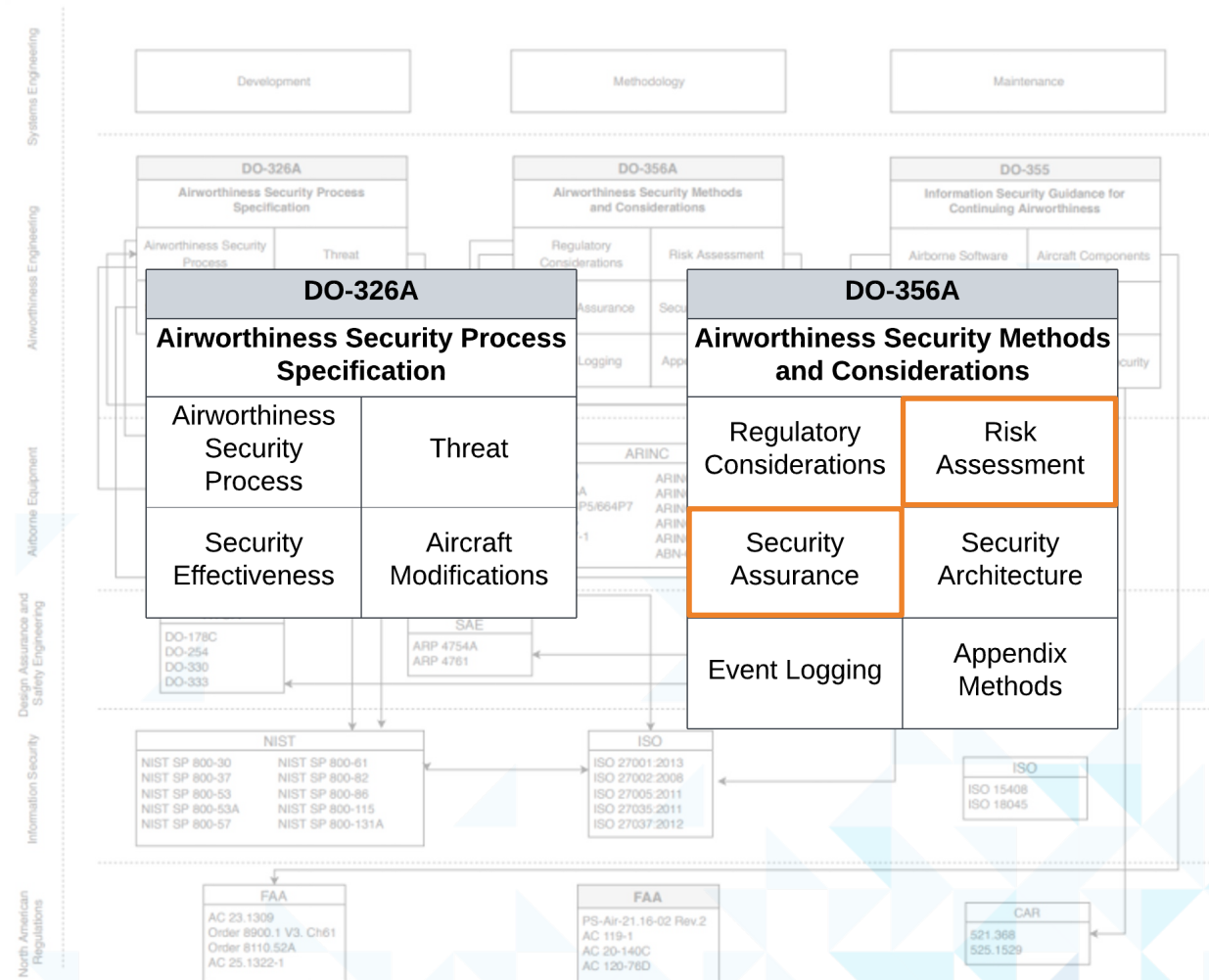
Threat Assessment (3)				Security Measure (1)			
	Project ID	Threat Condi...	Threat Scena...	Level of Threat	Project ID	Name	SAL
☐	AGS-TA-2	If aircraft syst...	If aircraft syst...	Moderate	☐	AGS-MEAS-2	Secure the Inertial V... SAL 2
☐	AGS-TA-3	If wrong attitu...		Very High	☐	AGS-MEAS-2	Secure the Inertial V... SAL 2
☐	AGS-TA-1	Loss of acclim...		Low			

Step 6 - Communication of Evidence

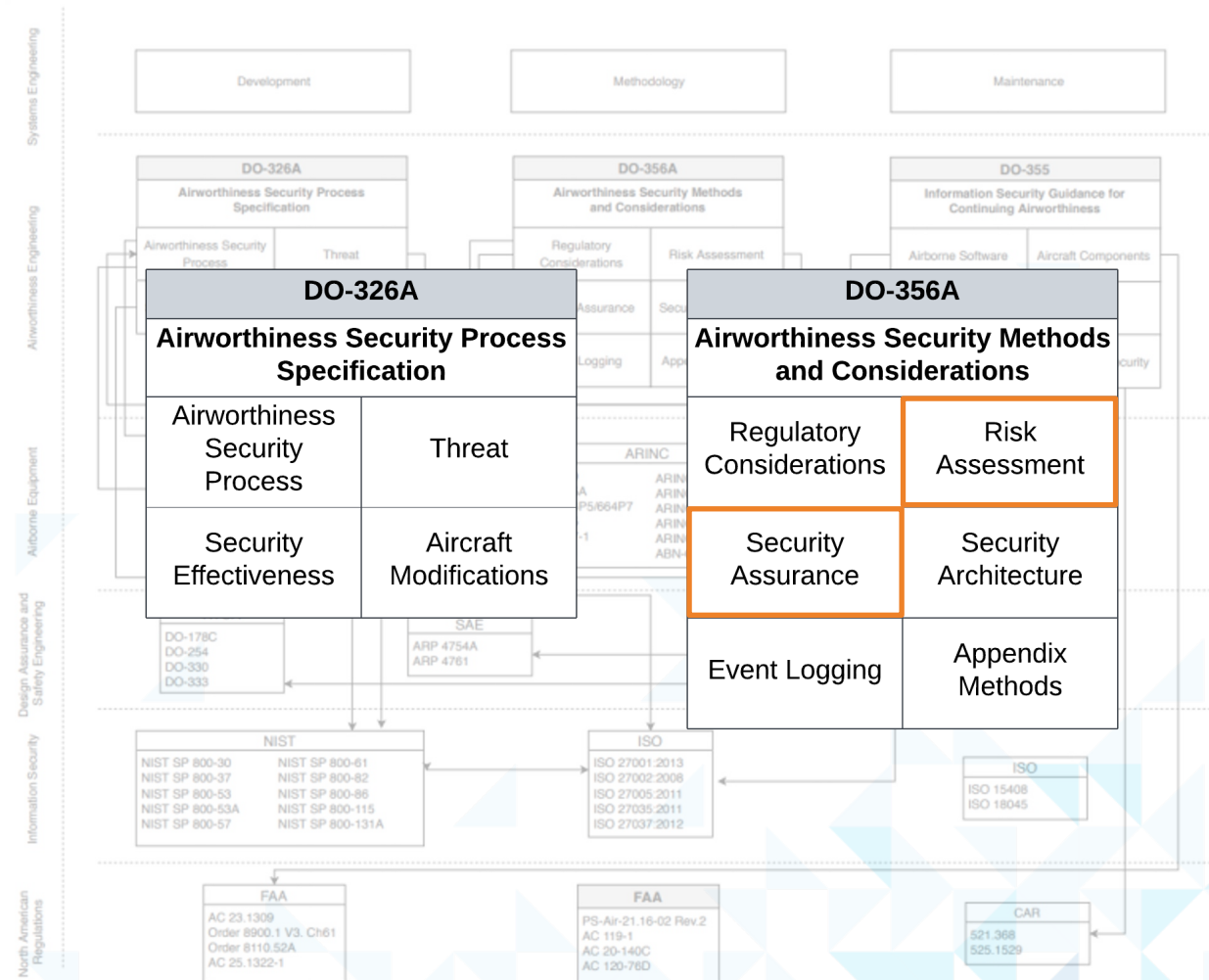
- Use the filters in Jama to identify Threat Assessments whose Acceptability is Unacceptable. For these Assessments, create a trace report (live trace in the UI or traceability export) to analyze and report on risk assessment activities.
- Update the PSecAC document as necessary with compliance demonstration evidence.

Risk Assessment Objectives

Ref.	Objective	Jama Supporting Capability
O1.1	The security scope is established and validated.	Security Assessment and Review Center
O1.2	The Threat Condition Identification and Evaluation is complete and validated.	Security Assessment and Review Center
O1.3	The Preliminary Aircraft/System Security Risk Assessments and Aircraft/System Security Risk Assessments are performed and consistent with related aircraft/system safety assessments.	Risk Assessment Process
O1.4	Preliminary Aircraft/System Security Risk Assessment results have been processed to define aircraft/system security architecture and identify the need for security measures.	PSecAC Summary Reporting
O1.5	Aircraft/System Security Risk Assessment is consistent and complete with respect to security scope, security guidance, security requirements, security verification, security refutation and vulnerability identification.	Security Assessments, Review Center, & Coverage Analysis

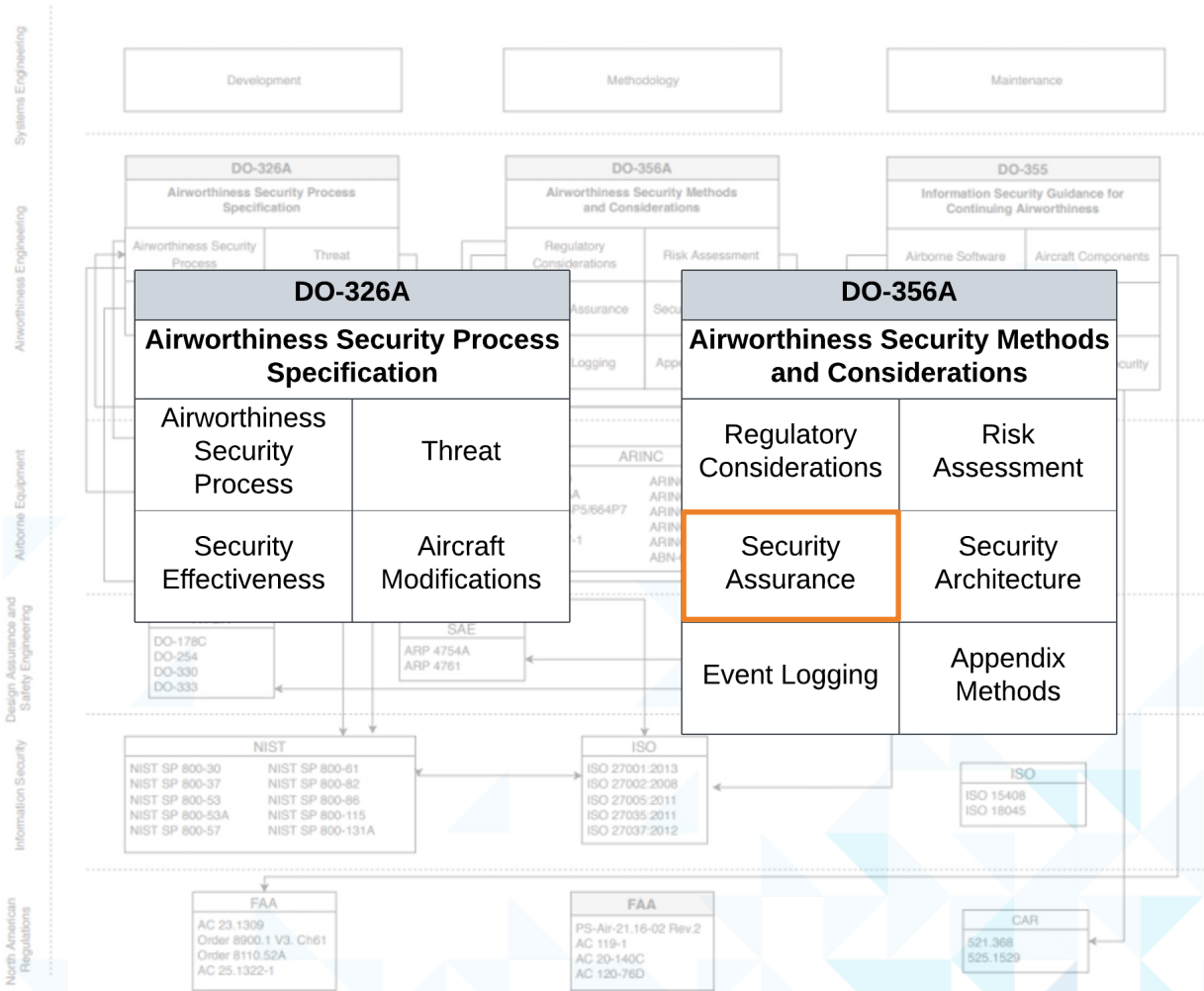


Ref.	Objective	Jama Supporting Capability
O2.1	Vulnerabilities in security measures and assets (including COTS) are identified and evaluated for their potential impact on safety.	Vulnerability and Threat Assessment Process
O2.2	Vulnerabilities are treated according to their evaluation.	Vulnerability and Threat Assessment Process



Security Refutation Objectives

Ref.	Objective	LDRA Supporting Capability
O3.1	Refutation analyses are performed to identify new vulnerabilities.	
O3.2	Refutation tests are performed to evaluate the exposure of vulnerabilities in the security environment and to challenge the vulnerability evaluation.	TBrun with TBextreme tests
O3.3	Refutation test plans are available. Refutation test results cover refutation test plans and performed tests. Refutation test results are analyzed and discrepancies are justified and traced.	TBmanager and TBreports



Refutation

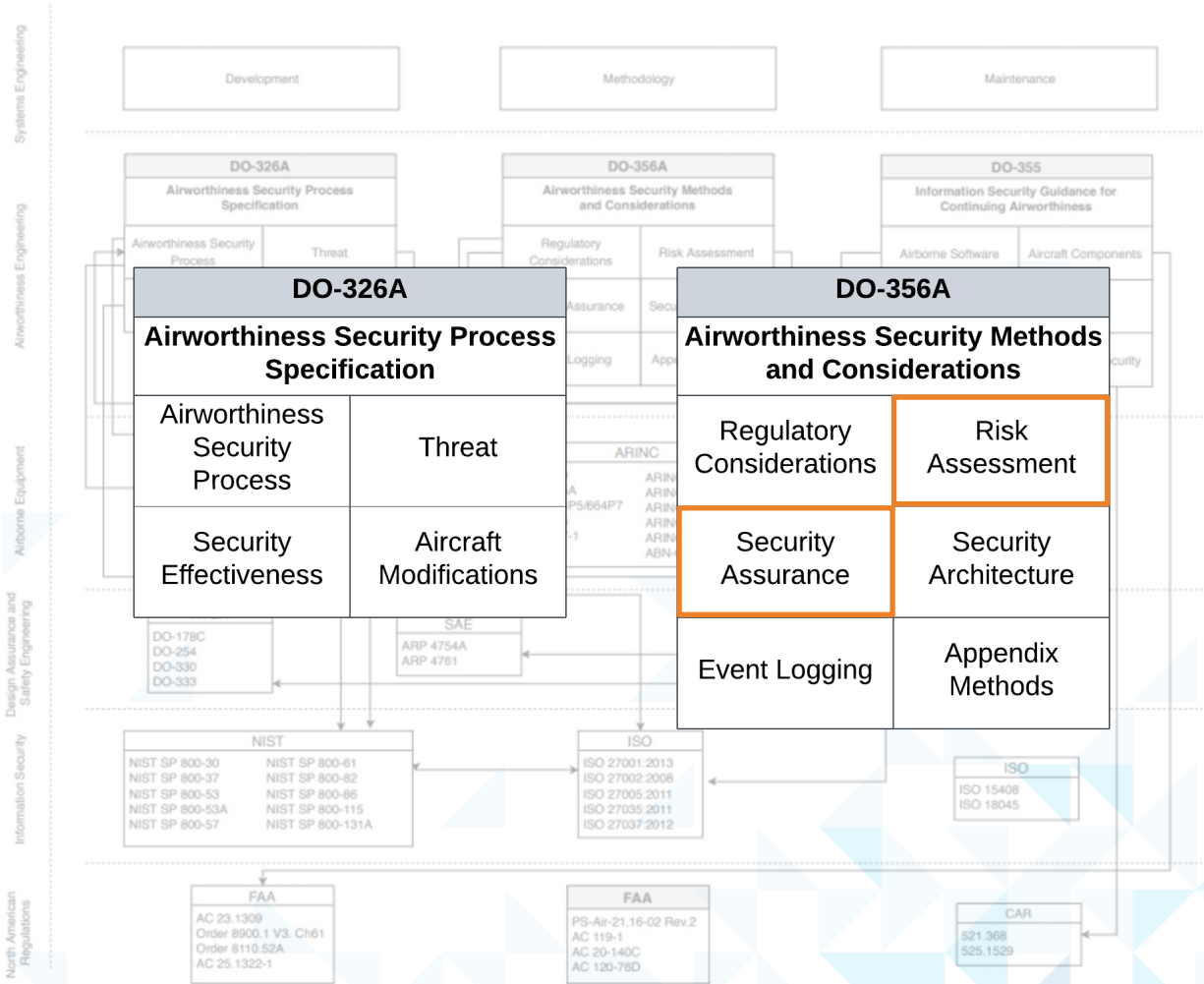
ED-203A / DO-356A

“Refutation acts as an independent set of assurance activities beyond analysis and requirements. As an alternative to exhaustive testing, refutation can be used to provide evidence that an unwanted behavior has been precluded to an acceptable level of confidence.

NOTE: Refutation is also known as Security Evaluation in some contexts.”

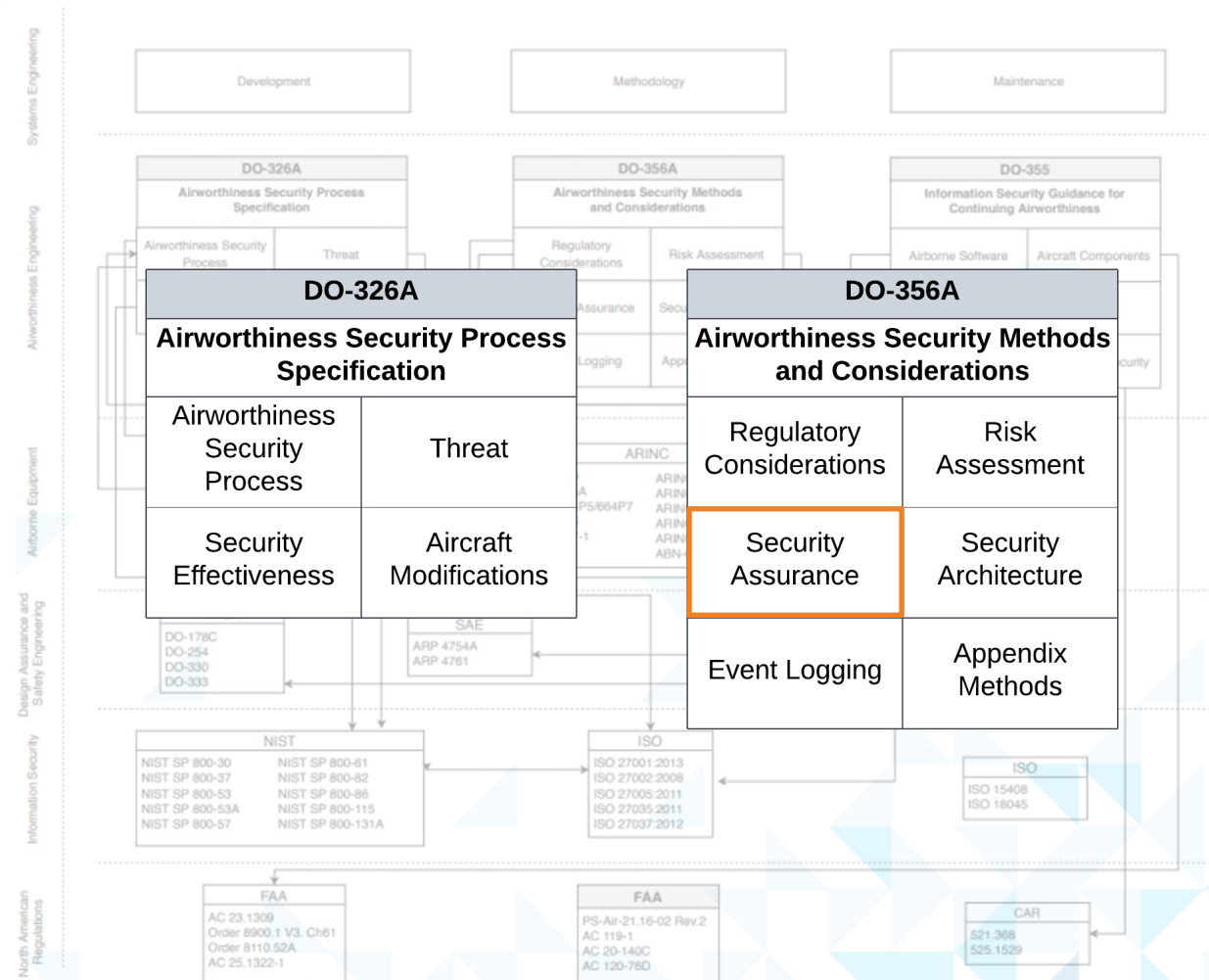
Security Deployment Objectives

Ref.	Objective	Jama Supporting Capability
O4.1	Security guidance is correct, complete and validated against technical and operational security measures and requirements.	Traceability Analysis & Reports

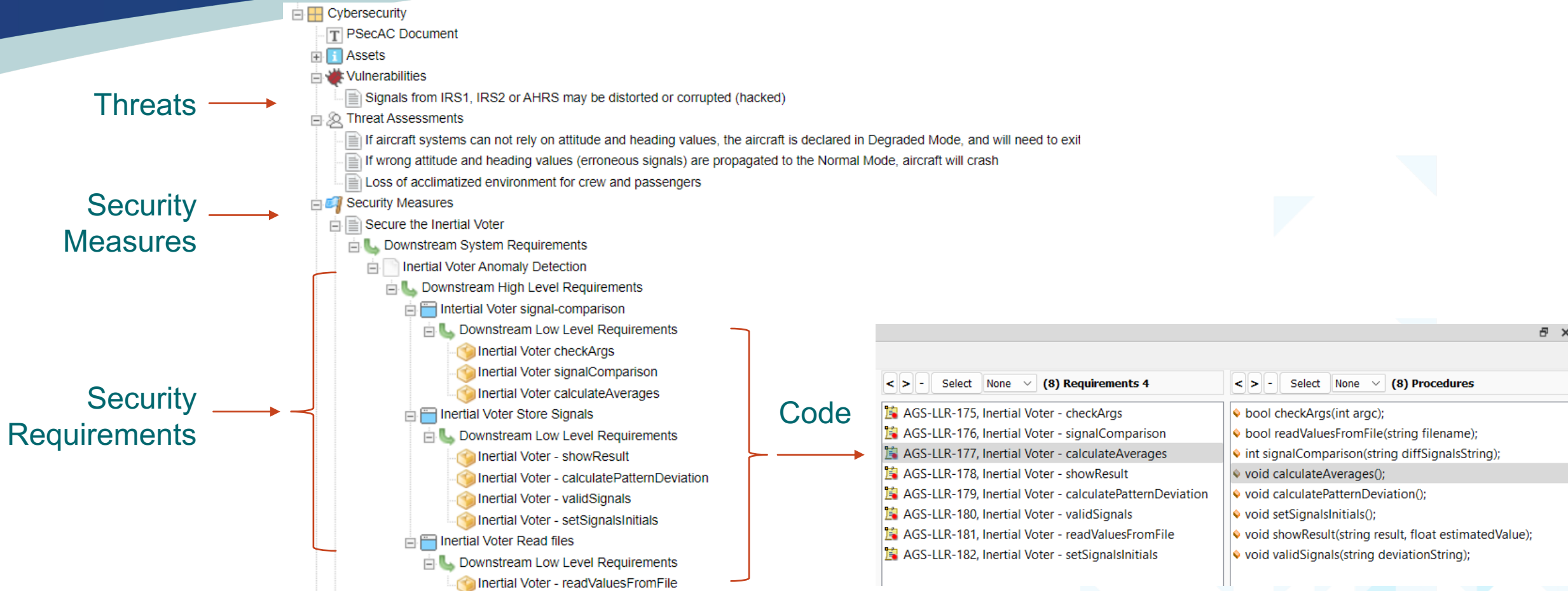


Security Implementation Objectives

Ref.	Objective	LDRA Supporting Capability
O8.1	Source code and/or hardware description complies with security requirements.	TBmanager
O8.2	Source code and/or hardware description conforms to security coding standards.	TBvision, TBsecure, and TBreports
O8.3	Source code and/or hardware description complies with item security architecture.	TBmanager



Source Code Complies with Security Requirements





Source Code Complies with Security Requirements

SECURITY MEASURE -> SECURITY REQUIREMENTS -> HIGH LEVEL REQUIREMENTS

The screenshot displays the LDRA TBmanager 10.1.0 interface. The title bar indicates the file is 'DO326_356_demo.tbp'. The menu bar includes Project, Source, Configure, View, Website Links, Reports, Impact Analysis, Project Baseline Analysis, Tools, Import, and Help. The toolbar contains various icons for file operations and analysis.

The **Project Tree** panel on the left shows a hierarchical structure:

- Objectives
- Requirements (0/12 Verified)
 - AGS-MEAS-2, The Inertial Voter (highlighted with a red box)
 - AGS-SS-2, Inertial Voter
 - AGS-HLR-138, Inertial Voter Signal Comparison
 - AGS-LLR-175, Inertial Voter - checkArgs
 - AGS-LLR-176, Inertial Voter - signalCompa...
 - AGS-LLR-177, Inertial Voter - calculateAve...
 - AGS-HLR-139, Inertial Voter Store Signals
 - AGS-LLR-178, Inertial Voter - showResult
 - AGS-LLR-179, Inertial Voter - calculatePatt...
 - AGS-LLR-180, Inertial Voter - validSignals
 - AGS-LLR-182, Inertial Voter - setSignalsInit...
 - AGS-HLR-140, Inertial Voter Read files
 - AGS-LLR-181, Inertial Voter - readValuesFr...
- SecurityMeasures
- SecurityRequirements
- HighLevelRequirements
- LowLevelRequirements
- TestCases
- Source Code
 - files.cpp, C:\LDRA_Workarea_C_CPP_10.1.0\Examples\T...
 - signal-comparison.cpp, C:\LDRA_Workarea_C_CPP_10.1...
 - values_utils.cpp, C:\LDRA_Workarea_C_CPP_10.1.0\Exa...

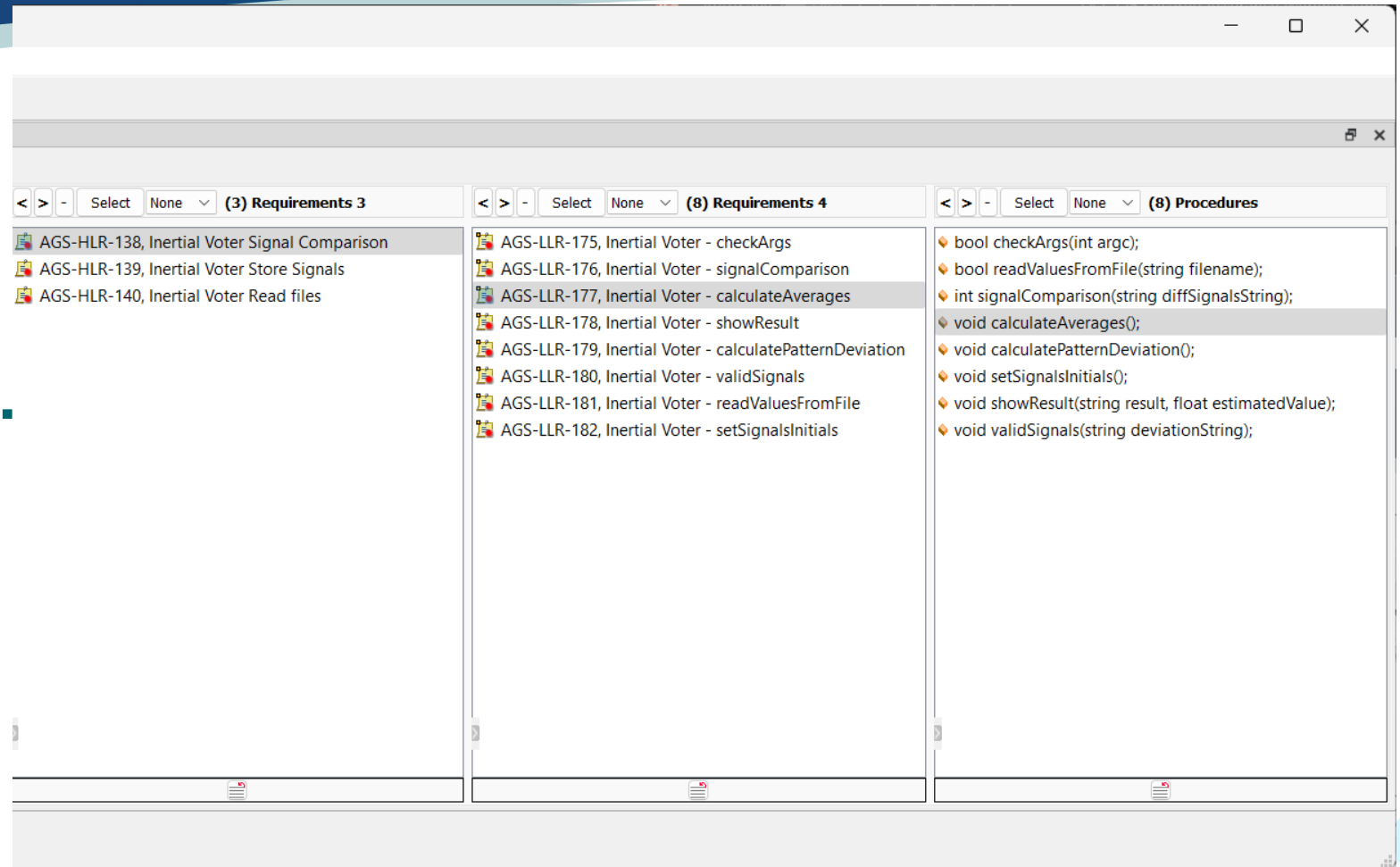
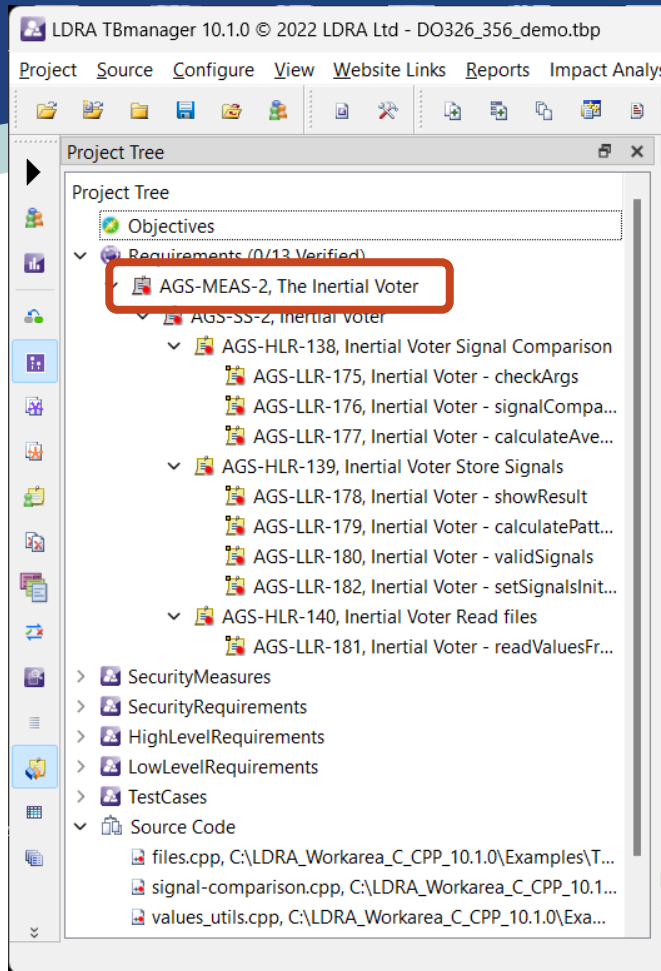
The **Relationships** panel on the right shows three tabs: (0) Four-Level Requirements to Procedures, (1) Source to Test (Mapped to Verification Task), and (2) Requirements 1. The (1) Source to Test tab is active, displaying three columns of requirements:

- (1) Requirements 1:** AGS-MEAS-2, The Inertial Voter
- (1) Requirements 2:** AGS-SS-2, Inertial Voter
- (3) Requirements 3:** AGS-HLR-138, Inertial Voter Signal Comparison; AGS-HLR-139, Inertial Voter Store Signals; AGS-HLR-140, Inertial Voter Read files

The bottom of the interface has buttons for 'Map Source' and 'Relationships'.

Source Code Complies with Security Requirements

HIGH LEVEL REQUIREMENTS -> LOW LEVEL REQUIREMENTS -> CODE



Source Code Conforms to Security Coding Standards



Security Report

MISRA-C++:2008

CERT-C++:2016

CWE



LDRA tool suite Security Report

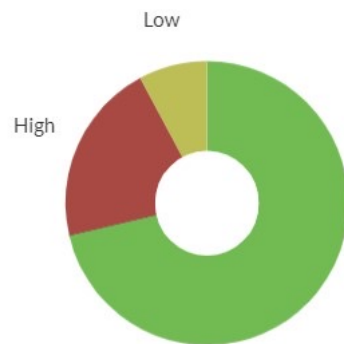
File: C:\LDRA_Workarea_C_CPP_10.1.0\Examples\Toolsuite\DO326_356_demo\Source\signal-comparison.cpp

Date of Analysis
Wed May 3 2023 13:27:26

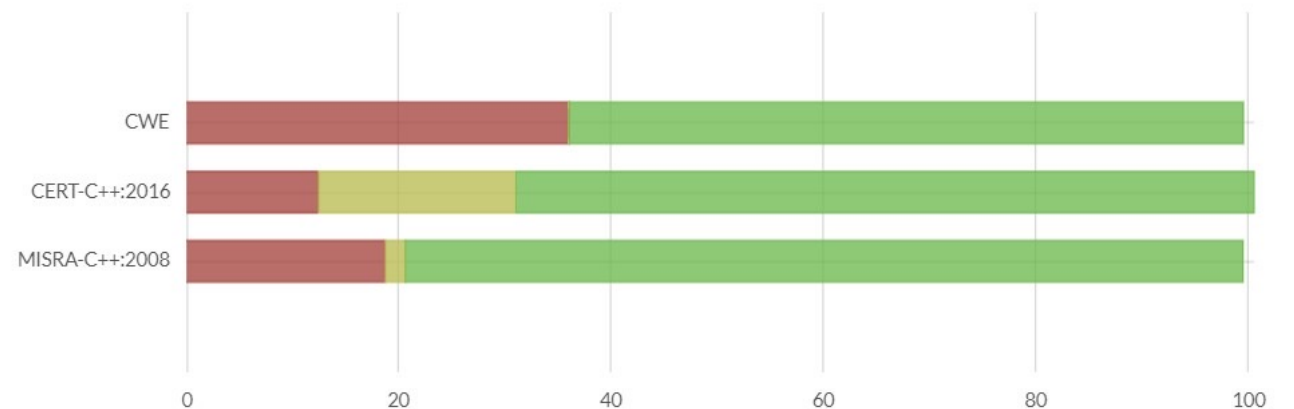
Report Produced on
Wed May 3 2023 13:28:42

LDRA Version
10.1.0

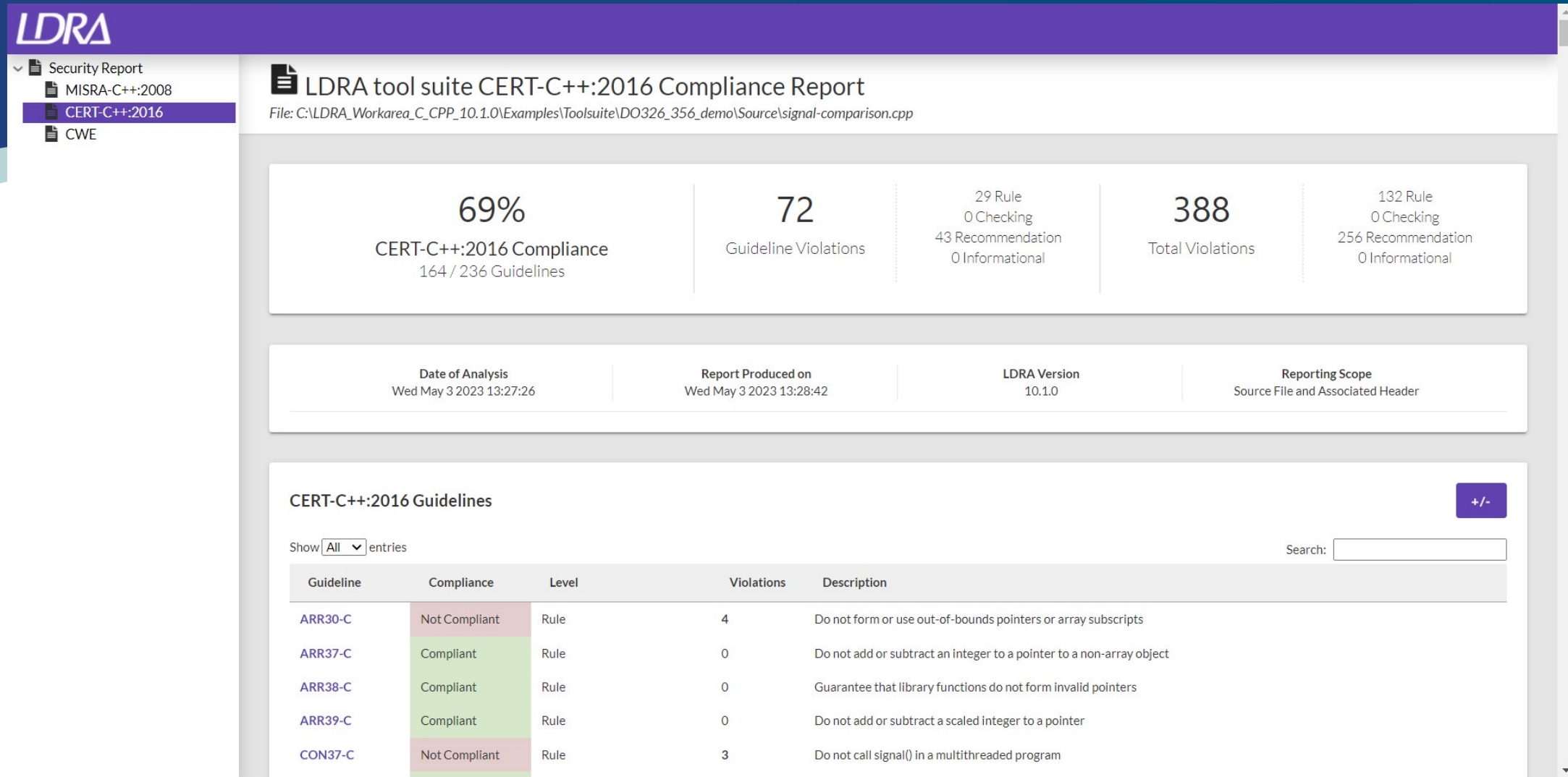
Security Summary



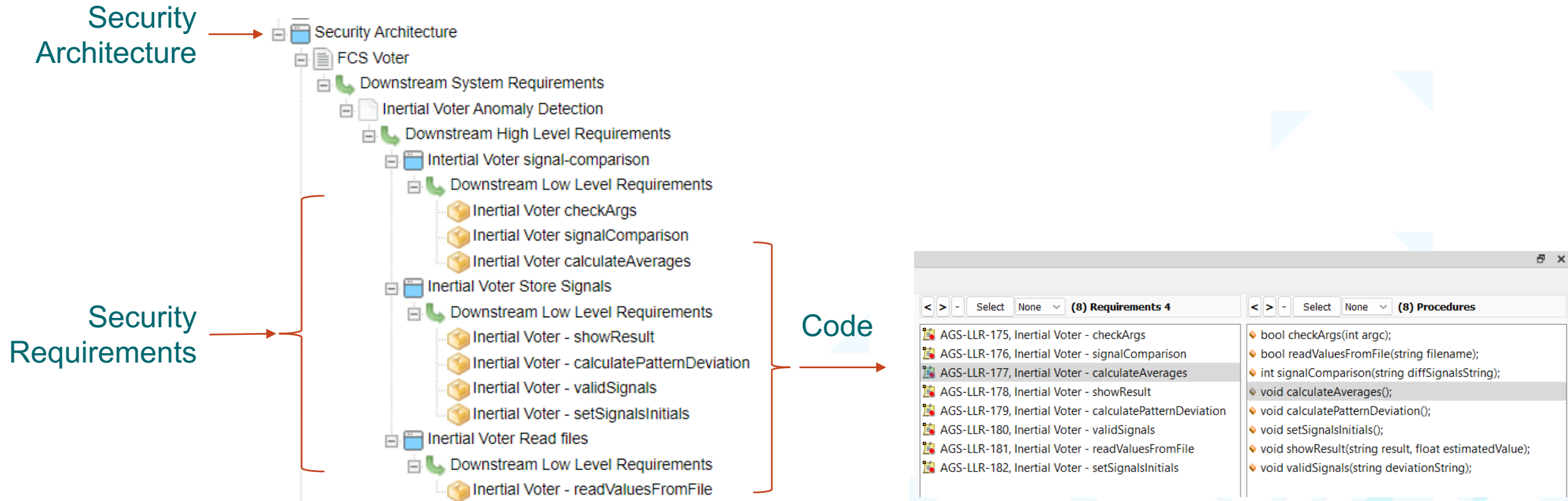
Security Distribution



Source Code Conforms to Security Coding Standards

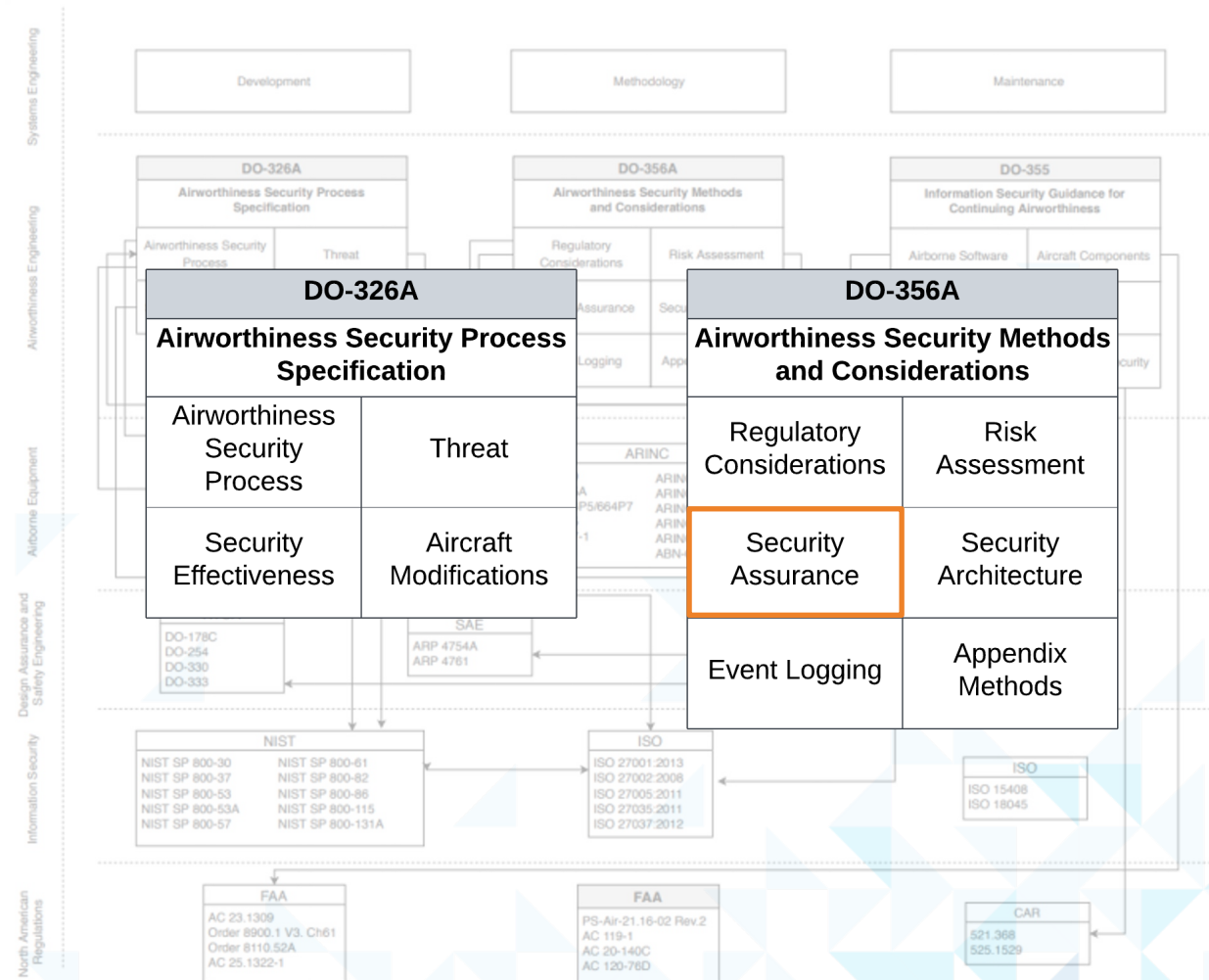


Source Code Complies with Security Architecture



Security Verification Objectives

Ref.	Objective	LDRA Supporting Capability
O9.1	Security functional and robustness verification elements are developed, validated and performed.	TBrun
O9.2	Security functional and robustness verification results are complete and correct, and discrepancies are justified and traced.	TBreports, TBsecure, TBjustify
O9.3	Test coverage of software and/or hardware implementation is achieved.	TBmanager



Security functional and robustness verification elements are developed, validated and performed

Source Code

Test Case Logs

Pass/Fail Results

Source Code Calls

The screenshot displays the TBrun software interface, which is used for security functional and robustness verification. The interface is divided into several panes:

- Log View:** Shows the execution log for the test case "FuzzTest_NullPointer (C++)". It includes input values for variables like `ldra_qq_lv_24`, `ahrs`, `irs1`, `irs2`, and `valuesUtils`. The log indicates that the test case failed due to an exception raised.
- File View:** Displays the source code files for the test case, including `signal-comparison.cpp`.
- Calls View:** Shows the calls made by the test case, including `new`, `pow`, `readValues...`, `setSignals...`, `showResult`, `signalCom...`, `sqrt`, `std::ifstream...`, `std::string::c...`, `std::string::s...`, `strcpy`, `strlen`, and `validSignals`.
- Test Case View:** A table showing the results of 26 test cases. The table has columns for Test Case, Regression P / F, Procedure, Object, Name / Description, Action Code, File Name, Folder, and Creation Date.

Test Case	Regression P / F	Procedure	Object	Name / Description	Action Code	File Name	Folder	Creation Date
17	PASS	Signal:se...	1			signal-co...	C:\LD...	May 3 2023 13:35:26
18	PASS	Signal:sg...	1			signal-co...	C:\LD...	May 3 2023 13:35:26
19	FAIL	setSignal...				signal-co...	C:\LD...	May 3 2023 13:35:26
20	PASS	readValu...				signal-co...	C:\LD...	May 3 2023 13:35:26
21	PASS	validSign...				signal-co...	C:\LD...	May 3 2023 13:35:26
22	PASS	calculate...				signal-co...	C:\LD...	May 3 2023 13:35:26
23	PASS	calculate...				signal-co...	C:\LD...	May 3 2023 13:35:26
24	FAIL	checkArg...				signal-co...	C:\LD...	May 3 2023 13:35:26
25	PASS	showResult				signal-co...	C:\LD...	May 3 2023 13:35:26
26	FAIL	signalCo...				signal-co...	C:\LD...	May 3 2023 13:35:26

Security functional and robustness verification results are complete and correct

Report Summary →

Report Details →

LDRA

Test Manager

Unit / Module Tests

FuzzTest_NullPointer

LDRA tool suite Regression Report

File: C:\LDRA_Workarea_C_CPP_10.1.0\Examples\Toolsuite\DO326_356_demo\Source\signal-comparison.cpp

Sequence FuzzTest_NullPointer

Overall Result

FAIL

Overall Summary

Out of 26 test cases, 23 (88.46 %) pass.

Out of 26 test cases, 3 (11.54 %) fail.

Out of 15 procedural outputs, 5 (33.33 %) fail.

Date of Analysis

Fri May 5 2023 15:53:59

Report Produced on

Fri May 5 2023 16:01:14

LDRA Version

10.1.0

Test Case Regression Summary Table

Test Case	Procedure	File	Inputs	Outputs	Status	Fail%	Suspended%
1	Signal::Signal	signal-comparison.cpp	0	0	PASS	0.00	0.00
2	Signal::Signal	signal-comparison.cpp	1	0	PASS	0.00	0.00
3	Signal::Signal	signal-comparison.cpp	2	0	PASS	0.00	0.00
4	Signal::calculatePatternDeviation	signal-comparison.cpp	0	0	PASS	0.00	0.00

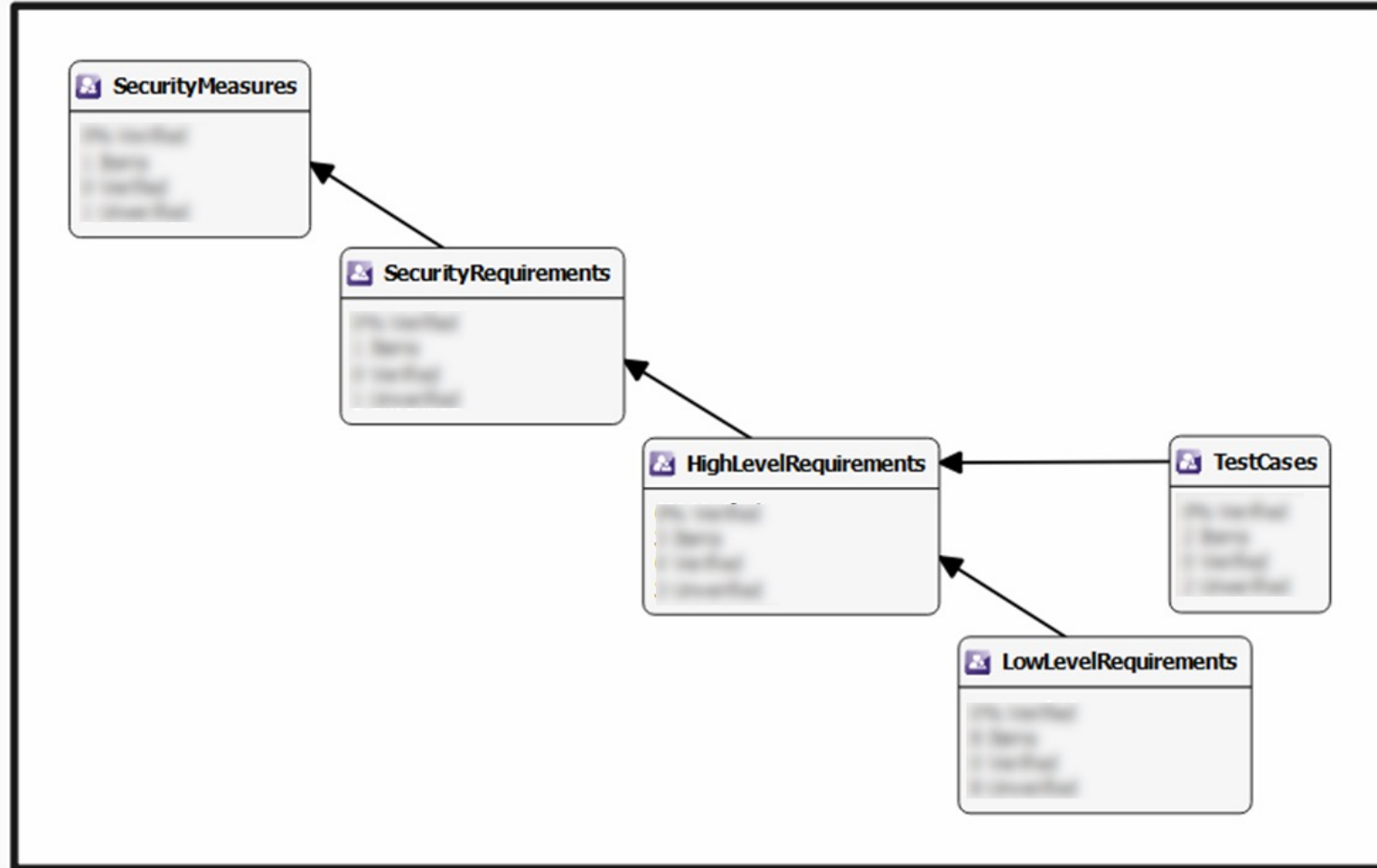
Test Coverage of Software is Achieved

Traceability

Traceability Hierarchy

- ✓  TCI_SAST_1: Perform static analysis and coding standards compliance with MISRA, CERT, CWE. Capture report showing no violations of
 - ◆ void calculateAverages();
- ✓  TCI_DAST_1: Perform Fuzz Testing attempting find null pointer exceptions. -- TBextreme Test
 - >  signal-comparison.cpp, C:\LDRA_Workarea_C_CPP_10.1.0\Examples\Toolsuite\DO326_356_demo\Source\signal-comparison.cpp
- ✓  TCI_1_7: Default
 - ◆ void calculatePatternDeviation();

Test Coverage of Software is Achieved



D0326_356_demo.tbp

Summary

Important capabilities for Verifying Security in a Safety Context include:

- Applying the Airworthiness Security Process
- Tracing Security Measures to Requirements and Tests
- Using SAST and DAST to Identify and Mitigate Security Vulnerabilities
- Improving Assurance through Refutation Testing

Using Jama Connect with the LDRA tool suite provides a comprehensive solution to manage assets, architecture, vulnerabilities, security threats, security requirements and the verification tests, reducing the time, cost, and risks of compliance with DO-326A/DO-356A and achieving airworthiness

Q&A



Thank You



jama
software®

