Challenges in Low-Level Integration Testing of T2080 in Integrated Modular Architecture

*Ezhil Venthan K, Manju Nanda, Monisa M, Anand G**

CSIR-National Aerospace Laboratories, Kodihalli, Bengaluru, Karnataka, India, 560017

Abstract. The aircraft's mission system is subject to continual revision by removal or obsolete upgrades. In addition to user and integration requirements, security considerations have become a major driving force in the design of today's avionics systems. To achieve this, system design uses functional risk analysis and abstract system description to provide a coherent design that meets certification needs. The objective of the study was to spot evaluation criteria that may allow developers and certification bodies to gauge specific real-time security-critical software development tools from a system security perspective. The report clarifies the context of software development tools in relevancy current aeronautical systems certification guidelines. Research efforts continue in two directions: (1) collecting data on instrument quality assessment efforts to consider possible future revisions of existing guidelines, and (2) create a classification for evaluating software development tools by identifying tool categories, characteristics, concerns, factors, and evaluation methods. The topic report has four components: (1) industry perspective, (2) expertise, (3) quality assessment, and (4) instrumental assessment. Data collected from the industry influences the assessment process and recommendations for development tooling practices. Several methods of instrument evaluation are described. The report presents the different types of tools identified during the research. This classification is prohibited within the scope of studies directed by DO-178C DAL standard. Finally, the report defines the tool evaluation structure and organization.

Keywords: Low Level Testing, system architecture, certification, safety, design, avionics, requirements, verification and validation.

1. Introduction

The LDRA tool suite offers unit testing and requirements engineering as well as a full range of static and dynamic software analysis. The LDRA Toolkit is a fully integrated solution that enables customers to build quality software from request to delivery. A few years ago, LDRA created and led the market for software that automates code analysis and software testing for safety, mission, security, and critical operations. LDRA works with customers to ensure early detection of defects and full compliance with industry standards,

*Corresponding Author: g.anand777@nal.res.in

tracking requirements through static and dynamic analysis, unit testing, integration testing, and validation. Multiple hardware and software platforms. By implementing LDRA's tools, customers can deliver well-built, documented, and tested software and, at the same time, provide significant time, cost, and operational savings to their customers. their business. LDRA Testbed with TBvision provides static and dynamic analytics, as well as visualizations, to easily understand and navigate compliance with standards, quality metrics, and code coverage analysis. The LDRA Testbed component of the LDRA Toolkit provides key static and dynamic analysis tools for other components. Test LDRA includes a proprietary scanning engine, not a commercial or open source scanning engine. This proprietary technology allows LDRA to quickly and easily incorporate new analytical techniques needed to meet new and evolving standards, keeping the LDRA toolkit at the forefront of technology. The TBvision component of the LDRA toolkit allows users to view coding standard violations and software bugs in the context of the original source code. An interactive environment that allows you to perform static and dynamic analysis with custom scope, allowing you to switch between reported violations, native code, and any LDRA-supported coding standards Testbed. Software integrity can also be measured and reported for quality and safety, or simply the presence of bugs (including dynamic memory errors). TBvision presents software defined from either of these angles, identifying problems that need to be addressed to ensure that the software project achieves its goals.

2. Objectives

Main purpose of this research project was to identify evaluation criteria that would allow both developers and certification bodies Evaluate specific real-time software development tools important to system and software security. . A related objective is to present and evaluate modern software development tools used in the development of safety-critical real-time systems and to form the basis for future certification guidelines. for software development tools.

3. LDRA’s software testing solutions expedite your certification/approval process through

- Automating software quality analysis, verification, requirements traceability, and standards compliance.
- Traceability of requirements, design and development artifacts, validation and verification throughout the software development lifecycle.
- Complies with encryption standards such as MISRA and CERT.
- Automate unit tests and integration tests.
- Produce and report coverage analysis report.
- Implement and plan test based on requirements.
- Toolchain integration.
- Tool norm packages. Automated production of software certification and approval certificates. Reduce critical software development costs and schedule risks.

3.1. LDRA tool suite

- Aggregate requirements data from multiple sources, including various ALMs and requirements management tools (DOORS, DOORS NG, Polarion, Word, Excel, etc.).
- Link processes in two directions from requirements to code, test cases and test results.

- Faster analysis of the impact of changes in requirements, code or test cases Automate the execution of test cases (static analysis, unit testing, system testing)
- Enable regression testing from the command line.
- Collect data and automatically generate reports to demonstrate compliance.

Static analysis techniques in the LDRA tool suite include data flow and control flow analysis. They offer the possibility to explore and visualize the architecture and structure of the source code. It's almost impossible to fully understand and document the coupling of controls and data in source code just by looking at the code itself. A static analysis tool automates this error-prone task to create a readable representation of the coupling. Dynamic analysis techniques allow you to test the functionality of your code and optionally expose the parts of your code exercised by those tests. TBvision provides the ability to instrument and run a copy of the source code under test, enabling structural coverage measurements. Structural coverage analysis shows how well the source code is tested, leaving the possibility of unintended or unexpected functionality in the code.

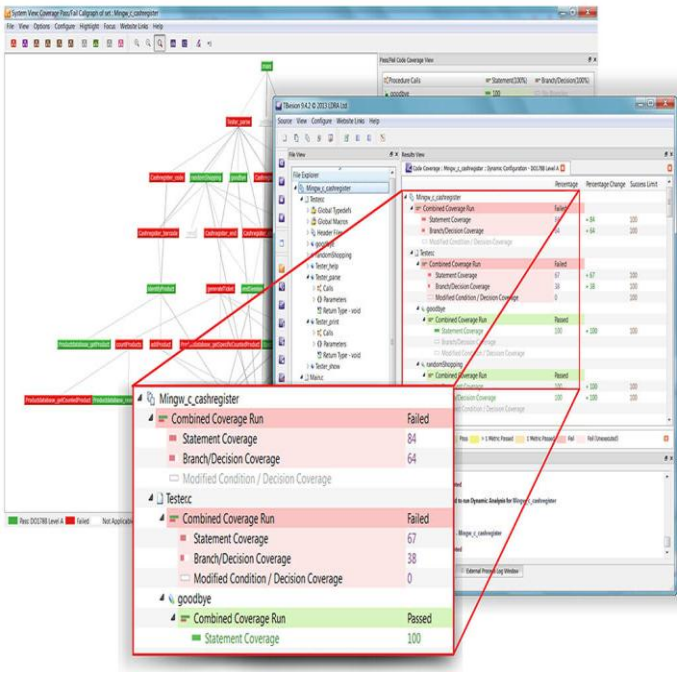

Fig. 1. Flowchart with Code coverage in LDRA TBrun

3.2. Operational Needs

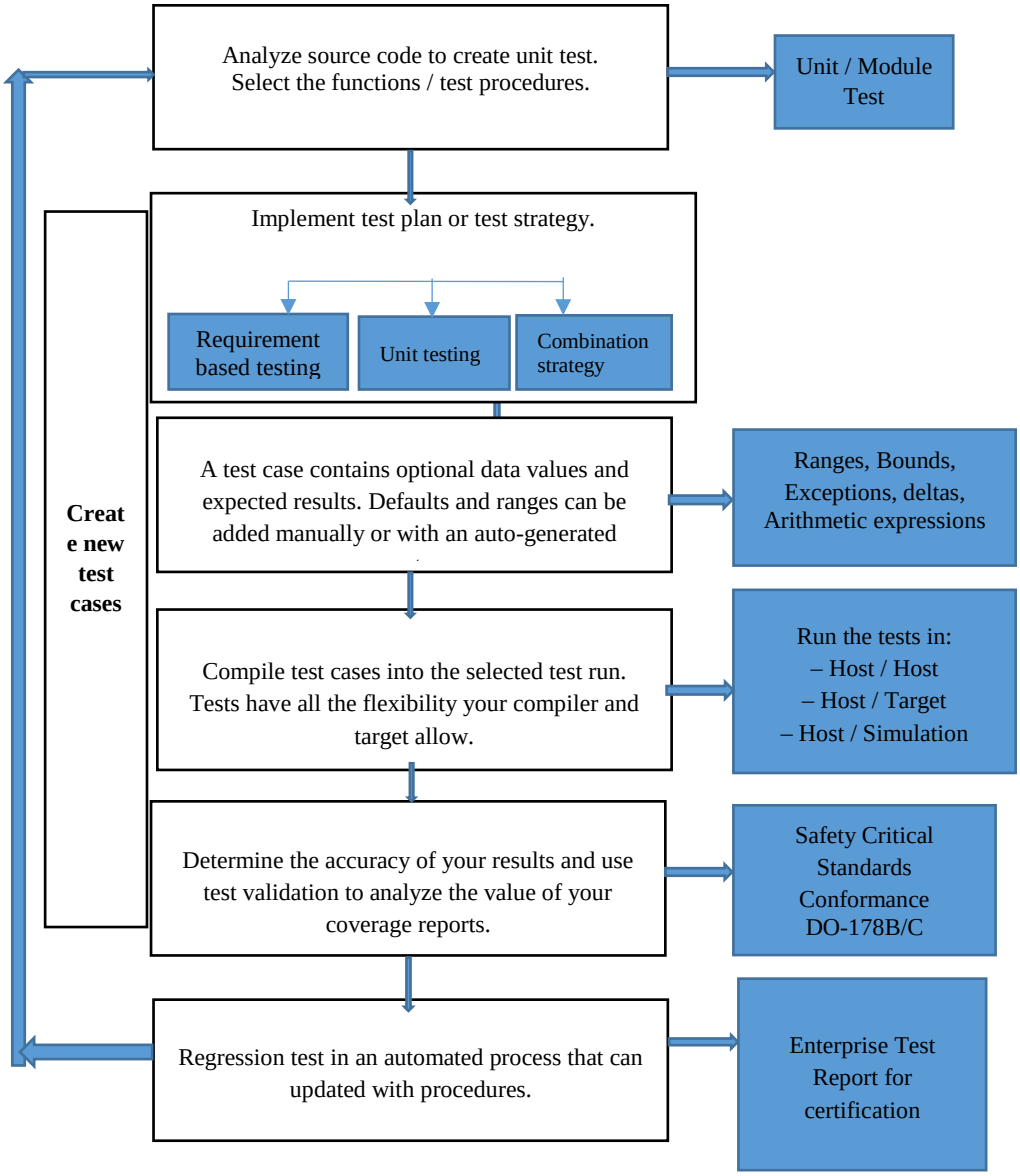

Fig. 2. Code Test Procedure

- Generation of wire harnesses or automated test drivers without manual scripting.
- High level testing with intuitive GUI and command line options.
- Automated analysis features reduce testing effort and make it easier for developers and testers.
- Data maintenance and archiving and test results are fully automated regression analysis.

- Automatically detect and document source code changes.
- Create tool-driven test vectors.
- Run tests on host, compiler, target and simulator.
- Automatically generate test case documentation including pass/fail and regression analysis reports.

3.3. Test case files and test case management and storage

TBrun stores groups of test cases as strings. User can search sequence and test case by & # 40; TCF) contains all the information needed to run the test case again. TCFs can be packaged and stored with regression reports for regression testing, stored with source files through a Software Configuration Management (SCM) system, or can be used to store requirement-based test data storage, including selected specific values, and requirements management. Can be used as a resource. System input variables are available for additional storage. As an SCM file, administrators can see developers test code directly from their SCM system. TCF can also be rerun in batch mode from the command line to modify the source code and check the module interface and output. For companies engaged in outsourced development management, TCF files are easily distributed and provide a standard template worldwide.

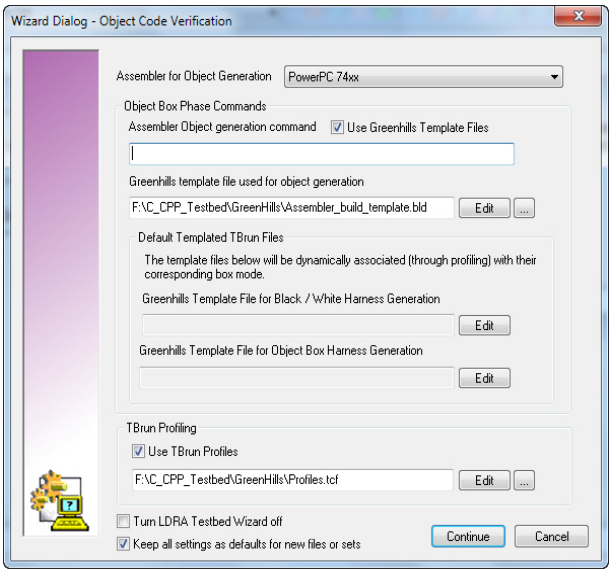

Fig. 3. Object Code Verification wizard

3.4. Assembly Level Coverage/Object Code Verification

TBrun can be used with the LDRA Toolkit Object Code Review feature to provide and apply test cases that ensure complete coverage of code at the assembly level. This feature is commonly used for safety-critical development requirements such as the DO-178B/C to add MC/DC coverage and support operations such as source code to object code mapping. These features enable comprehensive analysis and identification of codes of complex logical conditions.

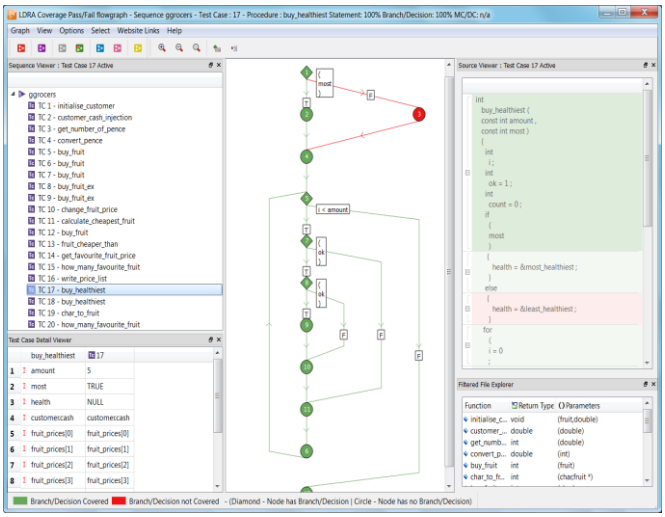

Fig. 4. Test Case flow graph

	Percentage	Success
Tenpercentoff.cpp		
Combined Coverage Run	Failed	
Statement Coverage	99	100
Branch/Decision Coverage	94	100
Modified Condition / Decision Coverage	50	100
main		
Combined Coverage Run	Passed	
Statement Coverage	100	100
Branch/Decision Coverage	100	100
Modified Condition / Decision Coverage		
TenPercentOff		
Combined Coverage Run	Failed	
Statement Coverage	100	100
Branch/Decision Coverage	100	100
Modified Condition / Decision Coverage	67	100

Fig. 5. Test Coverage view

3.5. Structural Coverage Analysis

TBrun has access to the full range of range indicators available in the LDRA Toolbox. This includes procedural appeals, statements, separations/decisions, and MC/DCs. The user can select an appropriate metric or set of metrics based on security and programmatic limitations. For example, MC/DC coverage is needed to ensure results are not obscured by input conditions, and coverage provides a comprehensive metric for evaluating loops. All of these metrics are available as graphs through TBrun's Flowchart Viewer, Call Graph Viewer, and File View/GUI. Users have direct access to compliance reports that provide overall pass/fail metrics for standards such as DO-178B/C. The line-by-line view shows statements, branches, and controls that have executed and also appear in these reports.

3.6. Software Errors

- There are two types of errors that can be found:
  - Technical errors – These are due to language and programming and can be found by performing static analysis
- Application errors – These are due to the code not implementing correctly the requirements – These can only be found by performing dynamic analysis on the actual hardware and measuring structure coverage.

3.7. Control flow Analysis

Control Flow Analysis - software correctness through logical control diagrams. Examine your logic flow to identify missing, incomplete, or incorrect requirements. Validate that the flow of control between functions is the correct solution to the problem.

3.8. Data flow Analysis

- Use of data flow diagrams
- Flow balancing
- Compare output data and subsequent input and derived data from each process to ensure data is available when its required
- Confirmation of derived data
- Involves examining derived data within a process for correctness and format
- Compares the data keys used to retrieve data from the in-process data stores to the database index design to ensure that invalid keys are not used and uniqueness properties are consistent.

4. Tool Qualification

- The purpose of the tool qualification process is to gain confidence in the functionality of the tool.
- LDRA's tool qualification packages are tailored per standard and software integrity level.

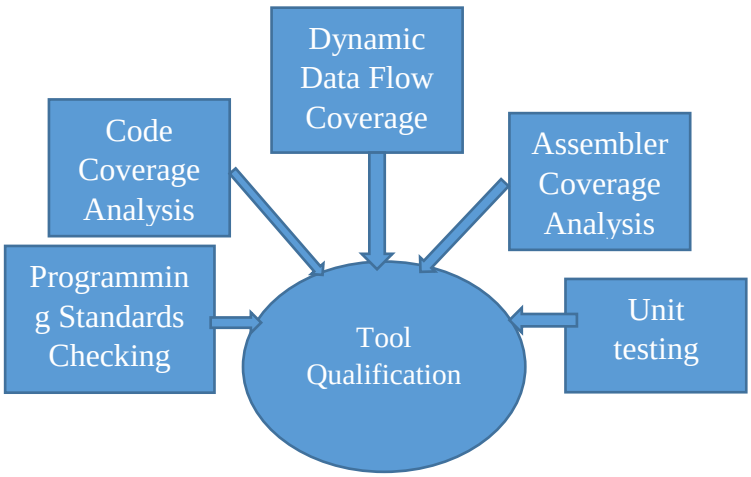

Fig. 6. Tool Qualification flowchart

5. Stepping Up to the DO-178C Certification Challenge

The objectives of the software verification process are defined in the DO-178C standard. As explained, testing approaches are considered at three levels. Standard DO-178C: low level test, software integration test and hardware/software integration test. Low level testing: Low level testing is used to test low level requirements and is usually done with a set of unit tests that isolate a single unit of source code. Vector CAST is used in this testing phase. Software integration testing: Software integration testing verifies the interrelationships of components. This level of testing is performed using Vector CAST to simultaneously test multiple software components under test. To support this type of testing, a complete test framework is automatically generated to map software requirements to specific test cases, ensuring that all requirements are tested. Hardware/Software Integration Testing: Hardware/Software Integration Testing is the process of testing a computer software component (CSC) for high-level functionality in a target hardware environment. The purpose of hardware/software integration testing is to test the behavior of software integration developed on hardware components.

6. LDRA test reviews provide the following coverage metrics

- Description
- Decision /Enactment
- Function/ Procedure call
- Branch decision coverage
- Combining branching conditions
- Modified Condition / Decision Coverage (MC/DC for DO-178B/C Level A)
- Dynamic data flow

6.1. LDRA Tool Suite Facilitates Design and Customer Collaboration

- Traceability of software changes. It identifies and tracks critical source code changes, enabling project teams to closely monitor the impact of these changes on testing processes and code analysis.
- Detect changes in lifecycle assets such as requirements, design and code to assess impact and optimize regression testing and resolution of suspicious links.
- Unit/module test recognition, validation and regression.

"When we work with multi-billion-dollar aerospace companies that also use LDRA tools, it's much easier to collaborate," "For example, we're working on the design of a toolkit. We centralize data remotely with large avionics and can easily push data to our teams. This is not an unusual way of working, but the more you can collaborate the more you can collaborate., the easier it will be for you to efficiently complete your design and meet certification requirements.

6.2. Unit Testing

Unit test tools are used by developers to test and retrieve the application source code.

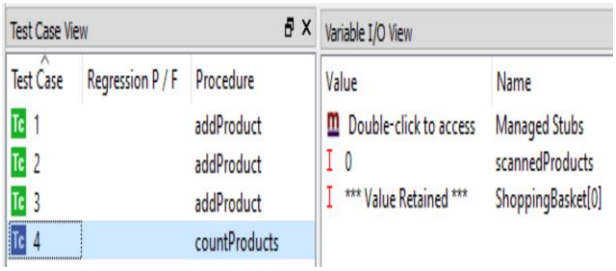

Fig. 7. Test Case Input / Output view

We should get Coverage as shown:

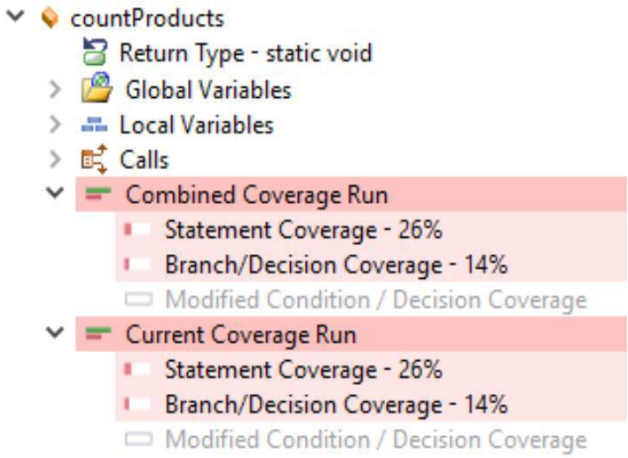

Fig. 8. Test run Coverage

6.3. Flow Graphs

A flowchart is a graphical representation of logic executed with zero or more filters (input parameters) to obtain an output.

"A signal flow graph is a network of nodes (or points) connected by directed edges that represent a set of linear algebraic equations.

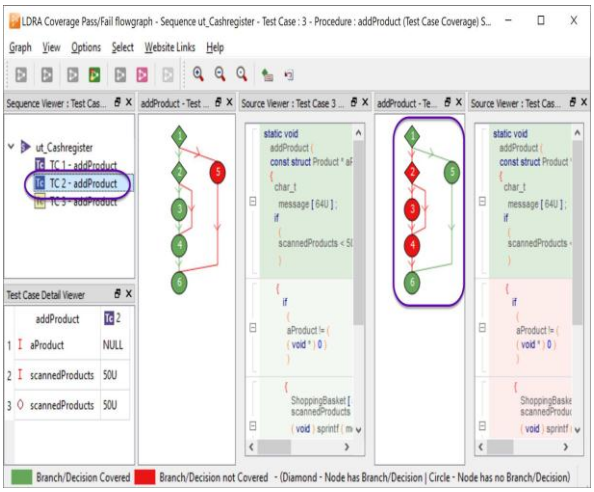

Fig. 9. Code Flow graph

6.4. LDRA testing

The test is not included in the project as there is no resource to any of the equipment. However, to successfully conduct the test, the following things should be considered during the design process.

1. Define the Program
2. Define the Objectives of the Test.
3. Define the requirement of the test (assets, equipment, instruments, data, etc.).
4. Provide test cases to cover most operating conditions.

In the test process, an Exhaustive testing is not practical as the system complicity is beyond the practice of exhaustive attempt. However, the specific exhaustive test may be conducted using all combinations of: data state; data bus state; signal value; pilot input; aircraft response. It is impossible to test, but a series feasible way is developed.

For the hardware life-cycle, an accelerated testing is suggested, as it proves the MTBF for the systems, which did not specify the MTBF from the suppliers. The accelerated test may involve following test methods.

1. Thermal loads, to accelerate the life cycle of semiconductors
2. Chemical loads, to simulate salty, humidity and acid corrosion
3. Electrical loads, from voltages and currents
4. Mechanical loads, including vibrations, shocks and mechanical load cycles

6.5. LDRA rules

LDRA rules is a cost-effective, standalone rule checker that requires no investment in a full toolchain. LDRA rules incorporate next-generation reporting capabilities for code quality metrics, error detection, and avoidance. Users can quickly and easily visualize results in call graphs, flowcharts, and code review reports in an intuitive, easy-to-read format, improving collaboration and communication among all team members in the development team. LDRA rules continues to improve automated code testing and analysis tools for very

complex and safety-critical markets common in industrial sectors such as aerospace, automotive, energy, industrial control, medical, military and transportation. The LDRA Toolkit is a fully integrated solution that enables customers to build quality for their software from requirements through deployment.

7. Verification and Validation

Verification: Inspect and test the product with all expected functionality and performance. Detailed verification must be performed using the list of requirements. A set of tools, e.g. IBM DOORS, IBM LDRA, RTRT can assist in the verification process. Validation: Test the product to ensure that it meets the design intent and functions correctly according to the requirements collected during the requirements capture phase. The results of the COTS screening and system security assessment have satisfied the requirements of customers and other system designers. The first print system confirmation is confirmed. However, the detailed system specifications could not be confirmed due to time constraints.

8. Target markets and applications in T2080 into an Avionics Architecture

The T2080 and T2081 processors are intended for mid-range drones or mixed data flight applications. A highly efficient 8-core virtual appliance achieves 1.8 GHz while maintaining a short 7-stage pipeline for better latency response to unpredictable control plane code branches. Advanced virtualization technology facilitates secure data plane partitioning in instrumentation and control applications.

- **Enterprise Equipment:** Ethernet Modular Switches, Service Cards, UTM Devices, Enterprise Storage, Data Center
- **Service Providers:** Core and Edge Routers, Tape Access Broadband, Metro Ethernet, Optical Network
- **Wireless infrastructure:** mobile backhaul, network card, channel card, LTE controller card, WCDMA, GSM, WiMAX
- **Aviation and defense:** Hard or high security routers, avionics networks, control panels, military SBCs
- **Industrial computer:** SBC, industrial automation, test and measurement

9. Conclusion

- Potential for software errors increases in proportion to software complexity. Best-practice development techniques help to limit them. These include the application of a state-of-the-art coding standard such as MISRA C: 2012, measuring metrics on the code, tracing requirements, and implementing requirements-based testing.

- The Testing helps to compliance and conformance with standards including DO-178B/C, DO-331, ISO 26262, IEC 61508, IEC 62304, EN 50128, and SAE J3061, ARP 4754A, and Future Airborne Capability Environment (FACETM).

- In source code Beamer ALM and LDRA's solutions combine to create a powerful tool in the development and test verification of critical software systems. Whether adhering to the V-model process or deploying an Agile, Spiral, Waterfall or Devops software development approach.

- The standard recommends these practices because experience has shown that they are the most effective ways to achieve high quality, reliable, and robust software. And whether or not a product is critical to safety is certainly a result that benefits all parties involved.

9.1. Summary of Standalone Products

- Don't have to invest in a complete tool chain. LDRA unit, LDRA rules, and LDRA cover are part of a family of certification technologies emerging from LDRA's 40 years of experience in helping customers achieve certification readiness. These tools support certification objectives at all levels of design assurance:

- LDRA unit supports automated test generation and management with sophisticated analysis that helps developers to eliminate manual scripting and minimize overall test effort. LDRA unit auto-generates test-case documentation, including pass/ fail and regression analysis reports, and simplifies the viewing of statement coverage, branch coverage, and MC/DC coverage results.

- LDRA rules offers standards-based rule checking that confirms adherence to company and industry coding standards. With the help of call graphs, flow graphs, and code review reports, the tool clearly demonstrates code quality, fault detection, and security vulnerabilities.

- LDRA coverage provides a test plan document that demonstrates the completeness of application testing and helps developers achieve the desired level of coverage. This ranges from procedure and function calls to safety-critical MC/DC coverage.

References

1. New challenges for future avionic Architectures by P.Bieber, F.Boniol, M. Boyer, E.Noulard,C.pagetti (Onera) publication at **https://www.researchgate.net/publication/244484691**
2. Assessment of Software Development Tools for Safety-Critical, Real-Time Systems by U.S. Department of Transportation Federal Aviation Administration.
3. Design of Avionics Integration Architecture (IMA) and Data Network (AFDX) publication at: https://www.researchgate.net/publication/339509102

4. Civil Aircraft Advanced Avionics Architectures – an Insight inta SARAS Avionics, Present and Future Perspective by C.M.Ananda Aerospace Electronics Systems Division National Aerospace Laboratories **ananda** cm@css.nal.res.in
5. Hard Real-Time Delay (RTD) Estimation and Analysis for Safety – Critical System by Anand G, Aerospace Software & Information Systems Division CSIR-NAL Banglore, India.
6. LDRA Coverage News from the world of testing Autumn/Winter 2007 Issue No. 10. and Spring 2014 Issue No.19.
7. LDRA Increase Productivity with Automated Unit/Integration Testing with TBrun.
8. Verification Techniques for Better Code and Higher Productivity mark.richardson@ldra.com
9. LDRA Software Technology Technical Note LDRA tool Suite and intland code beamer ALM Path to IEEE 1012.
10. Challenges of Certification and Integration of new hardware into legacy avionics architectures by Dr. Andreas Zeitler TOR Architecture & Integration Airbus Defence and Space GmbH Manching, Germany Andreas.Zeitler@airbus.com
11. The principles of Integrated Technology In Avionics Systems by Guoqing Wang, Wenhao Zhao.
12. LDRA Getting Started Tutorial Static/Dynamic Analysis and Unit Testing Using TBvision & TBrun.