The Impact of RTCA DO-178C on Software Development

By following DO-178C, organizations can implement aeronautical software with clear and consistent ties to existing systems and safety processes and address emerging trends and technologies across the industry.

Executive Summary

A new guideline has emerged to help regulate the development and certification of software and the delivery of multiple supporting documents and records used on aircraft or engines. The previous guideline — called RTCA DO-178B, Software Considerations in Airborne Systems and Equipment Certification, and produced by the Radio Technical Commission for Aeronautics Inc. — served as a de facto standard for avionics equipment and software development worldwide.

With the release of RTCA DO-178C — the new development guidance for certifiable aviation software — executives and product managers for manufacturers of airborne systems are examining the short- and long-term impacts on the cost, scheduling and risk of their certifiable product development approaches. Although the changes within DO-178C proper are relatively few, manufacturers can expect critical wide-ranging implications. More significant are the four detailed supplements intended to address 20 years of progress in technology and process since the last major revisions of the development guidelines.

The new guidance represents a significant change in the Federal Aviation Association's posture toward regulated software development. The DO-178C guidelines tighten some previously established controls, while also establishing concrete guidance for greater flexibility in development approaches. This flexibility, however, must be carefully examined in terms of the potential costs and benefits, to establish the most efficient certifiable product development approach.

This white paper discusses these shifts from a technical perspective and provides management visibility into the emerging challenges and opportunities associated with the updated guidance. Lastly, this paper also examines the relationship between DO-178C and the supplements: DO-330 (tool qualification), DO-331 (model-based development and verification), DO-332 (object-oriented technology and related techniques) and DO-333 (formal methods).

RTCA Guideline Progression

RTCA DO-178A was last revised in 1992, which begot DO-178B. DO-178C is the latest revision to the DO-178B guidelines released in January 2012,

Cognizant

which describe objectives for software lifecycle processes, activities and design considerations for achieving the objectives and proving that the objectives have been satisfied.

The majority of DO-178B is dedicated to describing the sequential development methodology for new, custom-built avionics software. This approach is a requirements-based development and verification methodology that includes a number of alternative methods for satisfying these objectives. DO-178B is not a strict or detailed standard; it is a general software development framework for developing provable, high-reliability software, consistently. Developers of avionics equipment and software must comply with the guidance provided by DO-178B.

Comparing DO-178B and DO-178C

The new guidelines include both minor and significant changes, all of which will significantly impact the way certifiable software development is managed.

Minor Changes

The minor changes include the removal of known errors and inconsistencies, as well as the addition of consistent terminology throughout the document. Wording improvements can be seen throughout the guidelines, as well.

Coordinated systems/software aspects are evident in the document, providing additional rationale for the overall software development objectives and their justifications.

- **Errors and inconsistencies:** DO-178C has addressed the errata of DO-178B and has removed the inconsistencies among the tables of DO-178B Annex A. To remove an inconsistency regarding software standards for Level D software, DO-178B objective A-9#1, plans and standards were split into two DO-178C objectives, specifically:
  - Assurance is obtained that software development and integral processes comply with approved software plans (Table A-9#2).
  - Assurance is obtained that software development and integral processes comply with approved software standards (Table A-9#3).
- **Consistent terminology:** DO-178C has addressed DO-178B's issues with the use of specific terms, such as "guidance," "guidelines," "purpose," "goal," "objective" and "activity." This was accomplished by expanding the glossary and changing the text accordingly so that the use of those specific terms was consistent throughout the document.
- **Wording improvements:** DO-178C has improved wording throughout the document, with the objective of making the document more precise, while maintaining the original intent of DO-178B.
- **Objectives and activities:** DO-178C reinforced the point that, in order to fully understand the recommendations, the full body of the document should be considered. For example, Annex A now includes references to each activity, as well as to each objective. Moreover, Section 1.4, now titled "How to Use This Document," reinforces the point that activities are a major part of the overall guidance.
- **Coordinated system/software aspects:** Section 2 of DO-178B was updated with software development principles to reflect current system practices. The updates were made based upon coordination with other avionics standards organizations that were updating their system-level guidance at the same time that SC-205/WG-71 (EUROCAE) was updating the DO-178B's software-level guidance.
- **DO-178B hidden objectives:** DO-178C has added so-called "hidden objectives" to Annex A, including:
  - A means for detecting the object code that is not directly traceable to the source code and to ensure its verification coverage is defined (Table A-1 #4).
  - Assurance that software plans and standards are developed and reviewed for consistency (Table A-9#1).
  - An explicit objective to ensure that object code is directly traceable to source code "Source to Object Traceability" (Table A-7#9).
- **DO-178B topic omissions:** DO-178C has addressed a few general topics that resulted in changes to several sections of the document, such as oversight of suppliers, parameter data items and traceability. In addressing these topics, two additional objectives were added to Annex A:
  - Parameter Data Item File (PDIF) to be loaded complies with low-level requirements (Table A-6#6).

- Verification coverage of PDIF elements (Table A-7#9).
- Added trace data, as required.
- Lifecycle Data to be provided and verified (Section 11.21).

- **DO-178B gaps and clarifications:** DO-178C addressed several specific issues that resulted in changes to only one or two paragraphs. Each such change may have an impact on the applicant, as they either addressed clear gaps in DO-178B or clarified guidance that was subject to differing interpretations.

Examples of gaps addressed include:

- Changes to the "Modified Condition/Decision Coverage" definition. Masking MC/DC and Short Circuit, as well as DO-178B's Unique Cause MC/DC, are now allowed (Glossary).
- An addition to Level A, stating that "if a complier, linker or other means generates additional code that is not directly traceable to source code statements, then additional verification should be performed to establish the correctness of such generated code sequences" (6.4.4.2.b).
- The need for derived requirements to be provided to the system processes, including the system safety assessment process (rather than just provided to the system safety assessment process) (5.1.1b, 5.2.1b).

Examples of clarifications include:

- The structural coverage analysis of data and control coupling between code components should be achieved by assessing the requirements-based tests (6.4.4.2.c).
- All tests added to achieve structural coverage are based on requirements (6.4.4.2.d).
- All the code that may be classified as deactivated code (6.4.4.3.d).

Significant Changes

The significant changes include the addition of technology supplements to keep the core of DO-178B intact for the future. New tool qualification guidance helps ensure separation of airborne software from tools that may not be airborne.

- **Technology supplements:** DO-178C recognizes that new software development methodologies may result in new issues. Rather than expanding the text to account for all the current software development methodologies (and being revised yet again to account for future software development methodologies), DO-178C acknowledges that one or more technology supplements may be used in conjunction with DO-178C to modify the guidance for specific technologies or methodologies. Section 12's addressing of tool qualification and alternative methods was heavily impacted, since planned technology supplements more completely address such technologies. The technology supplements include the following:
  - Model Based Development and Verification Supplement to DO-178C and DO-278A (DO-331.)
  - Object-Oriented Technology and Related Techniques Supplement to DO-178C and DO-278A (DO-332).
  - Formal Methods Supplement to DO-178C and DO-278A (DO-333).
  - Software Tool Qualification Considerations (DO-330).

  These new supplements provide guidance and objectives for both DO-178C and DO-278A. Rather than expanding the text in the body of DO-178B, each supplement describes how the objectives of DO-178C are revised for specific techniques, including:
  - Technology-specific interpretation.
  - Modification of objectives.
  - Additional objectives.

  Each supplement provides technology-specific supporting information to provide clarification on the use of technology. Each supplement defines the scope of the supplement and the objectives it contains.

  Objectives tables in the supplements follow the same structure as the objectives table in DO-178C, namely:
  - References to objective definitions.
  - References to activity definitions.
  - Identification of the applicability by DAL.
  - Identification of the output, documenting compliance.
  - Reference to the output definition.
  - Identification of the output configuration control category.

- **Model-Based Development and Verification Supplement to DO-178C and DO-278A:** This supplement contains modifications and additions to DO-178C and DO-278A objectives, activities, explanatory text and software lifecycle data that should be addressed when model-based development and verification are used as part of the software lifecycle. This includes the artifacts that would be expressed using models, as well as the verification evidence that could be derived from them. Therefore, this supplement also applies to the models developed in the system process that define software requirements or software architecture.

  A model is an abstract representation of a set of software aspects of a system that is used to support the software development process or the software verification process. This supplement addresses model(s) that have the following characteristics:

  - The model is completely described using an explicitly identified modeling notation. The modeling notation may be graphical and/or textual.
  - The model contains software requirements and/or software architecture definition.
  - The model is of a form and type that is used for direct analysis or behavioral evaluation as supported by the software development process or the software verification process.

- **Object-Oriented Technology and Related Techniques Supplement to DO-178C and DO-278A:** This supplement identifies the additions, modifications and deletions to DO-178C and DO-278A objectives when object-oriented technology or related techniques are used as part of the software development lifecycle and additional guidance is required. This supplement, in conjunction with DO-178C, is intended to provide a common framework for the evaluation and acceptance of object-oriented technology and related techniques-based systems.

  Object-oriented technology has been widely adopted in non-critical software development projects. The use of this technology for critical software applications in avionics has increased, but there are a number of issues that need to be considered to ensure safety and integrity goals are met. These issues are both directly related to language features and to complications encountered with meeting well-established safety objectives.

- **Formal Methods Supplement to DO-178C and DO-278A:** This supplement identifies the additions, modifications and substitutions to DO-178C and DO-278A objectives when formal methods are used as part of a software lifecycle and the additional guidance required. It discusses those aspects of air-worthiness certification that pertain to the production of software, using formal methods for systems approved using DO-178C.

  Formal methods are mathematically-based techniques for the specification, development and verification of software aspects of digital systems. The mathematical basis of formal methods consists of formal logic, discrete mathematics and computer-readable languages. The use of formal methods is motivated by the expectation that, as in other engineering disciplines, performing appropriate mathematical analyses can contribute to establishing the correctness and robustness of a design.

- **Software Tool Qualification Considerations (DO-178B Section 12.2):** The terms "development tool" and "verification tool" are replaced by three qualification criteria that determine the applicable tool qualification level (TQL) in regard to the software level. The guidance to qualify a tool is removed in DO-178C, but it is provided in a domain-independent, external document referenced in Section 12.2.

  The tool criteria are as follows:

  - **Criteria #1:** A tool whose output is part of the airborne software and thus could insert an error.
  - **Criteria #2:** A tool that automates verification processes and thus could fail to detect an error and whose output is used to justify the elimination or reduction of:
    - Verification processes other than automated by the tool.
    - Development processes that could have an impact on the airborne software.
  - **Criteria #3:** A tool that, within the scope of its intended use, could fail to detect an error.

The tool qualification criteria and qualification levels are shown in Figure 1.

Design Assurance Level (DAL)	Criteria		
	Criteria 1	Criteria 2	Criteria 3
Level A	TQL-1	TQL-4	TQL-5
Level B	TQL-2	TQL-4	TQL-5
Level C	TQL-3	TQL-5	TQL-5
Level D	TQL-4	TQL-5	TQL-5

Figure 1

The objectives of DO-178B and DO-178C are summarized in Figure 2.

Design Assurance Level(DAL)	DO-178B	DO-178C
Level A	66 Objectives	71 Objectives
Level B	65 Objectives	69 Objectives
Level C	57 Objectives	62 Objectives
Level D	28 Objectives	26 Objectives
Level E	No Objectives	No Objectives

Figure 2

- **New Lifecycle Data:** There are now requests for new data items to be made available, as well expansion of content for some existing data items. For example, the PSAC must address the supplier oversight and describe the means of ensuring that supplier processes and outputs will comply with approved software plans and standards.
  - Section 11 adds two new lifecycle data items.
    - **PDIF (Section 11.22):** To support new objectives, and it has the control category 1 for all DAL's.
    - **Trace Data (Section 11.21):** Implied for DO-178B, it is now clarified that it has to be bi-directional and control category on DAL.
- **Trace Data:** DO-178C has made an explicit data item related to traceability. It also required bi-directional traceability. Trace data has to demonstrate associations between:
  - Systems to high-level requirements
    - High-level to low-level requirements, dependent on level
    - Low-level requirements to source code, dependent on level
  - Requirements to test cases
    - High-level and/or low-level, dependent on level
  - Test cases to test procedures
  - Test procedures to test results

Conclusion

DO-178C has decreased the level of subjectivity for many activities in the software development lifecycle objectives. Yet even today, the definition of the processes and plans for execution of the process are key to successful compliance.

Without question, the new DO-178C guidance adopted by the avionics industry will impact established certifiable software development processes. Manufacturers will need to modify their existing practices to accommodate the revised guidance. Although the benefits from the adoption of new technologies and tools will be significant, expectations – especially in the short term – must be tempered by a clear vision of the challenges associated with migration and early adoption.

Manufacturers must carefully consider both short- and long-term costs and benefits. A comprehensive and detailed gap analysis – together with a well-considered certifiable development process roadmap – will be necessary for a certifiable product manufacturer to prepare for an effective treatment of the revised guidance. Avoiding costly rework, while effectively leveraging new technologies, is the key to remaining competitive. The time to plan is now; the costs of delay can be dramatic.

References

- RTCA DO-178B – Software Considerations in Airborne Systems and Equipment Certification
- RTCA DO-178C – Software Considerations in Airborne Systems and Equipment Certification
- DO-330 – Software Tool Qualification Considerations
- DO-331 – Model Based Development and Verification Supplement to DO-178C and DO-278A
- DO-332 – Object-Oriented Technology and Related Techniques Supplement to DO-178C and DO-278A
- DO-333 – Formal Methods Supplement to DO-178C and DO-278A

About the Author

Maddireddy Sudheer Reddy is a Senior Manager of System Engineering within Cognizant's Engineering Manufacturing Solutions Business Unit. He has more than 13 years of well-rounded engineering and management experience in safety critical software development within the avionics and railway domains. Sudheer has led and executed multiple projects/programs, with responsibilities focused on meeting program requirements, managing customer expectations and developing long-term relationships built on trust. His areas of expertise include systems engineering, software engineering and program management. Sudheer holds a master's degree in software systems and computer science, as well as an engineering degree. He can be reached at Sudheer.Maddireddy@cognizant.com.

About Cognizant

Cognizant (NASDAQ: CTSH) is a leading provider of information technology, consulting, and business process outsourcing services, dedicated to helping the world's leading companies build stronger businesses. Headquartered in Teaneck, New Jersey (U.S.), Cognizant combines a passion for client satisfaction, technology innovation, deep industry and business process expertise, and a global, collaborative workforce that embodies the future of work. With over 50 delivery centers worldwide and approximately 145,200 employees as of June 30, 2012, Cognizant is a member of the NASDAQ-100, the S&P 500, the Forbes Global 2000, and the Fortune 500 and is ranked among the top performing and fastest growing companies in the world. Visit us online at www.cognizant.com or follow us on Twitter: Cognizant.

World Headquarters
500 Frank W. Burr Blvd.
Teaneck, NJ 07666 USA
Phone: +1 201 801 0233
Fax: +1 201 801 0243
Toll Free: +1 888 937 3277
Email: inquiry@cognizant.com

European Headquarters
1 Kingdom Street
Paddington Central
London W2 6BD
Phone: +44 (0) 20 7297 7600
Fax: +44 (0) 20 7121 0102
Email: infouk@cognizant.com

India Operations Headquarters
#5/535, Old Mahabalipuram Road
Okkiyam Pettai, Thoraipakkam
Chennai, 600 096 India
Phone: +91 (0) 44 4209 6000
Fax: +91 (0) 44 4209 6060
Email: inquiryindia@cognizant.com