DO-178C compliance

Making the skies safer for software to fly

The companies designing and building safety-critical systems need robust processes and collaborative platforms. This paper introduces best practices for software development leveraging the DO-178C standard, and how the IBM Engineering Lifecycle Management solution can help organizations deliver safety-critical products, improve collaboration, and increase efficiency and profitability.

> The cost of designing and building safety-critical systems is dramatically increasing.

The aerospace and defense (A&D) industry, is seeing new levels of innovation and disruption at every turn[1] – putting significant pressure on A&D companies to do more with less by optimizing their development processes to meet cost pressures. Software development and testing alone may be a significant factor in these rising costs, and the DO-178C standard and its related technology supplements have the potential of adding even further stress if not handled optimally.

Projects that need to comply with DO-178C standards could see cost increases anywhere from 25 percent to 40 percent compared to projects that don't require compliance.[2]

The sources of additional impacts may include the following:

- Reduced developer productivity due to increases in process complexity
- Manual reporting and documentation processes that are not suited to the level of detail required to comply with DO-178C
- Qualification activities involved in compliance

IBM **Engineering**

DO-178C overview

DO-178C provides guidance for developing aviation software systems to ensure that they perform their intended function with a level of confidence commensurate with the projects' airworthiness requirement. The standard is objective driven, and companies may use a variety of means to achieve compliance as long as they meet the objective(s) in question. To comply with DO-178C, companies must provide multiple supporting documents and records surrounding their development processes.

Different airworthiness levels within DO-178C — A, B, C, D and E — directly correspond to the consequences of a potential software failure: catastrophic, hazardous/severe-major, major, minor or no effect, respectively as shown in Table 1. Each software level has a defined number of objectives that need to be satisfied (some with interdependence). These different software level certifications determine the rigor required in testing, with the software verification being the most challenging part of this process.

As you can see, compliance involves six key processes: planning, development, verification, configuration management, quality assurance (QA) and certification liaison. Because the certification liaison process is not a development activity, this paper only focuses on the first five areas.

One of the significant changes in DO-178C from DO-178B is that there are four additional

Levels / Failures condition	**Objectives / With independence**
A / Catastrophic	71 / 33
B / Hazardous	69 / 21
C / Major	62 / 8
D / Minor	26 / 5
E / No safety impact	0 / 0

Table 1: Objectives for each software level

DO-178C overview continued

supplements that may be used in conjunction with the DO-178C. These supplements are used to avoid the need to update or expand the text inside the main DO-178C document. For example, the software tool qualification has been deleted in the main DO-178C and has been replaced with Section DO-330. In addition to DO-330, the other criteria are:

- DO-331 - model-based development and verification
- DO-332 - object-oriented technology and related techniques
- DO-333 - formal methods

Planning and development

As with the other processes involved in proving compliance with DO-178C, planning requires associated output documentation, including the following:

- Plan for Software Aspects of Certification (PSAC)
- Software Development Plan (SDP)
- Software Verification Plan (SVP)
- Software Configuration Management Plan (SCMP)
- Software Quality Assurance Plan (SQAP)
- System Requirements Standard (SRS)
- Software Design Standard (SDS)
- Software Code Standard (SCS)

Output documents associated with meeting DO-178C standards across the development process include software requirements data, software design descriptions, source code and executable object code.

According to DO-178C stipulations, without verifiable, unambiguous, consistent and well-defined requirements, the development team is required to create a problem report and submit the issue back to the requirements input source to be clarified and corrected. The development team must be able to trace system requirements that will be implemented in high level software requirements to one or more low-level software requirements, and a low-level requirement to one or more high-level software requirements.

In addition, the development team needs to provide all of their derived requirements to the system safety assessment process. In a nutshell, this means that all of **the source code developed needs to be traceable, verifiable and consistent, and it needs to correctly fulfill the low-level software requirements**.

DO-178C requires effective processes for measuring and reporting project status deliverables. Leveraging automated measurement and reporting tools can help fulfill the DO-178C standard by:

- Allowing access to data in multiple tools across the development workflow to avoid slow, costly and error-prone manual data collection
- Automatically generating reports and dashboards to help generate consistent evidence of compliance and provide stakeholders with the correct information in a timely manner

Verification

To help ensure that your software fulfills the DO-178C standard, your development team must submit a verification report that shows the absence of errors — not just that they have tested for and detected no errors.

Your development team needs to prove that all lower-level artifacts satisfy higher-level artifacts, that there is traceability between requirements and test cases via requirements-based coverage analysis, and then demonstrate traceability between code structure and test cases through a structural coverage analysis. Each requirement in your software development process must be traceable not only to the code that implements it, but also to the review, test or analysis through which it has been verified. Your development team must also ensure that it can trace implemented functionality back to requirements and that testing can prove this, while eliminating any dead code or code that is not traceable to requirements.

The output documentation associated with DO-178C requires:

- Software verification cases and procedures (SVCP)
- Software verification results (SVR)
- Review of all requirements, design and code
- Testing of executable object code
- Code coverage analysis

Line, decision and condition coverage requirements are determined by the compliance level (A-E) as shown in Table 2.

Levels / Coverage	Coverage requirements
A / MCDC	Level B + 100 percent Modified Condition/Decision Coverage
B / DC	Level C + 100 percent Decision Coverage
C / SC	Level D + 100 percent Statement (or line) Coverage
D	100 percent Requirements Coverage
E	No coverage

Table 2: Coverage requirements by DO-178C levels

Verification continued

As listed, DO-178C defines specific verification objectives, including requirements-based testing, robustness testing and coverage testing, depending on the software level for which you are complying. Each level builds upon the previous level starting with Level E. Each type of coverage is defined in the standard — for example, statement coverage means that every statement in the program has been invoked at least once, while decision coverage means that every point of entry and exit in the program has been invoked at least once and every decision in the program has reached all possible outcomes at least once. The Coverage criteria List references Table 2, highlighting test requirements by compliance level.

– Every point of entry and exit in the program has been invoked at least once.
  - Decision coverage
  - Condition coverage
  - Condition/Decision coverage
  - Modified condition/Decision coverage
  - Multiple condition/Decision coverage

– Every statement in the program has been invoked at least once.
  - Statement coverage

– Every decision in the program has reached all possible outcomes at least once.
  - Decision coverage
  - Condition/Decision coverage
  - Modified condition/Decision coverage
  - Multiple condition/Decision coverage

– Every condition in a decision in the program has reached all possible outcomes at least once.
  - Condition coverage
  - Condition/Decision coverage
  - Modified condition/Decision coverage
  - Multiple condition/Decision coverage

– Every condition in a decision has been shown to independently affect that decision’s outcome.
  - Modified condition/Decision coverage
  - Multiple condition/Decision coverage

– Every combination of condition outcomes within a decision has been invoked at least once.
  - Multiple condition/Decision coverage

List – Coverage criteria: DO-178C stipulates coverage testing requirements by compliance level

Configuration management and quality assurance

To support compliance with DO-178C elements surrounding configuration management, companies are required to do the following:

- Uniquely identify each configuration item
- Protect baselines of configuration items from change
- Trace a configuration item to the configuration item from which it was derived (lineage and history)
- Trace baselines to the baselines from which they were derived
- Reproduce builds (replicate executable object code)
- Provide evidence of change approvals
- Produce output documentation for a software configuration index (SCI) and a software lifecycle environment configuration index (SECI)

Quality assurance (QA)

The QA process in DO-178C requires reviews and audits to demonstrate compliance. Key output documents in this process include software quality assurance records (SQARs), a software conformity review (SCR) and a software accomplishment summary (SAS).

DO-178C also requires that companies implement a problem reporting system to document any change to the formal design baseline.

IBM solutions to support DO-178C standard

As you can see implementing and demonstrating compliance to DO-178C can be a challenge in terms of the rigor, traceability and reporting required. To be competitive your company needs to adopt a solution that can help reduce both the burden and the costs of compliance. The IBM Engineering Lifecycle Management (ELM) solution for systems and software development provides cross-team and cross-lifecycle collaboration, automation and reporting capabilities to help comply with the DO-178C standard.

The IBM ELM platform provides a rich set of capabilities for managing your entire development lifecycle. Which includes managing requirements, test, workflow, as well as modelling and systems design activities. The integrated design of the IBM ELM suite ensures the seamless sharing of information enabling full transparency and traceability across the entire development lifecycle to help you meet the DO-178C standard. Figure 1 illustrates the implied information model that is expected by DO-178C in terms of the necessary engineering artifacts and the respective traceability. As illustrated, the DO-178C standard spans requirements, design, test, and software development – with accompanying traceability. Each one of those artifacts is managed by a dedicated ELM application as described below. Figure 1 also shows how the different artifacts are managed by the different ELM applications: DOORS Next, Rhapsody, ETM, and EWM.

IBM Engineering Requirements Management (DOORS or DOORS Next) manages the requirements artifacts and their respective traceability which is a core DO-178C certification requirement, as illustrated in Figure 1. It also helps maintain requirements configurations and audit trails

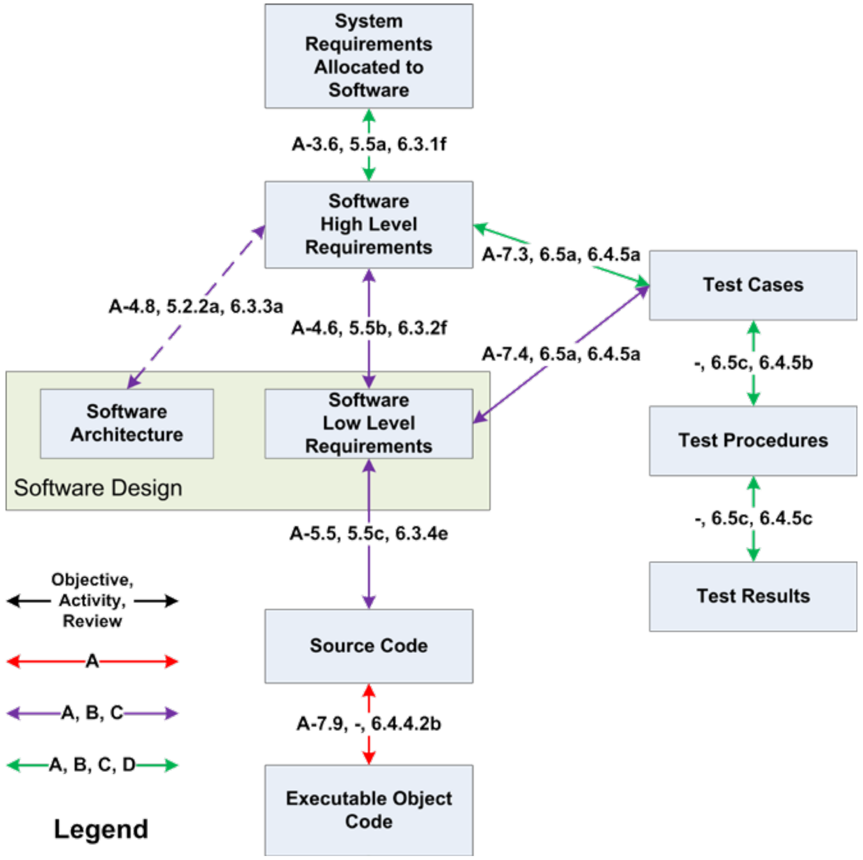


Figure 1: DO-178C stipulates coverage testing requirements by compliance level.

IBM solutions to support DO-178C standard continued

as mentioned in section 04 (Verification) earlier. Overall DOORS/DOORS Next helps development teams reduce costs, increase efficiency and improve quality by enabling teams to optimize requirements communication, collaboration and verification throughout your organization and across your development partners. IBM's requirements management tools share the ELM integrated data foundation across the entire ELM environment for complete data visibility and traceability spanning the development lifecycle (i.e. developers leveraging IBM Engineering Test Management can seamlessly demonstrate requirements-based test coverage).

IBM Engineering Test Management (ETM) manages the test artifacts referenced in Figure 1 test cases, test procedures and test results. ETM provides collaborative and customizable test planning, test specification, execution management, tracking and metrics reporting that provides a central hub through which to manage the verification process. Most important from a DO-178C standpoint is that ETM maintains and automates the traceability between requirements and test cases respectively. This capability helps identify any gaps between requirements and tests, addressing one of the key foundations for assuring airworthiness of software under the DO-178C standard. ETM also maintains all the testing evidence required to be presented to the certification authorities. By providing open interfaces, this solution allows development teams to connect IBM and third-party embedded test execution solution qualified for DO-178C embedded code testing and code coverage measurement.

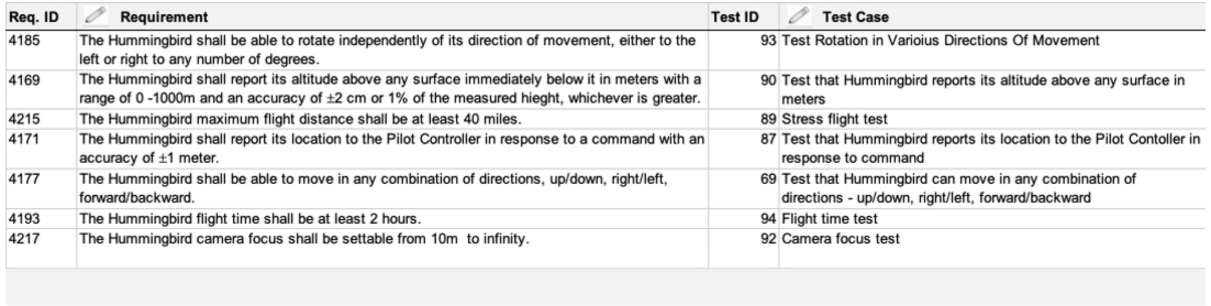
IBM Engineering Systems Design Rhapsody is a collaborative design and design verification environment for systems engineers and software engineers. It captures and manages the software architecture and the low level software requirements (design details) using the UML, and their traceability to the software high level requirements. It also facilitates the traceability between the SW design to the test artifacts in ETM. This traceability is mandated and is quite tedious to maintain if managed in documentation tools such as MS PowerPoint or Visio.

IBM Rhapsody also provides a model base testing (MBT) capability to verify the design models. Another key Rhapsody capability is code generation, where in addition to specifying the software design it leverages UML behavioural definitions such as Statecharts to automatically produce fully functional MISRA compliant C and C++ code that can be used as part of the airborne software.

IBM Engineering Workflow Management fulfills two key functions required by DO-178C. First is source code management and traceability, as illustrated in Figure 1. This also includes source code configuration management which was described earlier in section 4 (Verification) of this document. The second function is tracking the related activity such as performing reviews, and providing the evidence that the software development process presented to the authorities is indeed being followed. In general, EWM helps coordinate distributed teams' activities on a unified change, configuration and release management platform. This helps improve collaboration and increase individual and team productivity by coordinating software development around a single or multiple configuration management repository and help improve time to market by 20%.[3]

Additional IBM Engineering capabilities

The IBM ELM environment also contains reporting and analytics applications to provide visibility for decision making and help automate the creation of the necessary documentation evidence to the certification authorities. One example is the IBM Engineering Lifecycle Optimization – Publishing, which is an automated document generation solution that provides the capability to connect a variety of data sources, across the Engineering Lifecycle Management environment as well as select third-party tools, to produce the various DO-178C documents based on custom templates designed for that purpose – Figure 2 illustrates an automated requirements test coverage traceability report. Documentation automation is a major factor in reducing the large overhead incurred by the certification process.



Req. ID	Requirement	Test ID	Test Case
4185	The Hummingbird shall be able to rotate independently of its direction of movement, either to the left or right to any number of degrees.	93	Test Rotation in Varioius Directions Of Movement
4169	The Hummingbird shall report its altitude above any surface immediately below it in meters with a range of 0 -1000m and an accuracy of ±2 cm or 1% of the measured hieght, whichever is greater.	90	Test that Hummingbird reports its altitude above any surface in meters
4215	The Hummingbird maximum flight distance shall be at least 40 miles.	89	Stress flight test
4171	The Hummingbird shall report its location to the Pilot Controller in response to a command with an accuracy of ±1 meter.	87	Test that Hummingbird reports its location to the Pilot Contoller in response to command
4177	The Hummingbird shall be able to move in any combination of directions, up/down, right/left, forward/backward.	69	Test that Hummingbird can move in any combination of directions - up/down, right/left, forward/backward
4193	The Hummingbird flight time shall be at least 2 hours.	94	Flight time test
4217	The Hummingbird camera focus shall be settable from 10m to infinity.	92	Camera focus test

Figure 2: Automated traceability report.

Why IBM

The IBM ELM platform provides your engineering team with a best practices approach for adopting DO-178C standards into your development processes. This can help offset compliance overhead costs by improving efficiency and lowering rework costs. The IBM ELM solution for systems and software engineering is designed to help engineering teams collaborate and deliver the right products on time, on budget, with the right quality - and accelerate compliance readiness with the DO-178C standard. IBM ELM solutions for safety-critical software development are extensible, through both IBM and third-party offerings, to help address future development requirements.

> Offerings from IBM provide a measured, incremental implementation approach to help you build confidence, minimize risk and demonstrate return on investment.

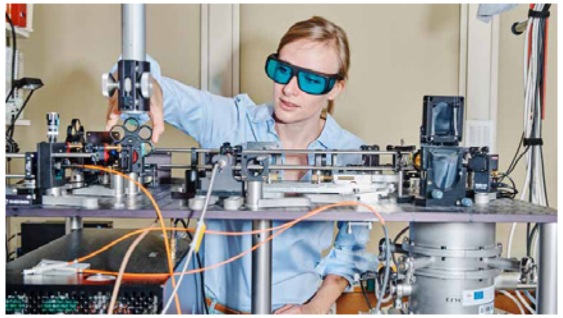
By deploying IBM ELM solutions, engineering teams can reuse software assets and skills to improve development productivity and accelerate time to market and innovation. Comprehensive traceability functionality allows development teams to enhance collaboration and communication and enables teams from multiple disciplines to coordinate system and software engineering activities. IBM's standards-based solution provides an open and extensible management platform across the development lifecycle — from requirements to deployment. Leveraging IBM ELM solutions, companies can better manage the collaboration across global development and delivery teams, be they internal, suppliers, agencies or contractors to more efficiently meet the DO-178C standard.

Learn more

Improve the quality of your work by building in knowledge driven requirements.

→

Next steps



Explore IBM ELM solutions

Read this whitepaper to learn more about IBM ELM solutions.

Download now

Forrester Opportunity Snapshot on ELM

Read this Forrester report, that surveyed 150 software development decision makers, to know how ELM enables predictability and innovation.

Read blog

Take the ELM product tour

See how you can scale, improve data transparency, automate processes and achieve compliance.

Read blog

Sources

1. Digital: The next horizon for global aerospace and defense, McKinsey & Company, May 2021

2. Citation: www.do178site.com/do178b_questions.php, do178site.com, 2008

3. Moving to knowledge driven requirements management, IBM

IBM Corporation
Route 100
Somers, NY 10589

Produced in the United States of America
January 2022

This document is current as of the initial date of publication and may be changed by IBM at any time. Not all offerings are available in every country in which IBM operates.

The information in this document is provided “as is” without any warranty, express or implied, including without any warranties of merchantability, fitness for a particular purpose and any warranty or condition of non-infringement.

IBM products are warranted according to the terms and conditions of the agreements under which they are provided.