LDRA
Software Technology

Executive Overview

Airworthiness Handbook

www.ldra.com

Contents

The business implications of airworthiness: Overcoming the challenges of certifying next-generation avionics software

Commercial and military avionics manufacturers are facing dramatic market changes. Challenges include expanding cybersecurity threats, advances in sustainability and autonomy, and the rise of new markets such as advanced air mobility. This increasing demand is an extension of a trend which has seen both the amount and complexity of avionics software nearly double every four years for the past five decades [1]. As avionics companies look to new markets and opportunities, the bulk of enabling capabilities will come from more—and more sophisticated—software.

As avionics companies look to new markets and opportunities, the bulk of enabling capabilities will come from more—and more sophisticated—software.

Development teams working to ensure adequate safety and security, otherwise known as airworthiness, to protect critical systems from failure or attack, are finding that the cost, time, and effort to meet these airworthiness certification goals, is rising proportionally. These demanding and changing circumstances also impact software development and verification decisions:

- **Increasing scrutiny of autonomous systems.**

  As autonomous aircraft become more mainstream, regulations for their development and certification will become more rigorous. Supporting technologies such as machine learning may see restrictions that can impact development and certification cost and time to market. Companies that use proven software development and testing processes will be better prepared to stay ahead of potential certification roadblocks and keep their early-mover advantage.

- **Dramatic surge in numbers of small aircraft.**

  Small aircraft such as air taxis and delivery drones are poised for dramatic growth in an industry that is ill-equipped to deal with it. Existing safety regulations for small aircraft reflected lower-volume, lower-risk scenarios as compared with those for large commercial aircraft. That situation will likely change with the increasing deployment of both piloted and autonomous smaller aircraft.
  Early movers will be dependent on a flexible, scalable design approach that can help them pivot as requirements evolve. An effective approach will lean into many of the stringent design practices that have been proven in commercial aircraft, and that will form the basis of many new requirements.

- **Design impacts for sustainability and zero emissions.**

  Sustainability initiatives include reduced emissions and electrification of aircraft, demanding new software systems for the generation and management of power. System decisions will look to reduce aircraft size, weight, and power requirements, potentially increasing complexity.

  For instance, software that takes full advantage of multicore processors can yield weight savings while increasing functionality and system agility, but software development, testing, and compliance of multicore systems adds layers of complexity and time. Complex software for new technologies such as electric vertical take-off and landing (eVTOL) can also raise certification difficulties. As regulations and technologies evolve, existing standards underpinned by automated tools can provide a sound foundation to mitigate the risk associated with new technologies.

- **New attack vectors from increased connectivity.**

  Avionics systems continue to be more connected for monitoring, maintenance, and diagnostics, as well as for the additional connectivity inherently required for autonomous aircraft. While this connectivity leads to more efficient operations, it also leads to increased security risks that impact safety. This makes a “shift left approach” to security vitally important - that is, ensuring that security is designed into the system alongside aviation safety.

  Connectivity also demands the ability to respond to vulnerabilities. Each newly discovered vulnerability implies a changed or new requirement, and one to which an immediate response is needed – even though the system itself may not have been touched by development engineers for quite some time. In such circumstances, the ability of automated requirements traceability tools to isolate what is needed and automatically regress on only the functions affected becomes something much more significant.

These trends and new capabilities may not only change what the system does, but how it does it—and those changes rely on software. To ensure that avionics software can be certified for airworthiness requires a focus on development processes as well as safety and security assessments. While reuse of existing software allows development and compliance resources to be focused on new functionality, complex systems are harder to develop and harder to get right. This has significantly increased the difficulty of the already-formidable task of certifying next-generation software for airworthiness. At the same time, a growing shortage of software developers means avionics and aerospace leaders are forced to find new efficiencies to meet ongoing product release schedules and to maintain and update existing systems. Initiatives aimed at reducing the cost, time, and risk of certification have become essential activities for business leaders.

Initiatives aimed at reducing the cost, time, and risk of certification have become essential activities for business leaders.

Typically, decisions around software development and verification tools have been managed within individual development teams. With today's complex regulatory landscape and short windows of competitive opportunity, however, these choices now have a broader impact. Initiatives aimed at reducing the cost, time, and risk of certification have become essential activities for business leaders. Understanding the business impacts can help executive teams take a more informed role in those decisions.

The increasingly complex civil aviation regulatory landscape

Airworthiness is the verified and documented capability of an air system configuration to safely attain, sustain, and terminate flight in accordance with the approved aircraft usage and operating limits. International authorities including the FAA [2], EASA [3] and Transport Canada [4] are responsible for drafting aviation safety standards, guidance, and legislation.

Figure 1 illustrates the relationships among the primary relevant standards. It is deliberately simplistic in the name of clarity – for example, documents that concern integrated modular avionics (IMA) and multicore processors clearly have hardware implications! The diagram also only shows FAA document nomenclature, but it is similarly applicable to EASA equivalent documents.

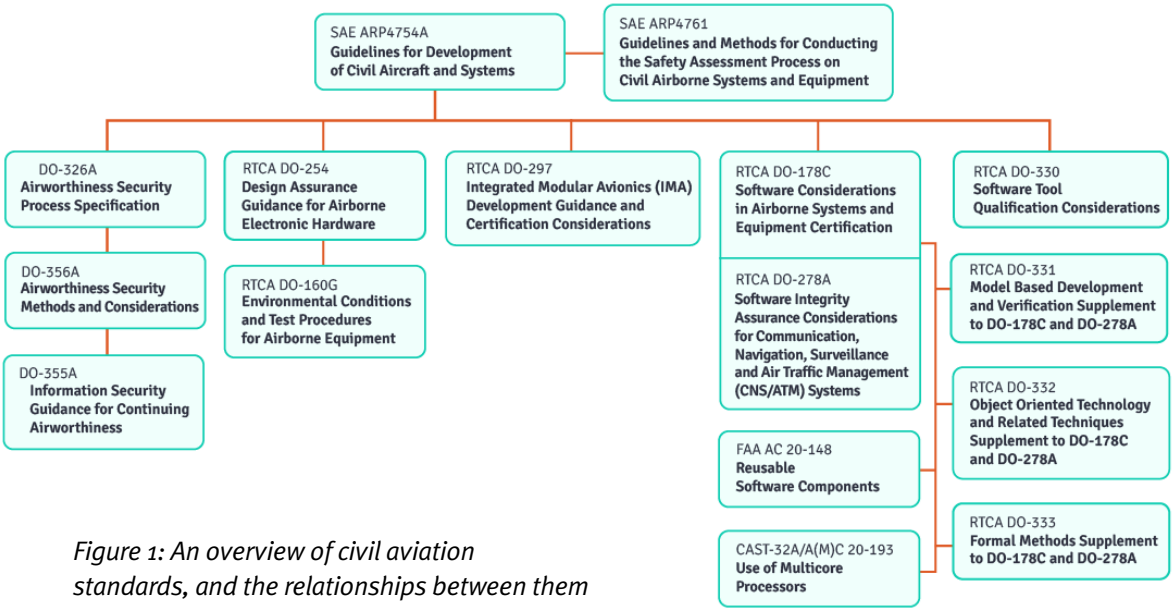

Figure 1: An overview of civil aviation standards, and the relationships between them

The regulatory landscape is also evolving more rapidly than ever before, especially in the area of software. Standards that have been in place for decades are now experiencing significantly more frequent updates, and multiple new standards have recently been released or are in discussion. As a result, the cost, time, and risk of compliance with these standards will be a major factor in determining the success of a new aircraft (Figure 2).

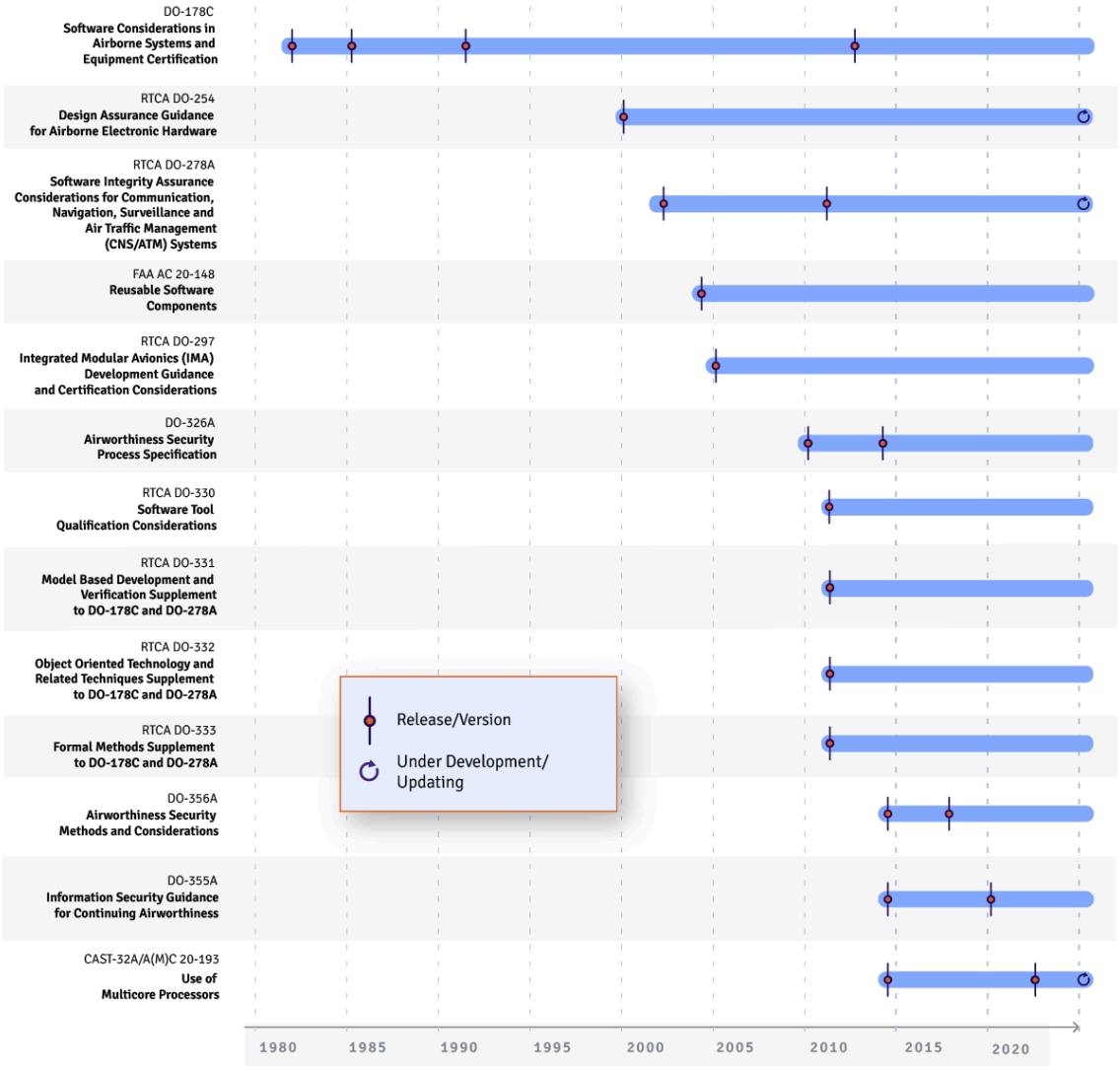

Figure 2: Both the number and the rate of change of civil aviation standards related to software are increasing.

The civil aircraft development lifecycle

Aerospace Recommended Practice ARP4754A [5] is the overarching document that addresses the certification of the aircraft in its entirety. It details the development processes which support certification of aircraft systems, addressing "the complete aircraft development cycle, from systems requirements through systems verification."

ARP4761 [6] is designed to support certification according to ARP4754A. It contains guidelines and methods relevant to the safety assessment for certification of civil aircraft.

Airborne software regulation

DO-178C [7] is the primary document referenced by certification authorities to approve all commercial software-based civil aviation systems. When it superseded DO-178B [8], DO-178C was supplemented by documents supporting the use of increasingly popular development approaches – namely model based development (DO-331 [9]), object-oriented technology (DO 332 [10]) and formal methods (DO-333 [11]).

A fourth document in the DO-33x series, DO-330 [12] deals with software tool qualification. It differs from the other three in that it is a standalone standard and not a supplement to DO-178C.

DO-326A [13] deals with aircraft security, with implications for both software and hardware. It is the sole Acceptable Means of Compliance (AMC) for cybersecurity airworthiness certification, and the core document of the Aerospace Security Framework. DO-356 [14] is supplemental to DO-326A and details security objectives that are to be met at each stage of development, along with airworthiness risk assessment processes and the evidential artefacts required.

Advisory circulars and position papers

CAST-32A [15] was the last position paper to be produced by the Certification Authorities Software Team [16]. It will be deprecated when its successor advisory documents AC 20-193 [17] and AMC 20-193 (known collectively as A(M)C 20-193 [18]) have both been published.

CAST-32A & A(M)C 20-193 deal with the unique certification challenges associated with the use of multicore processors, especially with reference to the allocation of resource associated with execution timing.

Another advisory circular, AC 20-148 [19], deals with the certification of software intended for reuse across multiple systems.

Ground-based software regulation

DO-278A [20] has a primary focus on software in ground-based Communication, Navigation, Surveillance and Air Traffic Management (CNS/ATM) Systems. As the diagram implies, it is largely identical to DO-178C aside from an additional section to discuss the application of COTS (Commercial Off-The-Shelf) software.

How tools help to overcome regulatory challenges

Software verification typically requires at least as much time, effort, and resources as the entire planning and development processes combined [21]. That makes testing and certification costly activities.

Compliant system development involves audits of development and verification activities both for new systems and for product upgrades. Unfortunately, many project teams put more focus on the outcome of individual audits and milestones than the software development and verification process itself. This short-sighted approach inevitably results in suboptimal software in the best case, and more often results in software failures. Successful development teams take a bigger-picture approach that addresses the entire software development lifecycle to ensure effective communication and knowledge transfer through every stage.

The move to an integrated suite of automated software development and verification tools can have a dramatic impact on these efforts. This can be especially important for remote and geographically dispersed teams who need to aggregate information over the course of development and testing or rework, during which manual processes or disparate tools can introduce inefficiencies and potential.

Figure 3 illustrates a common theme that runs throughout aviation standards, integral processes. The DO-178C integral processes, Software Configuration Management, Software Quality Assurance, Software Verification, and Certification Liaison, all span the entire lifecycle:

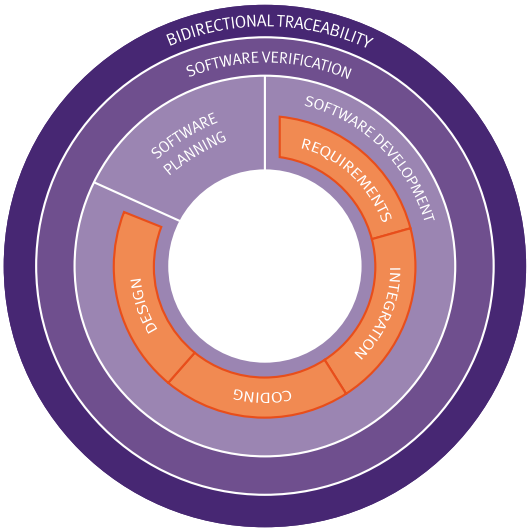

Figure 3: Regulatory guidelines have a common theme of ongoing verification and requirements traceability.

The **software configuration management** process fulfils of DO-178C's change management and baseline/storage objectives as detailed in the project's Software Configuration Management Plan (SCMP).

The **software quality assurance** process involves the application of the methods and the creation of associated records to satisfy the DO-178C quality assurance objectives in accordance with the project's Software Quality Assurance Plan (SQAP).

The **software verification process** including the production of Trace data, involves the use of reviews, analyses, and tests to satisfy DO-178C objectives in accordance with the project's Software Verification Process Plan (SVPP).

Certification liaison services provide an interface between the developing organisation and the certification authorities in relation to systems, safety, security, and electronic hardware and software.

Cost, time, and risk: The business impacts of airworthiness certification

While development teams will ultimately make technology decisions, business leaders have an opportunity to apply their big-picture perspective to manage the cost, time, and risk inherent in product development and certification. One of the clearest, proven ways to reduce cost, time, and risk is to identify and address issues as early as possible in the development process. Individual teams may feel comfortable with their existing software tools and processes – but the old ways of working are no longer effective in today's changing landscape.

The increasing complexity and volume of compliance requirements make errors more likely. Trying to address them without integrated, automated tools is no longer possible. Disparate groups must be able to work collaboratively and in parallel to reduce development and certification time. Significant advantages can be gained with the use of an integrated development and verification tool chain across groups and product lines.

The commercial benefits of an integrated tooling solution

The move to an integrated suite of automated software development and verification tools can result in dramatic improvements. This can be especially important for remote and geographically dispersed teams who need to aggregate information over the course of development and testing or rework, during which manual processes or disparate tools can introduce inefficiencies and potential for error.

Unlike spreadsheets or stand-alone document-management systems, integrated tools offer full visibility and change-impact analysis across projects and teams, enabling better design decisions more quickly. An ideal set of verification tools provides a broad range of capabilities including requirements traceability, change impact analysis, test management, coding standards compliance, code quality review, code coverage analysis, data- and control-flow analysis, unit/integration/system testing (including target testing), along with the automated generation of certification evidence.

Such tools lend themselves equally to any structured lifecycle model, including Agile and the V model as illustrated in DO-178C. They also lend themselves to a continuous integration workflow. Verification tools are an important part of the continuous integration (CI) workflow, allowing teams to use the same tools throughout the development process for rapid, iterative software development, verification, and deployment. Applying CI, developers can run static analysis and unit testing as part of their front-end integration, to make sure their code is doing what it's intended even before it's merged with other code bases.

Cybersecurity in avionics systems has become a constantly moving target with serious implications for safety, making it a significant driver of business risk and uncertainty. The recommended strategy to thwart cyberattack is to design in security so that vulnerabilities are minimised during development. Subsequent testing of the completed software can then help to prove the efficacy of that approach before the product is put into the field. Once there, a strategy to address any newly exposed vulnerabilities quickly and safely becomes important.

To address this challenge, many teams are moving to DevSecOps (development/ security/ operations), a CI/CD approach to help reduce costs and risk and improve efficiencies. In contrast to the traditional handoff to security teams after software is completed, this "shift left" enables software development and security teams to work efficiently and cost-effectively together, in parallel. Flexible and customizable software tools easily adapt to the level of risk and necessary rigour of mitigation, and requirements traceability tools enable a rapid response for dealing with a compromised vulnerability even for systems that have been unchanged for years.

This requirement to deal with both safety and security illustrates why it is important for verification to be extensible for specific needs. For example, the tools should support the development and verification of software that needs to achieve DO-178C and DO 326A certification in parallel (Figure 4). In this figure, the security process described in DO-326A has been aligned with the aircraft development process described in ARP4754A, showing how developers can follow a hybrid safety and security process to satisfy both standards. Trying to achieve compliance/conformance with multiple standards sequentially increases the time and cost to develop, and it would likely result in unnecessary rework including a costly cycle of regressions and fixes.

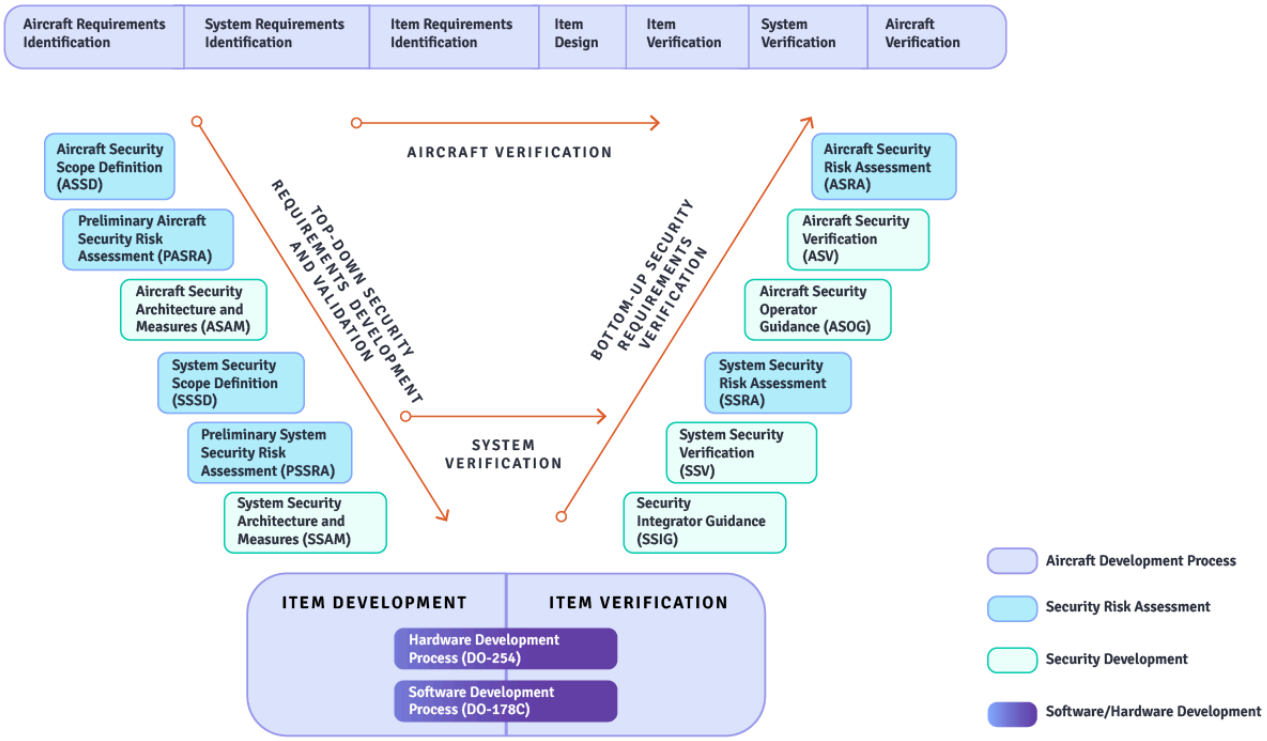

Figure 4: Security process and software development process as part of the aircraft development process [22]

Even with the support of comprehensive automation, getting to grips with airworthiness regulation can be a daunting task. Certification and regulatory support are available to provide a helping hand and give confidence that certification costs will be contained. In the US, services must be delivered by a team with experience liaising with FAA Aircraft Certification Offices (ACO) US as well as international partners. In addition to comprehensive audit support, services may include training, mentoring and the production of compliance artifacts that expedite and enhance life cycle data production.

Software development often begins before the project target hardware is available, and perhaps before it has been completely specified – an issue that is often exacerbated on smaller aircraft developed with the intent for frequent upgrade. **Hardware simulators** are usually deployed in such situations, although for critical systems with higher levels of design assurance, verification must ultimately be on the final target. Verification tools must therefore be flexible enough to support both simulation and target testing.

Highly critical software developed in compliance with DO-178C DAL A requires verification that the object code executed by the application correctly reflects the requirements and the intent of the developer. Using different tool chains for different levels of software criticality can introduce additional delays at certification. A single, flexible tool suite capable of demonstrating source code to object code traceability that can be used at any level of verification allows teams to quickly and efficiently match the level of risk identified without an additional learning curve.

Many technical leaders are incorporating new approaches such as **model-based software development** and **object-oriented analysis and design**. They allow for faster development, but new layers of abstraction mean that testing becomes critical to keep errors at bay. Verification tools should enable fast, iterative rounds of virtual prototyping and testing, either on the host development platforms or on the actual target hardware. This ensures that abstractions don't result in late discovery of errors that can cause delays at certification.

One final and very important consideration is tool qualification. Qualification of software verification tools is required for any certification exceeding DO-178C Level C and involves validating the operation of the tool in a project-specific environment. To reduce the cost associated with this qualification process, tool providers offer tool qualification packages and certification support services for programs requiring the appropriate level of assurance. Aircraft certification services encompass systems, safety, security, and electronic hardware and software.

Overcome the challenges of certifying next-generation flight software

Ultimately, no matter how well a company plans for cost, time-to-market, and risk factors, an evolving landscape hides unexpected setbacks. Those can range from supply chain and workforce shortages to cybersecurity threats and new regulatory requirements.

The avionics and aerospace industry is cautious for good reason. But in many organizations, that caution can lead to hesitation to change development and testing processes, tools, and models. Gap analysis to assess the additional processes and resource to achieve compliance allows business executives to question development and testing processes and tooling, even as standards are evolving and issues such as cybersecurity are gaining attention and enforcement.

Successful organizations understand that the future will only become more challenging, and that the time to address concerns and make fundamental changes is now—not during a future critical product launch. A flexible, robust, and scalable development process offers the best chance for successfully addressing challenges and reducing risk and uncertainty.

With a comprehensive verification tool suite boasting consistent user interface for all aspects of verification and validation, teams are well-armed with documentation and shared knowledge throughout the process. Integration with associated development tools throughout the product development lifecycle provides a foundation for overcoming the challenges of certifying next-generation flight software.

Item Description - RTCA DO-178C	RTCA DO-178C Reference	RTCA DO-178C DAL A	RTCA DO-178C DAL B	RTCA DO-178C DAL C	Supported by LDRA Tools
Test coverage of software structure (MC/DC) is achieved	6.4.4.2	✓	Not Required	Not Required	YES
Test coverage of software structure (decision coverage) is satisfied	6.4.4.2a & 6.4.4.2b	✓	✓	Not Required	YES
Test coverage of software structure (statement coverage) is satisfied	6.4.4.2a & 6.4.4.2b	✓	✓	✓	YES
Test coverage of software structure (data coupling and control coupling) is achieved	6.4.4.2c	✓	✓	✓	YES
Verification to additional code, that cannot be traced to source code, is achieved (Object Code Verification)	6.4.4.2d	✓	Not Required	Not Required	YES
Static analysis: software coding standards	11.8	✓	✓	✓	YES
Static analysis: non-conformant code highlighted	6.4.3d	✓	✓	✓	
Static analysis: uninitialised or unused variables and/or constants identified	6.4.3f	✓	✓	✓	YES
Table A-7.1 Test procedures are correct Table A-7.2 Test results are correct and discrepancies are explained	6.3.6b & 6.3.6c	✓	✓	✓	YES
Qualified tool per DO-17BC/DO-330	12.2.1	✓	✓	✓	YES

A(M)C-2D-193/CAST-32A Software Verification Objectives	RTCA DO-178C DAL A/B/C	Supported by LDRA Tools
MCP_Software_1: verify that all the hosted software components function correctly and have sufficient time to complete their execution	✓	YES
MCP_Software_2: verified that the data and control coupling between all the individual software components has been exercised during software requirement-based testing and is correct	✓	YES

Future Airborne Capability Environment (FACE) Conformance Activities	Supported by LDRA Tools
Manage traceability between the FACE Conformance Verification Matrix (CVM) requirements and software verification package artifacts	YES
Automate the filtering and generation of the FACE CVM and FACE Conformance Statement	YES
Automate the verification of compliance with FACE CVM Fully Tested Requirements using static code analysis	YES

Security Assurance Level	Severity Recommended	Recommended	w. Independence	As Negotiated
3	Catastrophic/Hazardous	5	2	3
2	Major	5	0	3
1	Minor	0	0	8
0	None	0	0	0

Figure 5: Support for airworthiness standards and guidelines from LDRA tools

About LDRA

For more than 45 years, LDRA has driven the market for software that automates code analysis and software testing for safety- and security-critical applications. Working with clients to achieve early error identification and elimination, and full compliance with industry standards (Figure 5), LDRA traces requirements through static and dynamic analysis to unit testing and verification for a wide variety of hardware and software platforms.

Today's LDRA continues to develop and drive the market for software tools that automate code analysis and software testing for safety-, mission-, security-, and business-critical markets. In tandem with our customer-focused certification services and consultancy offerings, LDRA tools achieve early error identification and elimination, tracing requirements through static and dynamic analysis to unit testing and verification for a wide variety of hardware and software platforms.

Works cited

[1] Aerospace Vehicle Systems Institute (AVSI), "Exponential Growth of System Complexity," 2023. [Online]. Available: https://savi.avsi.aero/about-savi/savi-motivation/exponential-system-complexity/. [Accessed 13th April 2023].

[2] Federal Aviation Administation (FAA), "Federal Aviation Administation (FAA)," [Online]. Available: https://www.faa.gov/. [Accessed 3 November 2021].

[3] European Union Aviation Safety Agency (EASA), "European Union Aviation Safety Agency (EASA)," 2021. [Online]. Available: https://www.easa.europa.eu/. [Accessed 3 November 2021].

[4] Government of Canada, "Transport Canada," 19 April 2023. [Online]. Available: https://tc.canada.ca/en. [Accessed 28 April 2023].

[5] SAE International: S-18 Aircraft and Sys Dev and Safety Assessment Committee, ARP4754A: Guidelines for Development of Civil Aircraft and Systems, SAE International, 2021.

[6] SAE International: S-18 Aircraft and Sys Dev and Safety Assessment Committee, ARP4761: Guidelines and methods for conducting the safety assessment process on civil airborne systems and equipment, SAE International, 1996.

[7] RTCC, DO-178C "Software Considerations in Airborne Systems and Equipment Certification", RTCA, 2011.

[8] RTCA, DO-178B, "Software Considerations in Airborne Systems and Equipment Certification", RTCA, 1992. [9] RTCA, Inc., RTCA DO-331 Model-Based Development and Verification Supplement to DO-178C and DO-278A, Washington DC: RTCA, 2011.

[10] RTCA, Inc., RTCA DO-332 Object-Oriented Technology and Related Techniques Supplement to DO-178C and DO-278A, Washington DC: RTCA, 2011.

[11] RTCA, Inc., RTCA DO-333 Formal Methods Supplement to DO-178C and DO-278A, Washington DC: RTCA, Inc., 2011.

[12] RTCA, Inc., RTCA DO-330 Software Tool Qualification Considerations, Washington DC: RTCA, Inc., 2011.

[13] RTCA, Inc., DO-326A Airworthiness Security Process Specification, Washington, DC: RTCA, Inc., 2014.

[14] RTCA, Inc., RTCA DO-356 Airworthiness Security Methods and Considerations, Washington DC: RTCA, Inc., 2018.

[15] Certification Authorities Software Team (CAST), Position Paper CAST-32A - Multicore processors, CAST, 2016.

[16] Federal Aviation Administration (FAA), "Certification Authorities Software Team (CAST)," 18 August 2020. [Online]. Available: faa.gov/aircraft/air_cert/design_approvals/air_software/cast/. [Accessed 11 November 2021].

[17] EASA, "AMC 20-193 Use of multi-core processors," 2022. [Online]. Available: https://www.easa.europa.eu/sites/default/files/dfu/annex_i_to_ed_decision_2022-001-r_amc_20-193_use_of_multi-core_processors_mcps.pdf. [Accessed 18th January 2023].

[18] F. Wolfe, "EASA and FAA to Issue Further Guidance on Multicore Certification This Year," 28 February 2020. [Online]. Available: https://www.aviationtoday.com/2020/02/28/easa-and-faa-to-issue-further-guidance-on-multicore-certification-this-year/. [Accessed 4 November 2021].

[19] Federal Aviation Administration, "AC 20-148 - Reusable Software Components Document Information," 7 December 2004. [Online]. Available: https://www.faa.gov/regulations_policies/advisory_circulars/index.cfm/go/document.information/documentID/22207. [Accessed 22 September 2021].

[20] RTCA, Inc., DO-278A, "Software Integrity Assurance Considerations for Communication, Navigation, Surveillance and Air Traffic Management (CNS/ATM) Systems, Washington DC: RTCA, Inc., 2011.

[21] G. P. Brat, "Reducing V&V cost of flight critical systems: myth or reality?," in SciTech forum, Grapevine, Texas, 2017.

[22] C. Torens, "Safety versus Security in Aviation, Comparing DO-178C with Security Standards," in AIAA Scitech 2020 Forum, Orlando, FL, 2020.

LDRA UK & Worldwide
Portside, Monks Ferry,
Wirral, CH41 5LH
Tel: +44 (0)151 649 9300
e-mail: info@ldra.com

LDRA Technology Inc.
2540 King Arthur Blvd, 3rd Floor, 12th Main, Lewisville, Texas 75056
Tel: +1 (855) 855 5372
e-mail: info@ldra.com

LDRA Technology Pvt. Ltd.
Unit B-3, Third floor Tower B, Golden Enclave
HAL Airport Road Bengaluru 560017
Tel: +91 80 4080 8707
e-mail: india@ldra.com