CHANGE

U.S. DEPARTMENT OF TRANSPORTATION
FEDERAL AVIATION ADMINISTRATION

ORDER 8110.49 CHG 2

National Policy

Effective Date: 4/10/2017

SUBJ: Software Approval Guidelines

1. Purpose. This change transmits revised pages to Order 8110.49 Chg 1, *Software Approval Guidelines.* This change is issued to:

a. Delete part of chapter 2 to allow flexibility in conducting software reviews.

b. Delete chapter 3 to allow alignment with risk-based directives in Order 8040.4A.

c. Replace Appendix 1 with Appendix A with worksheets taken from the deleted chapter 3.

d. Delete Appendices 2 – 4 which were examples related to chapter 3.

e. Update document references only in Chapters 1 and 2 to reflect current versions, where applicable.

2. Who this change affects. Managers and staff of the Federal Aviation Administration (FAA) Aircraft Certification Service (AIR), including any persons designated by the Administrator, and organizations associated with the aircraft certification process required by Title 14 of the Code of Federal Regulations (14 CFR).

3. Disposition of Transmittal Paragraph. Retain this transmittal sheet until the directive is cancelled by a new directive.

PAGE CHANGE CONTROL CHART

Remove Pages	Dated	Insert Pages	Dated
i	9/28/11	i	
iv	9/28/11	iv	
Chapter 1 (1 thru 8)	9/28/11	Chapter 1 (1 thru 7)	
Chapter 2 (9 thru 10)	6/3/03	Chapter 2 (8 thru 10)	
Chapter 2 (11 thru 20)	6/3/03	Deleted	

PAGE CHANGE CONTROL CHART (CONTINUED)

Remove Pages	Dated	Insert Pages	Dated
Chapter 3 (21 thru 22)	6/3/03	Deleted	
Chapter 3 (23 thru 27)	9/28/11	Deleted	
Chapter 3 (28)	6/3/03	Deleted	
Appendix 1 (A1-1)	9/28/11	Appendix A (A-1 thru A-4)	
Appendix 2 (A2-1)	9/28/11	Deleted	
Appendix 3 (A3-1)	9/28/11	Deleted	
Appendix 3 (A3-2)	9/28/11	Deleted	
Appendix 4 (A4-1)	9/28/11	Deleted	
Appendix 5 (A5-1)	9/28/11	Appendix B (B-1)	

Susan J. M. Cabler
Acting Manager, Design, Manufacturing, &
Airworthiness Division
Aircraft Certification Service

Table of Contents

Chapter 1. Introduction

1-1. Purpose. This order guides Aircraft Certification Service (AIR) field offices and designees on how to apply RTCA/DO-178B and RTCA/DO-178C, herein called RTCA/DO-178B/C for approving software used in airborne computers. Both are titled "Software Considerations in Airborne Systems and Equipment Certification". The guidelines are applicable to the approval of airborne systems and equipment and the software aspects of those systems related to type certificates (TC), supplemental type certificates (STC), amended type certificates (ATC), amended supplemental type certificates (ASTC), and technical standard order (TSO) authorizations (TSOA).

1-2. Distribution. Distribute this order to the branch level in Washington headquarters Aircraft Certification Service, section level in all aircraft certification directorates, all chief scientific and technical advisors (CSTA), all aircraft certification offices (ACO), all manufacturing inspection offices (MIO), all manufacturing inspection district or satellite offices (MIDO/MISO), and all flight standards district offices (FSDO). Make additional limited distribution to organization designation authorization administrators, designated engineering representatives (DER), flight standards air carrier district offices, the aeronautical quality assurance field offices, and the FAA Academy.

1-3. Related Publications. The latest amendments of the following publications are the primary reference materials for this order:

a. Code of Federal Regulations. 14 CFR part 21, *Certification Procedures for Products and Parts*.

b. FAA ACs and Orders. Copies of the following ACs and orders are available from the FAA website at http://www.faa.gov/regulations_policies.

(1) AC 20-115, *Airborne Software Assurance.*

(2) AC 20-153, *Acceptance of Aeronautical Data Processes and Associated Databases.*

(3) AC 21-43, *Production Under 14 CFR Part 21, Subparts F, G, K, and O.*

(4) AC 23.1309-1, *System Safety Analysis and Assessment for Part 23 Airplanes.*

(5) AC 25.1309-1, *System Design and Analysis.*

(6) AC 27-1, *Certification of Normal Category Rotorcraft.*

(7) AC 29-2, *Certification of Transport Category Rotorcraft.*

(8) AC 33.28-1, *Compliance Criteria for 14 CFR § 33.28, Aircraft Engines, Electrical and Electronic Engine Control Systems.*

(9) AC 33.28-2, *Guidance Material for 14 CFR 33.28, Reciprocating Engine, Electrical and Electronic Engine Control Systems.*

(10) AC 33.28-3, *Guidance Material for 14 CFR 33.28, Engine Control Systems.*

(11) AC 120-64, *Operational Use & Modification of Electronic Checklists*.

(12) AC 120-76, *Guidelines for the Certification, Airworthiness, and Operational Use of Electronic Flight Bag.*

(13) Order 8040.4, *Safety Risk Management Policy.*

(14) Order 8110.4, *Type Certification Process.*

(15) Order 8110.42, *Parts Manufacturer Approval Procedures.*

(16) Order 8110.55, *How to Evaluate and Accept Process for Aeronautical Database Suppliers*.

c. Reserved

d. RTCA, Inc. Documents. Copies of RTCA documents may be purchased from RTCA, Inc., 1150 18th St. NW, Suite 910, Washington, D.C. 20036. Alternatively, copies may be purchased on-line at http://www.rtca.org. RTCA documents referenced in this order are:

(1) RTCA, Inc., document RTCA/DO-178B, *Software Considerations in Airborne Systems and Equipment Certification*, dated December 1, 1992.

(2) RTCA, Inc., document RTCA/DO-178C, *Software Considerations in Airborne Systems and Equipment Certification*, dated December 13, 2011.

(3) RTCA, Inc., document RTCA/DO-200A, *Standards for Processing Aeronautical Data*, dated September 28, 1998.

(4) RTCA, Inc., document RTCA/DO-200B, *Standards for Processing Aeronautical Data*, dated June 18, 2015.

(5) RTCA, Inc., document RTCA/DO-248B, *Final Report for Clarification of DO-178B Software Considerations in Airborne Systems and Equipment Certification*, dated October 12, 2001.

(6) RTCA, Inc., document RTCA/DO-248C, *Supporting Information for DO-178C and DO-278A*, dated December 13, 2011.

(7) RTCA DO-330, *Software Tool Qualification Considerations*, dated 13 December 2011.

(8) RTCA DO-331, *Model-Based Development and Verification Supplement to DO-178C and DO-278A*, dated December 13, 2011.

(9) RTCA DO-332, *Object-Oriented Technology and Related Techniques Supplement to DO-178C and DO-278A*, dated December 13, 2011.

(10) RTCA DO-333, *Formal Methods Supplement to DO-178C and DO-278A*, dated December 13, 2011.

e. SAE Documents. Copies of SAE documents may be purchased from SAE International, 400 Commonwealth Drive, Warrendale, PA 15096-0001. Alternatively, copies may be purchased on-line at www.sae.org. SAE documents referenced in this order are:

(1) Aerospace Recommended Practice ARP4754a, *Development of Civil Aircraft and Systems*.

(2) Aerospace Recommended Practice ARP4761, *Guidelines and Methods for Conducting the Safety Assessment Process on Civil Airborne Systems and Equipment.*

1-4. Cancellation. This order cancels and supersedes the following Order:

a. Order 8110.49 Chg 1, *Software Approval Guidelines*, dated September 28, 2011.

1-5. Background. Between 1998 - 2010, the FAA produced a number of software-related notices to provide guidelines for FAA Aviation Safety Engineers (ASE), Aviation Safety Inspectors (ASI), and designees in various areas of software approval. This order includes those notices and implements improvements to the policy based on lessons learned.

1-6. Software Topics Covered In This Order.

a. On July 19, 2013, the FAA issued AC 20-115C which recognizes RTCA/DO-178C and supplements DO-330, DO-331, DO-332, DO-333 as a means of demonstrating compliance to regulations for the software aspects of airborne systems and equipment certification. (Hereafter, references to use of DO-178C in this order include use of supplements and DO-330, as applicable.) AC 20-115C also provides guidance when making modifications to software approved to previous versions of RTCA/DO-178C. This order assumes that RTCA/DO-178B/C is the means of compliance proposed by the applicant for software approval (except for chapters 8 and 10, where previously developed software and legacy systems are addressed). If the

applicant proposes other means, additional policy and FAA guidance may be needed on a project-by-project basis.

b. This order addresses a variety of software-related topics and is supplemental to RTCA/DO-178B/C. Guidelines in the following areas are addressed:

(1) The software review process (chapter 2),

(2) Reserved

(3) Software conformity inspections (chapter 4),

(4) Field-loadable software (chapters 5 and 6),

(5) User-modifiable software (chapter 7),

(6) Level D previously developed software (chapter 8),

(7) Software tool qualification (chapter 9),

(8) Software changes in legacy systems (chapter 10),

(9) Software change impact analysis (chapter 11),

(10) Reuse of software life cycle data (chapter 12),

(11) Properly overseeing suppliers (chapter 13),

(12) Software problem reporting (chapter 14),

(13) Assuring airborne system databases and aeronautical databases (chapter 15), and

(14) Managing the software development or verification environment (chapter 16).

1-7. Definitions. For purposes of this order, the following definitions apply:

a. Certification authority is the aviation authority that accepts and/or approves software life cycle data.

b. Certification credit is the acceptance by the certification authority that a software process, software product, or demonstration satisfies a certification requirement (see RTCA/DO-178B/C, Glossary).

c. Chief Scientific and Technical Advisor (CSTA) is an executive-level technical expert in the FAA. Previously, a CSTA was referred to as a "National Resource Specialist" (NRS).

d. Configuration item is (1) one or more software components treated as a unit for software configuration management purposes, or (2) software life cycle data treated as a unit for software configuration management purposes (see RTCA/DO-178B/C, Glossary).

e. Field-loadable software (FLS) is software that can be loaded without removal of the equipment from the installation. FLS can refer to either executable code or data (see RTCA/DO-178B, Section 2.5 and RTCA/DO-178C, Section 2.5.5). FLS might also include software loaded into a line replaceable unit at a repair station or shop.

f. Finding is the identification of a failure to show satisfaction to one or more of the RTCA/DO-178B/C objectives.

g. Deleted

h. Option-selectable software is software that contains approved and validated components and combinations of components that may be activated by the user, either through selection by the flight crew or activation by ground personnel (see RTCA/DO-178B, Section 2.4; and RTCA/DO-178C, Section 2.5.4).

i. Original certification project is the first use of the software life cycle data in a completed certification project.

j. Reuse is the subsequent use of unaffected, previously approved software life cycle data.

k. Review is the act of inspecting or examining software life cycle data, software project progress and records, and other evidence to assess compliance with RTCA/DO-178B/C objectives. Review is an encompassing term and may consist of a combination of reading documents, interviewing project personnel, witnessing activities, sampling data, and participating in briefings. A review may be conducted at your own desk, at an applicant's facility, or at an applicant's supplier's facility.

l. Sampling is selecting a representative set of software life cycle data for inspection or analysis. The purpose is to determine the compliance of all software life cycle data developed up to that point in time in the project. Sampling is the primary means of assessing the compliance of the software processes and data. Examples of sampling may include the following:

(1) Inspecting the traceability from system requirements to software requirements to software design to source code to object code to test cases and procedures to test results.

(2) Reviewing analyses used to determine system safety classification, software level, or RTCA/DO-178B/C objective compliance (for example, timing analysis).

(3) Examining the structural coverage of source code modules.

(4) Examining software quality assurance (SQA) records and configuration management records.

m. Software is computer programs and, possibly, associated documentation and data pertaining to the operation of a computer system (see RTCA/DO-178B/C, Glossary).

n. Software Configuration Index (SCI) identifies the configuration of the software product. It can contain one configuration item or a set of configuration items (see RTCA/DO-178B/C, Section 11.16).

o. Software library is a controlled repository of software and related data and documents designed to aid in software development, use, or modification (see RTCA/DO-178B/C, Glossary).

p. Software life cycle data are data produced during the software life cycle to plan, direct, explain, define, record, or provide evidence of activities (see RTCA/DO-178B/C, Section 11.0). Sections 11.1 through 11.20 of RTCA/DO-178B and Sections 11.1 through 11.22 of RTCA/DO-178C describe different kinds of software life cycle data.

q. Software Life Cycle Environment Configuration Index identifies the configuration of the software life cycle environment. It is written to aid reproduction of the hardware and software life cycle environment (see RTCA/DO-178B/C, Section 11.15).

r. Software plans and standards are a set of data that directs the software development processes and integral processes (see RTCA/DO-178B/C, Sections 4.0 and 11.1 through 11.8).

s. Software tool is a computer program used to help develop, test, analyze, produce, or modify another program or its documentation (see RTCA/DO-178B/C, Glossary).

t. Subsequent certification project is the follow-on project in which software life cycle data from the original certification project is reused.

u. Test for certification credit is system certification test conducted under a FAA-approved test plan for the purpose of showing compliance to the regulations.

v. Tool qualification is the process necessary to obtain certification credit for a software tool within the context of a specific airborne system (see RTCA/DO-178B/C, Section 12.2 and Glossary).

w. User-modifiable software (UMS) is software intended for modification by the aircraft operator without review by the certification authority, the airframe manufacturer, or the equipment vendor. Modifications by the user may include modifications to data, modifications to executable code, or both (see RTCA/DO-178B, Section 2.4; and RTCA/DO-178C, Section 2.5.2).

1-8. Acronyms. The following is a list of acronyms used in this order:

AC	Advisory Circular
ACO	Aircraft Certification Office
AIR	Aircraft Certification Service
ASE	Aviation Safety Engineer
ASI	Aviation Safety Inspector
ASTC	Amended Supplemental Type Certificate
ATC	Amended Type Certificate
CFR	Code of Federal Regulations
CMR	Certification Maintenance Requirements
CRC	Cyclic Redundancy Check
CSTA	Chief Scientific and Technical Advisor
DER	Designated Engineering Representative
FAA	Federal Aviation Administration
FLS	Field-Loadable Software
MEL	Minimum Equipment List
MIDO	Manufacturing Inspection District Office
MISO	Manufacturing Inspection Satellite Office
PDS	Previously Developed Software
PMA	Parts Manufacturer Approval
PSAC	Plan for Software Aspects of Certification
SAS	Software Accomplishment Summary
SCI	Software Configuration Index
SCMP	Software Configuration Management Plan
SQA	Software Quality Assurance
STC	Supplemental Type Certificate
TC	Type Certificate
TIA	Type Inspection Authorization
TSO	Technical Standard Order
TSOA	Technical Standard Order Authorization
TSR	Total Score Result
UMS	User-Modifiable Software

1-9. Records Management. Refer to Orders 0000.1g and 1350.14b or your office Records Management Officer (RMO)/Directives Management Officer (DMO) for guidance regarding retention or disposition of records.

1-10. Suggestions for Improvement. If you find deficiencies, a need for clarification, or want to suggest improvements on this order, send a copy of FAA Form 1320-19, Directive Feedback Information, to the Aircraft Certification Service, Attention: Directives Management Officer at 9-AWA-AVS-AIR-DMO@faa.gov, for consideration. If you urgently need an interpretation, you may contact the Design, Manufacturing, and Airworthiness Division, for guidance. You should also use the FAA Form 1320-19 as a follow-up to verbal conversation. FAA Form 1320-19 may be found in Appendix B and electronically at https://employees.faa.gov/tools_resources/forms/.

Chapter 2. Software Review Process

2-1. General.

a. Section 9 of RTCA/DO-178B/C describes the certification liaison process. This process is the vehicle to establish communication and understanding between the applicant and the certification authority. Sections 9.2 and 10.3 of RTCA/DO-178B/C state that the certification authority may review the software life cycle processes and data to assess compliance to RTCA/DO-178B/C. This chapter does not change the intent of RTCA/DO-178B/C.

b. Although desk reviews may be used to successfully review software, on-site reviews have the advantages of access to software personnel, to all automation, and to test setup. Both on-site and desk reviews may be delegated to properly authorized designees. For on-site reviews, the certification authority should include the following practical arrangements with the software developer:

(1) Agreement on the scope of review(s) that will be conducted.

(2) Agreement on date(s) and location(s) of the review(s).

(3) Identification of the certification authority's personnel involved.

(4) Identification of any designees involved.

(5) Development of the agenda(s) and expectations.

(6) Listing of software data to be made available (both before and at the review(s)).

(7) Clarification of procedures to be used.

(8) Identification of any required resources.

(9) Specification of date(s) and means for communicating review results (may include corrective actions and other post-review activities).

2.2 Objectives of the Software Review Process.

a. The certification authority may review the software life cycle processes and associated data to obtain assurance that a software product submitted as part of a certification application complies with the certification basis and satisfies the applicable objectives of RTCA/DO-178B/C. The software review process assists both the certification authority and the applicant to determine if a particular project will meet the certification basis, applicable guidance, and RTCA/DO-178B/C objectives by providing:

(1) Timely technical interpretation of the certification basis, RTCA/DO-178B/C objectives, FAA guidance, issue papers, and other applicable certification requirements.

(2) Visibility into the implementation compliance and the applicable data.

(3) Objective evidence that the software project adheres to its approved software plans and procedures.

(4) The opportunity for the certification authority to monitor designee activities.

b. The level of certification authority involvement in a software project should be determined and documented as soon as possible in the project life cycle. Appendix A provides examples that may be used to determine the level of involvement. The scope and number of software reviews, if any, will depend on several factors including:

(1) Software level(s), as determined by a system safety assessment.

(2) Product attributes (such as size, complexity, system functionality or novelty, and software design).

(3) Use of new technologies or unusual design features.

(4) Proposals for novel software methods or life cycle model(s).

(5) Applicant's experience in satisfying the objectives of RTCA/DO-178B/C.

(6) Availability, experience, and authorization of designees.

(7) Issues associated with Section 12 of RTCA/DO-178B/C in the project.

(8) Applicability of issue papers for software-specific aspects of the certification project.

Chapter 3. Reserved

Chapter 4. Software Conformity Inspection

4-1. General. This chapter describes the software conformity inspection process. This process applies to TC, STC, ATC, ASTC, and TSO authorization projects. This chapter is based on FAA Order 8110.4B and RTCA/DO-178B. While RTCA/DO-178B is recognized by AC 20-115B as a means, but not the only means, to secure FAA approval of the digital computer software, it is used here because it is the typical means of compliance used by applicants integrating airborne software. If another means of compliance other than RTCA/DO-178B is used, the conformity concepts of this chapter should still apply.

4-2. Discussion. A conformity inspection is required to determine that the applicant complies with 14 CFR § 21.33(b) and that the product and components conform to approved type design. For software, type design consists of, as a minimum, Software Requirements Data, Design Description, Source Code, Executable Object Code, Software Configuration Index, and the Software Accomplishment Summary (see RTCA/DO-178B, Section 9.4). Determination of an applicant's compliance to software type design is largely assessed through ASE or DER (if authorized) reviews throughout the software development life cycle; the details of which are presented in chapter 2 of this order. However, there are instances where the state of the software must be reviewed and documented before issuance of TC, STC, ATC, ASTC, or TSO authorization (specifically, test acceptance and installation). Accordingly, there are two means for achieving this: (1) software part conformity inspection, and (2) software installation conformity inspection.

4-3. Software Part Conformity Inspection. The conformity of the test article, test setup, test procedures used, and the validity of the test results should be established for each test conducted for certification credit. *Test for certification credit* is defined in this chapter as system certification test conducted under an FAA-approved test plan for the purpose of showing compliance to the regulations. The FAA-approved test plan is the test plan approved before conducting an official FAA ground or flight test. It is not the Software Verification Plan referenced in RTCA/DO-178B. Examples of tests conducted to satisfy FAA certification credit are RTCA/DO-160D environmental qualification tests, system functional tests, systems integration tests, aircraft ground functional tests, and aircraft Type Inspection Authorization Tests (TIA) flight tests.

a. The ASE should perform the following tasks:

(1) Establish that the software baseline complies to its type design and released software plans by conducting FAA desk and/or on-site reviews (see chapter 2 of this order); or establish that the software DER (if delegated) has approved the baseline software by submitting a FAA Form 8110-3, "Statement of Compliance with the Federal Aviation Regulations." The DER should state in Form 8110-3 that the "purpose of Form 8110-3 is to approve the software baseline for the purposes of conducting FAA testing for certification credit."

Note: In some cases, special purpose software is used for environmental qualification testing. When this is the case, the manufacturer must verify, validate, and control the configuration of the special purpose test software. The test software should be included as part of the test setup conformity conducted before the qualification testing.

(2) Establish that the test configuration of the software to be installed in the Line Replaceable Unit complies with its software test baseline.

(3) Establish that all software artifacts associated with the test baseline are properly identified, under configuration control, and reflect the current state of the software under test.

(4) Establish that any software development tools or software verification tools that require qualification have been qualified. However, if the tool qualification activities are not completed at the time of conformity, the tools and supporting data should have their configuration documented.

(5) Initiate a FAA Form 8120-10, Request for Conformity, and submit it to the MIDO/MISO, providing instructions for the ASI to perform the following:

(a) Verify that the proper build and load file(s) was/were removed from the software configuration management (SCM) library.

(b) Verify that approved build and load instructions are followed during the software build and load process.

(c) Verify that any data integrity checks and software part numbers (including version numbers) are verified in the Line Replacement Unit.

(d) Verify that the test setup conforms to the test setup configuration identified in the approved engineering test plan.

(6) Establish that the procedure used for retention, archive, and retrieval of the software life cycle data is compliant with the approved SCM plan.

b. The ASI should perform those tasks mentioned in paragraph 4-3a(5) above and as identified on the Form 8120-10.

c. The software part conformity inspection should be successfully conducted before requesting a software installation conformity inspection.

4-4. Software Installation Conformity Inspection. A software installation conformity inspection is required anytime an FAA aircraft-level ground or certification flight test is performed, such as tests conducted per the TIA. The main objectives are to show that:

- An approved, controlled version of the software is loaded successfully into the target system in conformance with approved system installation procedures and/or software loading procedures, and

- The correct version for that system was loaded and will successfully initialize.

a. The ASE should ensure:

(1) A prior software part conformity inspection was successfully completed.

(2) Load procedures have been approved.

(3) A FAA Form 8120-10, Request for Conformity, or FAA Form 8110-1, Type Inspection Authorization (TIA), is initiated and contains the software part number and/or version number for which an installation conformity inspection is being requested. The software part number and/or version number should be identifiable, under configuration control, reproducible, and documented in the SCI or similar configuration documentation. The request should also include any actions/activities to be verified by the ASI including:

(a) Verification that the correct software version has been loaded into the system and that the correct system hardware (part numbers and serial numbers) has been installed on the aircraft.

(b) Verification that the loading procedure(s) ensures the correct software part number (and version number) is loaded into the correct system hardware components (serial numbers and part numbers). An error indication should result anytime that the software loading procedure or ground support equipment detects a mismatch of part and version numbers or an unsuccessful load. The installation conformity inspection should determine that the manufacturer's loading procedure(s) are followed and that the software load initializes correctly. Mismatches should be identified and documented.

b. The ASI should perform the software installation conformity inspection addressed in the FAA Form 8120-10, or FAA TIA Form 8110-1 (see item 4-4a(3) of this chapter) by one of two methods:

(1) By physically witnessing the successful loading of the correct software part number and version into the actual system (that is, actual part number and serial number) installed on the aircraft or to be installed on the aircraft. Successful load may be determined by witnessing that an integrity check was used to verify the software load (for example, comparison of cyclic redundancy checks (CRC)), and by witnessing that the software successfully executed the initialization procedure. The software loading process must be done in accordance with the software load procedures reviewed and approved by the ASE.

(2) By obtaining the manufacturing inspection records that document the results of the actual software loading. These records should include aircraft identification information, system

hardware part numbers and serial numbers, and software part numbers and version number, as applicable. The records provided should identify the hardware unit part number and serial number information so that the ASI (or designee, if delegated) can trace it to the system installed on the aircraft. The records provided should also show the software part number that was loaded into the system hardware. The records should indicate when and how the software was loaded and that the loading and initialization process was successful.

c. The software installation conformity inspection ensures that the system(s) installed on the aircraft and the software loaded into the system(s) for the purpose of conducting aircraft-level testing conforms to the FAA-approved type design data.

4-5. Summary. The purpose of a conformity inspection is to ensure that the product built (hardware and software) conforms to the type design. The two types of software conformity inspections addressed in this chapter are "software part conformity inspections" and "software installation conformity inspections." The responsibilities for ASEs and ASIs are identified for each of the two types of software conformity inspections addressed in this chapter. Software part conformity inspections and software installation conformity inspections are required whenever an applicant is to conduct laboratory system/hardware testing for certification credit as defined in paragraph 4-3, and during the installation of the system with the embedded software for the purpose of conducting aircraft-level ground and/or flight testing. The purpose of the aforementioned software conformity inspections is to ensure:

a. That the configuration of the unit under test reflects the correct hardware and software configuration that was approved for the given test being conducted for FAA certification credit.

b. That the configuration of the unit under test is well documented should there be any changes to the hardware and/or software after the tests have already been conducted.

c. That the systems installed on the aircraft and the software loaded into the installed systems for the purpose of conducting aircraft-level testing conforms to the FAA-approved type design.

d. That the final software and hardware configuration product baseline presented for certification conforms to the type design.

Chapter 5. Approval of Field-Loadable Software (FLS)

5-1. General. Through technological advances, the field loading of software has become a common process. This process reduces aircraft down-time for maintenance and increases efficiency of maintaining airborne equipment. RTCA/DO-178B, Section 2.5, provides some system design considerations for FLS; however, the existing guidance for approval of FLS through the TC, ATC, STC, ASTC, or TSO authorization processes is limited. This chapter provides additional guidelines for the certification authority or authorized DER approving FLS using the TC, ATC, STC, ASTC, or TSO authorization process. This chapter should be applied in conjunction with RTCA/DO-178B, Section 2.5. Chapter 6 of this order addresses the Parts Manufacturer Approval (PMA) process for FLS.

5-2. Approval of FLS. The following procedures should be carried out by the certification authority as part of the TC, ATC, STC, ASTC, or TSO authorization process for the approval of FLS:

a. Confirm that the software meets the objectives of RTCA/DO-178B or another acceptable means of compliance, as agreed to between the applicant and the certification authority.

b. Confirm that the considerations outlined in RTCA/DO-178B, Section 2.5, have been addressed.

c. Confirm that the software and hardware configurations were verified together during the verification process (that is, the software must be installed on the target computer in which the approval was granted).

d. Confirm that the applicant has a configuration management process in place to assure that the installation configuration (that is, the software part number, the hardware part number, the aircraft or engine model, and the aircraft or engine serial number combinations, as applicable) is the same configuration that was approved during the TC, ATC, STC, ASTC, or TSO authorization process.

e. If redundant parts on the aircraft or engine are field-loadable, confirm that the applicant has defined the following: (1) the requirements for intermixing different software loads on the parts, (2) requirements for partially successful and partially unsuccessful loads, and (3) the aircraft or engine dispatchability effects of successful and unsuccessful loads on redundant parts.

f. Confirm that there is a process in place to ensure that the software loaded is the software approved and that the software has not been corrupted (for example, verification with an appropriate data transfer integrity check, such as a CRC).

> **Note 1:** Per 14 CFR § 21.1(b), a "product" is an aircraft, an aircraft engine, or an aircraft propeller.

Note 2: Different data transfer integrity algorithms give different assurances that the data transferred are correct. The applicant should ensure that the algorithm used is sufficient for the integrity required for the software level of the data being loaded.

g. If there is no process in place to assure that paragraph 5-2f has been addressed, confirm during the verification process that the airborne equipment to be field loaded demonstrates compatibility with the onboard loading system. Additionally, the certification authority should ensure that the onboard loading system is approved considering the following items:

(1) The applicant should demonstrate that the onboard loading system complies with RTCA/DO-178B, Section 2.5, or an alternate means of compliance as agreed upon between the applicant and the certification authority.

(2) The applicant should provide documentation defining the operation of the onboard loading system and the recommended means for maintaining configuration control of equipment by the operator. This documentation should include guidelines for the configuration control processes that meet the guidelines outlined in this chapter.

(3) The applicant's onboard loading system and procedures should be approved by the certification authority. Depending on the implementation, this approval may include the data loader, as well as the procedures.

Note: Many approaches to data loading do not require evaluation of the data loader because integrity checks are built into the data and the data transfer process (see paragraph 5-2f of this chapter).

(4) If the applicant proposes more than one medium for onboard loading (such as diskette, mass storage, or compact disk), loading from all mediums should comply with the guidelines in this chapter.

h. For TC, STC, ATC, or ASTC projects, confirm that the applicant can verify the airborne equipment software part number with onboard equipment, carry-on equipment, or other appropriate means. For TSO projects, the appropriate part marking data (per 14 CFR § 21.607(d)) must be verifiable on the ground at any geographical location.

i. Confirm that changes to FLS will undergo a software change impact analysis to determine the safety impact and major or minor classification (unless the FLS is also user-modifiable software, which is addressed in chapter 7 of this order). Chapter 11 of this order provides additional guidelines on software change impact analysis.

j. Confirm that loading protection mechanisms are implemented to inhibit loading of FLS during flight.

Note: FLS that is also user-modifiable and has been approved by the certification authority as user-modifiable does not require further determinations of compliance for dissemination and installation (see RTCA/DO-178B, Section 2.4). Chapter 7 of this order provides additional guidelines for user-modifiable software.

5-3. FLS Installation Considerations. The approved FLS may be installed on the aircraft via Service Bulletin, Engineering Change Request, or other FAA-approved means. The approved means vary, depending on the method for granting approval. Whether the FLS approval is through TC, ATC, STC, ASTC, TSO authorization, or some other approval process, the document used to install the FLS should be approved by the certification authority and should specify the following elements:

a. The aircraft and hardware applicability and inter-mixability allowances for redundant systems software loading.

b. Verification procedures to assure that the software was correctly loaded into an approved and compatible target computer and memory devices.

c. Any post-load verification and/or test procedures required to show compliance to the guidelines specified in this chapter.

d. Actions to be taken in the event of an unsuccessful load (for example, prohibit dispatch of the aircraft).

e. Approved loading procedure or reference to approved loading procedure.

f. Maintenance record entry procedures required to maintain configuration control.

g. Reference to Aircraft Flight Manual, Aircraft Flight Manual Supplement, or Operator's Manual, as appropriate.

5-4. Maintenance and Part Marking Considerations. FLS maintenance and part marking should be performed in accordance with the appropriate part of 14 CFR (for example, §§ 45.15 and 21.607). Additional maintenance and part marking considerations that apply specifically to FLS using the TC, ATC, STC, ASTC, or TSO authorization process are discussed below:

a. The applicant's Aircraft Maintenance Manual or Instructions for Continued Airworthiness should include the procedures to be followed when conducting maintenance on airborne equipment using FLS.

b. The applicant's Aircraft Maintenance Manual or Instructions for Continued Airworthiness should include a procedure that requires maintenance personnel to verify the software part number configuration before and after maintenance is performed on the airborne equipment.

Note: If the software loading cannot be verified (for example, procedures do not render proper results, CRC fails, or part number does not match approved part number), the system should not be considered functional and the aircraft should not be dispatched. In some cases Minimum Equipment List (MEL) procedures may allow dispatch with some inoperative equipment. With equipment whose software part number cannot be verified, the MEL should specify whether the affected equipment may be disabled and the aircraft subsequently returned to service. Other means to clear the aircraft for dispatch depend on the MEL limitations.

c. There should be a process in place to ensure that maintenance personnel record loaded FLS part number in the necessary maintenance logs.

d. For airborne equipment having separate part numbers for hardware and software, the software part numbers need not be displayed on the outside of the unit, as long as it can be verified through some kind of electronic query. It is the maintenance personnel's responsibility to ensure that the software part number has been logged. When new software is loaded into the unit, the same requirement applies and the approved software part number should be verified before the unit is returned to service.

e. For airborne equipment with only one part number, which represents a specific configuration of software and hardware, the unit identification on the nameplate should be changed when the new software is loaded. When new software is loaded, the software part number stored in the target computer after data loading should be verified electronically. It should be verified that the electronic software part number and the unit part number displayed on the nameplate are an approved configuration before returning the unit to service.

f. When FLS is used in TSO-authorized articles and the applicant wants to use electronic part marking for the FLS, the FLS must meet the part marking requirements of 14 CFR § 21.607(d). The specific information required by 14 CFR § 21.607(d) must be verifiable in the aircraft at any geographic location on the ground, just as a hardware part number is identifiable on the ground.

g. If electronic part marking is used for FLS approved via the PMA process, the FLS must meet the part marking requirements of 14 CFR § 45.15 (see chapter 6 of this order for more information on PMA of FLS).

h. Changes to software part number, version, and/or operational characteristics should be reflected in the Operator's Manual, Aircraft Flight Manual, Aircraft Flight Manual Supplement, and/or any other appropriate document.

Chapter 6. Approval of Field-Loadable Software (FLS) by Finding Identicality Through the Parts Manufacturer Approval (PMA) Process

6-1. General.

a. To increase efficiency of field loads, software developers obtain PMA on their FLS to directly ship the software to the airline or operator. However, software does not fit the traditional concept of a "part." The diskette or CD-ROM serves only as the media that carries a transformable representation of the software's executable image. The desired approval is not for the media; it is for the data on the media after it has been loaded into the target computer (that is, the executable software itself). Since software does not fit the traditional definition of a part and has some unique considerations, this chapter provides additional guidelines to use the PMA process for FLS.

b. This chapter only addresses the PMA of FLS by identicality; it does not address PMA for FLS via the test and computation process. Should any issues regarding PMA of FLS via the test and computation process arise, please contact a software specialist in the Aircraft Engineering Division, Technical Programs/Continued Airworthiness Branch, at FAA Headquarters.

c. This chapter focuses on the manufacturing and production issues for PMA of the FLS. Chapter 5 of this order addresses guidelines for FLS development and approval, and should be applied in conjunction with this chapter.

6-2. Establishing Identicality.

a. The PMA is used for replacement or modification parts for sale for installation on a type certificated product. Design approval using the PMA process may be accomplished in two ways: (1) by showing that the design is identical to a previously FAA-approved design, or (2) by submitting test results and computations (data) showing that the design meets all applicable airworthiness requirements. This chapter addresses the process for approving FLS using the identicality approach. The test and computation approach is not addressed in this chapter.

b. Identicality can be established in one of two ways: (1) by showing evidence that the applicant obtained the design through licensing agreement, or (2) by comparing the applicant's design to a previously approved design. PMA for FLS should follow the same procedures outlined in 14 CFR part 21 and Order 8110.42, with the following additional considerations unique to software:

(1) Finding of identicality by showing evidence of a licensing agreement.

(a) Design Approval. Order 8110.42, Paragraph 9a(3)(a) pertaining to licensing agreement states that the PMA applicant should submit "an appropriate document from the TC

holder authorizing use of the submitted data package." The following items should be considered for PMA design approval via the licensing agreement method:

1. FLS to be approved via PMA should have been previously approved by the FAA through the TC, ATC, STC, or ASTC process and should have the procedures in place discussed in chapter 5 of this order.

2. The approved software may be installed on the aircraft using a Service Bulletin or some other FAA-approved means.

3. There should be a configuration management process in place to assure that the combination of the software part number, the hardware part number, the aircraft model(s), and the aircraft serial number(s), as appropriate, is the same combination that was approved during the TC, ATC, STC, or ASTC process.

(b) Design Changes. Order 8110.42, Paragraph 9g(5) addresses the situation of design changes for PMA. For FLS that was approved via the PMA by showing evidence of licensing agreement, apply the following guidelines:

1. Coordinate changes to FLS with the TC, ATC, STC, or ASTC holder and certification authority to assess if the effect of the change on the aircraft is major or minor. Major or minor change classification is described in 14 CFR §§ 21.93. Chapter 11 of this order provides guidelines on using the software change impact analysis process to determine major or minor classification.

2. Paragraph 9g(5)(a) of Order 8110.42 states that major changes "must be substantiated and approved prior to implementation in the same manner as that for the original PMA."

3. If the change is determined to be minor, follow the procedure defined in Order 8110.42, Paragraph 9g(5).

(2) Finding of identicality without a licensing agreement.

(a) Design Approval. Order 8110.42, Paragraph 9a(3)(b) states that the applicant's identicality statement must certify that the "design is identical in all respects to the design of the part covered under an approved design." The following items should be considered for PMA design approval using identicality without a licensing agreement:

1. The FLS to be approved must be proven to be identical to software previously approved by the FAA through the TC, ATC, STC, or ASTC process.

> **Note:** The FLS originally approved as part of the TC, ATC, STC, or ASTC process should have procedures in place as discussed in chapter 5 of this order and Section 12.5 of RTCA/DO-178B.

2. Design identicality may be demonstrated through some form of bit-by-bit check to show that the electronic image of the software is exactly the same.

3. In addition to the bit-by-bit check, there should be design evidence available to support the identicality claim. Evidence of design identicality includes availability to all software development and design data required as part of the original approval. The data required by RTCA/DO-178B or other acceptable means of compliance should be made available to the FAA to ensure identicality. This includes such items as Software Requirements Data, Design Description, Source Code, Executable Object Code, Software Configuration Index, and Software Accomplishment Summary, as listed in Section 9.4 of RTCA/DO-178B. The presence of this design data is necessary to demonstrate that the software development process is identical and to support continued airworthiness.

(b) Design Changes.

1. Design changes to FLS by identicality without a licensing agreement should be considered major.

2. Paragraph 9g(5)(a) of Order 8110.42 states that major changes "must be substantiated and approved prior to implementation in the same manner as that for the original PMA."

(c) The FAA and DER responsibilities. Responsibilities for the PMA for FLS are the same as outlined in Order 8110.42 (that is, the ASI or authorized designee addresses identicality by licensing agreement; while the ASE or authorized DER addresses other PMA approaches).

c. In a PMA for hardware, type design is established by the engineering drawing. However, the approach for software may be different. The following approaches should be considered by ASIs when granting PMA for FLS:

(1) The top-level engineering drawing may be accomplished via a SCI. Therefore, it is acceptable to list the SCI and its release date on the PMA supplement as the type design data.

(2) If the SAS is not included in the SCI, it should also be included on the PMA supplement.

(3) Some projects may have a higher-level drawing that references the SCI. If this is the case, the higher-level drawing may be included on the PMA supplement instead of the SCI.

6-3. Applicability to TSO. The applicability of the PMA to a unit containing FLS with TSO authorization is the same as discussed in 14 CFR, part 21, subpart O and Order 8110.42. If the PMA process is used for a unit with TSO containing FLS, it should follow the guidelines of this chapter, in conjunction with 14 CFR, part 21, subpart O and Order 8110.42.

This Page Intentionally Left Blank

Chapter 7. Approval of Airborne Systems and Equipment Containing User-Modifiable Software (UMS)

7-1. General. This chapter applies to UMS only. It does not apply to option-selectable software or field-loadable software, except where such software is also user-modifiable.

7-2. Safety Considerations.

a. UMS is software within an airborne system approved for user modification. Users (such as airlines and operators) may modify UMS within the specified modification constraints and with approved modification procedures without any further involvement by the certification authority. It is intended that once the system with the UMS has been certified, the ACO should require no further visibility, review, or approval of modifications made to that UMS component. Therefore, modification of the UMS by the user should have no effect on the aircraft safety margins, aircraft operational capabilities, flight crew workload, any non-modifiable software components, or any protection mechanisms of the system.

> **Note:** Some modifications to UMS by the user may require an operational approval or acceptance, for example, modifications to operators' aircraft-specific operating procedures on data used in performing those procedures.

b. A UMS component is software within the airborne system that is designed and intended to be changed by the user. A non-modifiable software component is not designed or intended to be changed by the user. Modification constraints for UMS should be developed by the applicant and provided to the users. Any changes to UMS that affect the following items warrant rescinding the classification of the software as user-modifiable, and requires design approval under the applicable regulations:

(1) Safety margins, operational capabilities, flight crew workload, any non-modifiable software components, protection mechanisms, and software boundaries.

(2) Pre-approved range of data, parameters, or equipment performance characteristics.

> **Note:** Multiple trim values used as UMS that may affect safety require special attention. In general, it is not acceptable to simply test the trim value throughout its trim range, because of the uncertainty for acceptability of all the combinations of the trims. In most cases, it is not possible to verify all possible combinations of multiple trims. Therefore, in the case of multiple trims used as UMS, acceptance of verified sets of trims is generally required.

c. The potential effects of UMS modification must be determined by the safety assessment process and mitigated by system and software design means, development and verification assurance, approved procedures, and approved tools (if applicable). When evaluating data as part of the RTCA/DO-178B process, the applicant and the approving ACO should ensure that the

protective mechanisms, verification, and user-modification procedures do not interfere with the non-modifiable components and protection integrity. The applicant should obtain the concurrence of the ACO early in the program as to the acceptability of the protective mechanism, protection verification, and modification procedures and tools.

> **Note:** The purpose of the protective mechanism is to ensure that the user-modifiable component does not interfere with the non-modifiable component. This protective mechanism should be evaluated during the initial approval of the system containing UMS. It should be assured that no modification of the software by the user affects the protective mechanism. Paragraphs 7-5 and 7-6 of this chapter will further address protection.

7-3. Considerations for Displayed Data. Where information is displayed to the flight crew and is derived from UMS, the information should be identified to distinguish it as "advisory data only" that has not been approved as part of the aircraft type design by the certification authority. If the information displayed has received an operational approval as part of the operational procedures of the aircraft by an appropriate operation approval authority, this distinction may not be necessary. If the design or inherent nature of the equipment or user-modifiable component makes the distinction between approved and unapproved information so readily apparent to the flight crew that errors distinguishing the two types of information are reasonably precluded, explicit identification of the information as "advisory data only" may not be required. Such identification, where required, should be provided by a non-modifiable component and allow the flight crew to readily distinguish between information approved or accepted by the certification or operational approval authority. "Advisory data only" information should be verifiable by the flight crew from another source on the aircraft, should not be used to display any information where the potential worst case failure condition for displaying misleading data is any greater than minor, or should not be used by the flight crew in performing any aircraft operational procedures (for example, supplemental situational awareness only).

7-4. Modification of Aircraft Performance Parameters. Modifications that could affect the safety margins, operational capabilities of the aircraft, or crew workload include modifications of displayed data or other data used by the flight crew to determine aircraft performance parameters. These types of modifications require certification authority approval. Modification of the user-modifiable component to provide or revise these parameters, regardless of whether they are provided as primary or advisory information, requires certification authority approval. Such a change would warrant rescinding the classification of the software as user-modifiable and would require design approval and part number revision.

7-5. Protection. Non-modifiable software components of the airborne system should be protected from UMS components. The system requirements should specify the protection mechanisms that prevent the user modification from affecting system safety, operational capability, or flight crew workload. If the system requirements do not include provisions for user modification, the user should not modify the software. The protection mechanism should be assigned the assurance level of the most severe failure condition of the system as determined by the system safety assessment. If software provides the protection mechanism for UMS, that

software protection should be assigned the highest software level of the system as determined by a system safety assessment. The protection should prevent any modification or failure of the UMS from causing loss of protection. Protection integrity cannot depend on any activities of the user. The protection integrity should be such that it can neither be breached accidentally nor intentionally. The applicant-provided means of modification of the UMS should be the only means to change the modifiable component.

7-6. Tools Used To Protect Non-Modifiable Components.

a. RTCA/DO-178B, Section 5.2.3, requires that the non-modifiable software components be protected from modifiable components to prevent interference with the safe operation of the non-modifiable software components. To enforce this protection, tools are allowed to make changes to the modifiable component. If such tools will be used to enforce this protection, then the following information should be provided by the applicant to the certification authority for approval:

(1) Plans for controlling tool version;

(2) Plans for controlling tool usage;

(3) Plans for qualifying or verifying the tool (see RTCA/DO-178B Section 12.2 and chapter 9 of this order); and

(4) Procedures for modifying the tool.

b. Software forming a component of the tool and used in the protective function should be developed to the software level of the most severe failure condition of the system, as determined by a system safety assessment.

c. Use of software tools for user modifications requires tool qualification (see RTCA/DO-178B Section 12.2 and chapter 9 of this order) and approval of procedures to use and maintain the tool. Changes to the tool or procedures may require re-qualification of the tool.

7-7. Data Requirements.

a. Applicants should identify in the PSAC their intention to develop an airborne system that will contain a UMS component(s). The PSAC should also describe: (1) the means of complying with RTCA/DO-178B (including the design considerations of Section 5.2.3), (2) the protection mechanism, and (3) the means of ensuring the integrity of the protection mechanisms. If software tools will be used for the modification, the PSAC should also identify tool qualification plans or verification procedures to ensure that the tool has modified the UMS to approved procedures and constraints, and it has not affected the non-modifiable software or protection mechanisms.

b. The Software Development Plan and design data should specify the design methods and details of implementation for ensuring protection from user modifications.

c. The SCI should identify the approved procedures, methods, and tools for modifying the UMS, including tool qualification data, if applicable.

d. The SAS should summarize the entire development and verification of the non-modifiable software components, UMS component(s), protection mechanism, and modification procedures and tools, including tool qualification, if applicable.

7-8. Other Considerations. At the time of user modification, the user assumes responsibility for all aspects of the UMS components and tools used for modifying the software. These include software configuration management, SQA, and software verification. User modifications should be performed to approved procedures established by the system requirements and software data, using approved tools. If the user makes any modification to the non-modifiable software components, the protection mechanisms, the approved procedures, or the approved tools (other than those established by the system requirements and approved procedures), then they have violated the type design, and the type certificate of the aircraft may be rescinded.

Note 1: During certification, the ACO should coordinate with the regulatory authorities responsible for approving changes to the aircraft configuration in the field (for example, operational approvals). This helps ensure the practicality and acceptability of the tools and procedures used to control the aircraft configuration.

Note 2: A system to track or log software modification should be considered (where appropriate) so that both the Certification and Continued Airworthiness aspects of the modifications may be reviewed by the cognizant authorities, as needed.

Chapter 8. Previously Developed Software (PDS) – Applying RTCA/DO-178B Level D Criteria

8-1. General.

a. A number of RTCA/DO-178B objectives cause confusion when applying them to PDS. This chapter provides guidelines to apply RTCA/DO-178B to PDS that is categorized as Level D – the guidelines do not apply to other software levels.

b. RTCA/DO-178B provides for five different levels of software based on the software's contribution to potential failure conditions. These software levels represent differing levels of development process rigor based on the severity of the potential failure conditions to which the software can cause or contribute. Level D is assigned to software that can cause or contribute to no more than a minor aircraft failure condition. RTCA/DO-178B contains 28 objectives for Level D software that should be satisfied before approval is granted.

c. To be consistent with a minor aircraft failure condition, the primary intent of Level D software objectives is to provide a thorough investigation of the functional behavior of the software and to provide the necessary configuration control. However, some of the objectives for Level D are difficult to understand when considered with the overall objective of establishing correct functional behavior.

d. Many developers may decide to do more than the objectives for Level D; however, this chapter concentrates on the minimum objectives to be satisfied. Proper application of Level D objectives permits the use of PDS, which is software that was not originally approved using RTCA/DO-178B (such as commercial-off-the-shelf software, software developed using military standards, software developed using RTCA/DO-178 or RTCA/DO-178A, and software developed using other industry standards).

e. See Section 12.1 of RTCA/DO-178B for additional guidance on using PDS. In particular, see Section 12.1.4 for additional considerations when upgrading a previous development baseline.

8-2. Five Misinterpreted Objectives. A consistent interpretation of RTCA/DO-178B for Level D software is important for the approval of PDS software. Of the 28 objectives found in Annex A of RTCA/DO-178B for Level D software, experience has shown that five objectives are frequently difficult to understand. One of the objectives is related to the integral processes; the remaining four objectives are related to source code, software architecture, and low-level requirements. The discussion below provides clarification of RTCA/DO-178B Level D objectives for PDS approval consideration. Paragraph 8-3 of this chapter provides specific procedures for the approval of Level D PDS.

a. Objective 1 in RTCA/DO-178B, Annex A, Table A-1, "Software development and integral processes activities are defined." Some applicants believe that Objectives 1 and 6 of

RTCA/DO-178B, Annex A, Table A-1 ("Software development and integral processes activities are defined" and "Software plans comply with this document") conflict for Level D. These applicants contend that since software plans do not comply with RTCA/DO-178B, those plans are not needed. However, objectives of RTCA/DO-178B ensure that even for Level D software: (1) there are some plans (for example, Plan for Software Aspects of Certification, Software Development Plan, Software Configuration Management Plan, Software Quality Assurance Plan, Software Verification Plan), even if the plans themselves do not comply with RTCA/DO-178B (see Objective 1 in RTCA/DO-178B, Annex A, Table A-1), and (2) those plans are followed (see Objective 1 in RTCA/DO-178B, Annex A, Table A-9). Additionally, the plans should enable compliance to the RTCA/DO-178B objectives applicable for Level D software (see Section 4.3 of RTCA/DO-178B).

b. Objective 4 in RTCA/DO-178B, Annex A, Table A-2, "Low-level requirements are developed." For Level D software, the intent of this objective is to assure that the low-level requirements are defined. However, Table A-4 contains no objectives related to explicit verification of the low-level requirements for Level D software, except for verifying the integrity of any software partitioning. Therefore, Objective 4 of Table A-2 is satisfied implicitly by satisfying Objectives 1 and 2 in RTCA/DO-178B, Annex A, Table A-6. The satisfaction of Objectives 1 and 2 demonstrates that the executable code complies with and is robust with high-level requirements. Since there is no objective for Level D to ensure that the executable code complies with the low-level requirements, it is not necessary to ensure that the low-level requirements are developed and traceable to the high-level requirements**.**

c. Objective 3 in RTCA/DO-178B, Annex A, Table A-2, "Software architecture is developed." The logic applied in paragraph 8-2b above may also be applied to Objective 3 (that is, Objective 3 is implicitly satisfied by other objectives and does not need to be explicitly satisfied for Level D PDS, since Table A-4, Objectives 8 through 12 do not require verification of the software architecture).

d. Objective 5 in RTCA/DO-178B, Annex A, Table A-2, "Derived low-level requirements are defined." Objective 5 (Paragraph 5.2.1b of RTCA/DO-178B) states that "Derived low-level requirements are provided to the system safety assessment process," rather than just "defined." As with the low-level requirements and software architecture, there is no objective for explicit verification of derived low-level requirements for Level D software. The satisfaction of this objective is implied by satisfying Objective 2 in RTCA/DO-178B, Annex A, Table A-2, "Derived high-level requirements are defined" and the associated verification of high-level requirements.

e. Objective 6 in RTCA/DO-178B Annex A, Table A-2, "Source code is developed." Objective 6 (Paragraph 5.3.1a of RTCA/DO-178B) states, "Source code is developed that is traceable, verifiable, consistent, and correctly implements low-level requirements." However, according to Annex A, Table A-5, there are no verification objectives for Level D source code. Therefore, there is no verification review objective to establish consistency between source code, low-level requirements, and high-level requirements. The consistency objective is between the executable code and the high-level requirements for Level D: the objective is for the executable

code to meet all functional requirements. Furthermore, the existence of object code implies the existence of source code so that Objective 6 of RTCA/DO-178B, Annex A, Table A-2 is reasonably covered by satisfying other objectives (that is, Objectives 1 and 2 of Table A-2; Objective 2 of Table A-3; Objectives 1 and 2 of Table A-6; and Objective 3 of Table A-7) for level D software.

8-3. Approving Level D PDS. For a project involving approvals of Level D PDS, the certification authority and/or the DER (if authorized) should follow the procedures listed below:

a. Software reviewers should review the software plans to assure that:

(1) Some plans exist (for example, Plan for Software Aspects of Certification, Software Development Plan, Software Configuration Management Plan, Software Quality Assurance Plan, Software Verification Plan);

(2) Those plans are followed (see RTCA/DO-178B, Annex A, Table A-9, Objective 1); and

(3) The plans enable compliance to RTCA/DO-178B objectives for Level D software.

b. Software reviewers should ensure that low-level requirements, software architecture, derived low-level requirements, and source code are defined and exist for Level D PDS. The software reviewers should not assess the quality or content of these software life cycle data items to RTCA/DO-178B objectives and software life cycle data content requirements, except where necessary to ensure that software partitioning integrity is confirmed (Objective 13 of Table A-4). The intent of these objectives will be satisfied by Table A-6 and A-7 objectives.

c. When evaluating the PDS, the following steps should be carried out by the applicant and confirmed by the certification authority:

(1) Verify that a failure condition or malfunction of the Level D software can cause or contribute to no worse than a minor failure condition. The certification authority should confirm the safety assessment, system architecture, and software level determination.

(2) Identify the functions to be used from the PDS, any PDS components to be integrated, and any software developed to specifically mitigate any failures or malfunctions of the PDS (for example, wrapper code, partitioning, or monitors). The certification authority should confirm that safety implications are addressed.

(3) Ensure that the PDS cannot result in any unacceptable failure condition in the target application. The certification authority should confirm this assessment.

d. Where software applications of multiple software levels are contained in a given system and/or component, the protection and associated mechanisms between the different software levels (such as partitioning, safety monitoring, or watchdog timers) should be verified to meet the

objectives of the highest level of software of the system and/or component. This can occur when there are multiple functions in a component (such as maintenance and navigation) or when there are different categorizations of types of failure conditions, such as loss of function versus a corrupted function (for example, misleading display data). An example of the latter case is a navigation system supported by a PDS operating system. The loss of the navigation function can be shown to produce only a minor aircraft failure condition, whereas misleading navigation is usually considered to be a major aircraft failure condition. If the navigation function is protected (partitioned) from the operating system in such a way that any failure of the operating system can be shown to produce only a loss of function, then the operating system only needs to be evaluated to Level D criteria. However, the applicant needs to verify that the operating system can only contribute to loss of navigation function and not to a misleading navigation failure condition. The applicant also needs to verify that common-cause and common-mode losses of identical functions or common resources cannot result in a worse failure condition than was originally assigned to the individual system. In this case, part of the development effort would be to demonstrate that the PDS can be shown to meet all the Level D objectives, as outlined above.

e. It is possible for Level D software to operate in conjunction with software of other levels. If so, a thorough protection/partitioning analysis should be performed in conjunction with the system safety assessment. However, discussion of protection/partitioning is outside the scope of this order and will not be discussed further.

f. See RTCA/DO-178B, Section 12.1, for additional guidance on the use of PDS.

Chapter 9. Qualification of Software Tools Using RTCA/DO-178B

9-1. General. Section 12.2 of RTCA/DO-178B states that qualification of a tool is needed when processes in RTCA/DO-178B "are eliminated, reduced, or automated by the use of a software tool, without its output being verified as specified in section 6" of RTCA/DO-178B. RTCA/DO-178B states, "The objective of the tool qualification process is to ensure that the tool provides confidence at least equivalent to that of the process(es) eliminated, reduced, or automated." The paragraphs below provide further information regarding tool qualification:

a. Software development can be a very repetitive and human-labor intensive process. This can result in errors, as well as high costs. For these reasons various tools have been developed to automate portions of this process. If the tools are dependable, then improvements in productivity and lower numbers of in-service errors may be realized.

b. To certify systems developed with tool support, the FAA, DERs, and applicants need to obtain confidence by qualification that these tools are dependable. RTCA/DO-178B, Section 12.2 was designed to provide criteria for establishing which tools require additional confidence and the criteria and data needed to establish that confidence. However, several provisions of this section are difficult to interpret. This chapter clarifies the intent of RTCA/DO-178B, Section 12.2 and its application.

c. Some areas that will be clarified are:

(1) When a tool should be qualified.

(2) Justification for the different criteria for qualifying software development tools and software verification tools.

(3) Which criteria apply to software development tools and which apply to software verification tools.

(4) Data to be produced for software development tools and for software verification tools.

(5) Acceptance criteria for tool operational requirements.

(6) Tool determinism.

(7) Tool partitioning assurance and evidence.

(8) Tool configuration control.

9-2. Two Kinds of Tools That May Be Qualified.

a. Not all software tools require qualification. According to RTCA/DO-178B Section 12.2, qualification of a tool is needed only when processes described in RTCA/DO-178B are eliminated, reduced, or automated by the use of that tool without its output being verified as specified in RTCA/DO-178B, Section 6. This means that if the results of the tool are being relied on to supply the sole evidence that one or more objectives are satisfied, the tool must be qualified per RTCA/DO-178B, Section 12.2. If the result of the verification activity performed by the tool is confirmed by another verification activity, then there is no need to qualify the tool.

b. RTCA/DO-178B, Section 12.2 identifies two types of tools: software verification tools and software development tools. Each type will be discussed below.

c. RTCA/DO-178B defines verification tools as "tools that cannot introduce errors, but may fail to detect them."

(1) The following are examples of verification tools:

(a) A tool that automates the comparison of various software products (such as code or design) against some standard(s) for that product.

(b) A tool that generates test procedures and cases from the requirements.

(c) A tool that automatically runs the tests and determines pass/fail status.

(d) A tool that tracks the test processes and reports if the desired structural coverage has been achieved.

(2) Many claim that verification tools can be more reliable than humans in a number of verification tasks, if their correct operation is demonstrated. To encourage the use of verification tools, RTCA/DO-178B Section 12.2 was designed to provide an acceptable approach to qualifying verification tools.

d. RTCA/DO-178B defines development tools as "tools whose output is part of airborne software and thus can introduce errors." If a tool can generate an error in the airborne software that would not be detected, then the tool cannot be treated as a verification tool. An example of this would be a tool that instrumented the code for testing and then removed the instrumentation code after the tests were completed. If there was no further verification of the tool's output, then this tool could have altered the original code in some unknown way. Typically, the original code before the instrumentation is what is used in the product. This example demonstrates that tools used during verification are not necessarily verification tools. The effect on the final product should be assessed to determine the tool's classification.

e. The reason for the distinction between development and verification tools is based on the likelihood of allowing an error into the airborne system. For development tools there is a

potential to introduce errors directly into a system. However, a verification tool can only fail to detect an error that already exists in the product. Therefore, tools need to be deficient in two different processes to allow an error to get into the airborne software: the development process introducing the error and the verification process failing to detect the error. This is why, RTCA/DO-178B calls for different levels of rigor in the qualification of verification and development tools.

f. The remaining paragraphs of this chapter provide guidelines for certification authorities and authorized DERs to consider, when qualifying software tools.

9-3. Determining Whether A Tool Should Be Qualified.

a. Whether a tool needs to be qualified is independent of the type of the tool (development or verification). There are three questions to ask to determine if a tool needs qualification. If the answer is "Yes" to all of the questions below, the tool should be qualified:

(1) Can the tool insert an error into the airborne software or fail to detect an existing error in the software within the scope of its intended usage?

(2) Will the tool's output not be verified or confirmed by other verification activities, as specified in Section 6 of RTCA/DO-178B?

(3) Are processes of RTCA/DO-178B eliminated, reduced, or automated by the use of the tool? That is, will the output from the tool be used to either meet an objective or replace an objective of RTCA/DO-178B, Annex A?

b. Once it has been determined that a tool does not require qualification, the remainder of RTCA/DO-178B, Section 12.2 is not applicable to that tool. To ensure a timely response, the certification authority or DER (if authorized) should be involved early in the certification project's tool qualification agreements.

c. The PSAC should include a listing of all software tools and justification for why each tool does or does not require qualification.

> **Note:** The inclusion of all software tools in the PSAC is encouraged to provide early visibility of tools that may require qualification.

9-4. Determining Which Tool Qualification Criteria Apply. Figure 9-1 below applies to tools requiring qualification and can be used to determine which criteria of RTCA/DO-178B, Section 12.2 apply to which type of tool. Figure 9-1 shows the similarities and differences in the qualification criteria for development and verification tools. The column in figure 9-1 titled "Criteria" summarizes the RTCA/DO-178B requirement; the column titled "Dev./Ref." lists the applicability of the criteria for development tools and the appropriate RTCA/DO-178B section reference; and the column titled "Verif./Ref." lists the applicability of the criteria for verification tools with the appropriate RTCA/DO-178B section reference.

Figure 9-1. RTCA/DO-178B Criteria Applicable to Tool Qualification

Criteria	Dev./Ref.	Verif./Ref.
Only deterministic tools may be qualified (to be further clarified in paragraph 9-6d of this chapter).	Yes/12.2	Yes/12.2
Qualification should only be for a specific system; the intention should be stated in the Plan for Software Aspects of Certification.	Yes/12.2	Yes/12.2
Combined tools should be qualified to RTCA/DO-178B, Section 12.2.1 unless partitioning can be shown (to be further clarified in paragraph 9-6e of this chapter).	Yes/12.2.b	Yes/12.2.b
Software configuration management and software quality assurance process objectives should be applied to tools being qualified (to be further discussed in paragraph 9-6f of this order).	Yes/12.2.c	Yes/12.2.c
Qualification should satisfy the same objectives as the airborne software.	Yes/12.2.1.a	No
The software level of the tool may be reduced.	Yes/12.2.1.b	No
A trial period may be used as a means to demonstrate compliance with the tool operational requirements.	Yes/ 12.2.1.c	Yes/12.2.2
Tool Operational Requirements should be reviewed.	Yes/12.2.1.d(1)	Yes/12.2.2
Compliance with Tool Operational Requirements under normal operating conditions should be demonstrated.	Yes/12.2.1.d(2)	Yes/12.2.2
Compliance with Tool Operational Requirements under abnormal operating conditions should be demonstrated.	Yes/12.2.1.d(3)	No
Requirements-based coverage should be analyzed.	Yes/12.2.1.d(4)	No
Structural coverage appropriate for the tool's software level should be completed.	Yes/12.2.1.d(5)	No
Robustness testing appropriate for the tool's software level should be completed.	Yes/12.2.1.d(6)	No
Potential errors should be analyzed.	Yes/12.2.1.d(7)	No

9-5. Guidelines for Data Submittal and Data Availability to Demonstrate Tool Qualification.

a. The guidelines for data to support tool qualification are listed throughout RTCA/DO-178B, Section 12.2; however, there is no definitive guidance for the minimum level/amount of data to be submitted to the FAA for tool qualification. The data submittals vary according to the type of tool being developed. Even though there are some similar guidelines for the two tool types, the data requirements for each are different. Figure 9-2 summarizes the tool qualification data. The "Data" column lists the data for tool qualification. The "Applicability"

column summarizes if the data apply for development tool qualification (Development) or verification tool qualification (Verification). The "Available/Submit" column summarizes if the data should be submitted to the FAA or just available for FAA review. The column titled "RTCA/DO-178B Ref." lists the RTCA/DO-178B section(s) referencing the criteria. The remainder of this chapter discusses the tool qualification data summarized in figure 9-2.

Figure 9-2. Data Required for Tool Qualification

Data	Applicability	Available/ Submit	RTCA/DO-178B Ref.
Plan for Software Aspects of Certification (PSAC)	Verification & Development (see Note 1 below)	Submit	12.2, 12.2.3.a, & 12.2.4
Tool Qualification Plan	Development (see Note 2 below)	Submit	12.2.3.a(1), 12.2.3.1, & 12.2.4
Tool Operational Requirements	Verification & Development	Available	12.2.3.c(2) & 12.2.3.2
Software Accomplishment Summary (SAS)	Verification & Development (see Note 1 below)	Submit	12.2.4
Tool Accomplishment Summary	Development (see Note 2 below)	Submit	12.2.3.c(3) & 12.2.4
Tool Verification Records (for example, test cases, procedures, and results)	Verification & Development	Available	12.2.3
Tool Qualification Development data (for example, requirements, design, and code)	Development	Available	12.2.3

Note 1: For development tool qualification, the PSAC should reference the Tool Qualification Plan, and the SAS should reference the Tool Qualification Accomplishment Summary.

Note 2: For verification tool qualification, the applicant can develop a Tool Qualification Plan and a Tool Qualification Accomplishment Summary

b. Verification Tool Qualification Data. Of the two tool qualification types, verification tools require the fewest data submittals and availability. Data for verification tool qualification are discussed below:

(1) For verification tools, the applicant should specify the intent to use a verification tool in the PSAC (see RTCA/DO-178B, Section 12.2). The PSAC should be submitted to the FAA and should include the intended tool qualification schedule. This alerts the certification

authority to respond to the intended use of the tool and opens a dialogue on acceptable qualification methods and documentation approaches. The certification authority and/or DER (if authorized) should provide a written response to the applicant on the acceptability of the approach listed or referenced in the PSAC in a timely manner (that is, the verification tool qualification approaches in the PSAC should be reviewed and approved or addressed in a timely manner).

(2) For verification tool qualification, the Tool Operational Requirements should be documented and available to the FAA (see RTCA/DO-178B, Section 12.2.3.2). The requirements for the Tool Operational Requirements data are discussed in paragraph 9-6a of this chapter.

(3) Data showing that all of the requirements in the Tool Operational Requirements have been verified should also be documented and available for FAA review. Sufficient verification data are needed to demonstrate normal operation only and will vary depending on the complexity and purpose of the tool, and how it is used. The applicant may package these verification data in any document they choose.

(4) An entry summarizing the results of the verification tool qualification should be included in the SAS. The SAS should be submitted to the FAA. This allows the certification authority to approve the results of the verification data and is evidence of the tool's qualification status.

> **Note:** The applicant may choose to provide a separate Tool Qualification Plan and Tool Accomplishment Summary referenced by entries in the PSAC and the SAS for software verification tools. Entries are still required in the PSAC and SAS. This is an acceptable approach and has the added benefit of permitting reference to a data package for reuse in subsequent certifications or in different certifications where the usage of the tool can be shown to be identical.

c. Development Tool Qualification Data. There are additional qualification criteria for a software development tool. The criteria for qualifying a software development tool are similar to the approval process for the airborne software. For the software development tool qualification, consider the following data submittal and availability items:

(1) The actual qualification approach and data to be provided are specified in the Tool Qualification Plan. The Tool Qualification Plan should be submitted by the applicant for FAA approval.

(2) The Tool Accomplishment Summary should also be submitted to the FAA. It summarizes the results of the tool qualification process and describes and references the relevant tool qualification data.

(3) The PSAC and SAS should be submitted by the applicant for FAA approval. However, these documents will likely only reference the Tool Qualification Plan and the Tool Accomplishment Summary documents.

(4) The Tool Operational Requirements should be documented and available to the FAA (see RTCA/DO-178B, Section 12.2.3.2). The requirements for the Tool Operational Requirements data are discussed in paragraph 9-6b of this chapter.

(5) Data that show that all requirements in the Tool Operational Requirements have been verified should also be documented and made available for FAA review. Sufficient verification data are needed to demonstrate tool operation under normal and abnormal operation conditions. The data will vary depending on the complexity of the tool, the purpose of the tool, and how the tool is used. The applicant can package this verification data in any document they choose.

(6) Other tool qualification data, such as design, code, test cases, and procedures should be available for FAA review.

d. Document Format and Media Type. The certification authority and/or DER (if authorized) should strive to use the document format and media used by the applicant. Any repackaging for submittal to the FAA should be undertaken only when the FAA cannot review the data in the manner presented by the applicant or the applicant cannot meet the data retention provisions of the applicable 14 CFR sections.

9-6. Guidelines for Evaluating Acceptability of Tool Operational Requirements Data. Tool Operational Requirements for any tool that requires qualification should be completed and made available for FAA review. A complete set of operational requirements is necessary to communicate to both the user and the certification authority (or authorized DER) what the tool does, how it is used, and the environment in which it performs. The Tool Operational Requirements should identify all functional and technical features of the tool and the environment in which it is installed (see RTCA/DO-178B, Section 12.2.3.2). The information required is different depending on the type of tool:

a. For a verification tool, the Tool Operational Requirements should provide at least the following information:

(1) The tool's functionality in terms of specific requirements verified as part of the tool's qualification tests.

(2) A definition of the tool's operational environment, including operating system and any other considerations (for example, an analysis of what the tool will not do and what is required to cover that shortage (such as extensions to checklists, test cases) and any specialized hardware requirements (such as processors, special test equipment, or interfaces)).

(3) Any other information necessary for the tool's installation or operation (such as User's Manual) should be included in the Tool Operational Requirements.

b. A development tool needs to include all the information listed above for verification tools but should also include at least the following:

(1) Software development processes performed by the tool.

(2) Expected response under abnormal operating conditions.

> **Note:** In some cases the User's Manual or other supplier's documentation may contain the needed information. Where additional information is included over and above the required information, the required information should be clearly identified. In the case where there is insufficient information from the tool supplier, the applicant should provide the missing information.

c. Guidelines on acceptable verification of the Tool Operational Requirements: Development and verification tools require verification of the Tool Operational Requirements. For verification tools, only verification over the normal operating conditions is required; for development tools, verification over the abnormal operating conditions is also required. RTCA/DO-178B, Sections 6.4.2.1 and 6.4.2.2 describe verification for normal and abnormal conditions and will not be covered in this chapter. However, since the operational requirements may contain additional information not directly related to the verification activity (such as the appearance of menus, dialog boxes, and configuration), additional guidance is needed to reduce unnecessary verification for verification tools. For verification tools only, those portions of the operational requirements used directly in the setting up, conducting, monitoring, and reporting of verification need to be verified as part of tool qualification. The applicant should ensure that those features/portions of the verification tool that are not used, have no adverse effect on the features/portions being used. If additional features are used later, additional verification will be required.

d. Guidelines on the interpretation of the determinism of tools:

(1) Although only deterministic tools can be qualified (see Section 12.2.3 of RTCA/DO-178B), the interpretation of determinism is often too restrictive. A restrictive interpretation is that the same apparent input necessarily leads to exactly the same output. However, a more accurate interpretation of determinism for tools is that the ability to determine correctness of the output from the tool is established. If it can be shown that all possible variations of the output from some given input are correct under any appropriate verification of that output, then the tool should be considered deterministic for the purposes of tool qualification. This results in a bounded problem.

(2) This interpretation of determinism should apply to all tools whose output may vary beyond the control of the user, but where that variation does not adversely affect the intended use

(for example, the functionality) of the output and the case for the correctness of the output is presented. However, this interpretation of determinism does not apply to tools that have an effect on the final executable image embedded into the airborne system. The generation of the final executable image should meet the restrictive interpretation of determinism.

(3) As an example, a tool may have a graphical user interface that allows the user to interact in a diagrammatic fashion. Underlying this tool are data tables that capture the intended meaning of those diagrams. Often, however, the output from these tools is at least partially driven by the physical ordering of the entries in these data tables, and the ordering of the data table entries is not controlled by the tool user. However, the correctness of the tool's output can be established. With the restrictive interpretation of determinism, this tool could not be qualified. However, with the expanded interpretation, qualification may be possible.

e. Guidelines for qualifying combined development and verification tools:

(1) This section applies only to tools that provide combined development and verification functions where the output of both the development and the verification functions are used to eliminate, reduce, or automate processes of RTCA/DO-178B. Combined tools that are used to eliminate, reduce, or automate only development objective(s) or only verification objective(s) should be qualified as such, irrespective of the other capabilities present in that tool.

(2) Qualification of combined tools (when both the development and verification functions are being used to meet or replace objectives of RTCA/DO-178B) should be performed to the guidance equivalent to the airborne software level ***unless protection/partitioning between the two functions can be demonstrated.*** Acceptable evidence of this protection/partitioning would be to show that the output of one function of the tool has no effect on the output of the other function of the tool (that is, the tool capabilities are functionally isolated).

(3) When protection/partitioning between the development and verification functions is shown, the protected/partitioned functions may be qualified as if they were separate development and verification tools (that is, the verification functions may be qualified to the criteria for verification tools).

f. Guidelines on configuration management of qualified tools: To receive credit (that is, to meet or replace RTCA/DO-178B objectives) for the use of qualified tools, those tools should be kept under configuration management. Not all requirements for configuration management of tools are in RTCA/DO-178B, Section 12.2. Section 12.2.3b of RTCA/DO-178B specifies the control categories for development and verification tool qualification data (see also Section 7.2.9b of RTCA/DO-178B). The control category for development tools qualification data should be the same as that required for airborne software of the same level (that is, the "CC1" and "CC2" criteria in Annex A tables applies to development tool qualification data). Verification tool qualification data, on the other hand, may be categorized as control category #2.

g. Guidelines on verifying changes to previously qualified tools: A software change impact analysis should be conducted on all changes to tools previously qualified. The analysis

should be thorough enough to assess the impact of the tool change on the product, as well as other tools under the influence of the change. A regression analysis may form part of the change impact analysis.

h. Guidelines on DER approval of tool qualification data: If the certification authority has delegated compliance findings for tool qualification data, DERs may approve the tool qualification data that comply with the guidance of RTCA/DO-178B, Section 12.2. However, the certification authority should retain approval of alternative methods and the resultant data.

i. Guidelines for tools developed before AC 20-115B issuance: Software tools used on pre-RTCA/DO-178B projects may be qualified for use on projects where RTCA/DO-178B is the means of compliance, if they meet the guidelines of this chapter. As an alternative, service history may be considered for such tools (see Section 4.11 of RTCA/DO-248B for more information on qualification of tools using service history).

Chapter 10. Approval of Software Changes in Legacy Systems Using RTCA/DO-178B

10-1. General. Many airborne systems were approved using RTCA/DO-178 or RTCA/DO-178A. These systems are referred to as legacy systems. Since the issuance of AC 20-115, many manufacturers are striving to use RTCA/DO-178B on their legacy systems. There are several items to keep in mind when addressing the use of RTCA/DO-178B on legacy systems:

a. RTCA/DO-178B is different from the two previous versions of RTCA/DO-178. The major change from the previous versions is the emphasis on a set of coordinated objectives rather than a collection of unrelated goal statements. There is also a change in emphasis from documentation to process objectives and the data needed to demonstrate compliance to those objectives. Software testing is the most visible difference between RTCA/DO-178B and previous versions. Therefore, software in legacy systems approved under a previous version may not have the same level of software testing assurance as that invoked by RTCA/DO-178B (that is, RTCA/DO-178B clarifies the scope and extent of software testing and test coverage). AC 20-115 effectively cancels all previous versions of RTCA/DO-178 as acceptable means of compliance in new projects. Therefore, changes/modifications to software accepted before the issuance of AC 20-115 should be evaluated using RTCA/DO-178B, when they are migrated to newer aircraft.

b. Another difference between RTCA/DO-178B and earlier versions is the classification of software levels and the need to perform a safety assessment to determine the software level. Previous versions only recognized three software levels, whereas RTCA/DO-178B recognizes five software levels. However, RTCA/DO-178B provides no guidance to show correspondence between these levels. This chapter will provide a method to establish that correspondence. Once the correspondence has been established, then RTCA/DO-178B may be applied to upgrade from a lower level to a higher level.

c. Prior versions of RTCA/DO-178 do not address the qualification of software development and verification tools. In many cases, tools are involved in changing legacy systems. Therefore, modification projects for legacy systems are faced with the issue of how to address tools used and not evaluated or qualified as part of the original certification approval. The subject of tool qualification is specifically addressed in chapter 9 of this order; paragraph 9-6i addresses pre-RTCA/DO-178B tools.

d. After reviewing field experience with numerous changes, a procedure was developed to provide a more consistent approach to address changes to the software of legacy systems. The approach described in this chapter attempts to take advantage of previous system approvals while ensuring that software changes are properly implemented and satisfy current FAA regulations and guidance.

Note: If the system contains multiple levels of software applications in protected partitions, the procedure should be applied to each of the partitioned applications affected by the change(s).

10-2. Discussion.

a. If the software level of the legacy system cannot be shown to be equivalent or better than that required by the product installation being considered, then the software should be upgraded per RTCA/DO-178B, Section 12.1.4, "Upgrading a Development Baseline." This may necessitate a complete reevaluation to demonstrate assurance to the appropriate objectives of RTCA/DO-178B. Determining equivalence is covered in paragraph 10-3; however, application of RTCA/DO-178B, Section 12.1.4 is not covered further.

b. There are four variables that can affect the actions needed in response to changes to software in legacy systems:

(1) The certification basis for the original product or installation of the legacy system containing the legacy software (that is, the regulations, the RTCA/DO-178 version, and software level applied to the original approval);

(2) Whether RTCA/DO-178B or a previous version is the accepted means of compliance for software for the product or installation under consideration (and if the software level is the same as or equivalent to the software level for the original certification);

(3) Whether the software is being modified or is unchanged (and how many other times it has been changed since the original certification, and the reason for those changes); and

(4) Whether the software and the legacy system are being installed on the same or a different aircraft or engine.

c. Assuming that the software levels can be shown to be equivalent, the majority of legacy system issues of concern can be categorized into the following groups:

(1) Legacy systems software is not modified and is reinstalled on the original aircraft (see paragraph 10-3b of this chapter).

(2) Legacy systems software is not modified but is installed on a different aircraft or engine where RTCA/DO-178B is not adopted as the means of compliance for software (see paragraph 10-3b of this chapter).

(3) Legacy systems software is modified and is reinstalled on the original aircraft or engines (see paragraph 10-3c of this chapter).

(4) Legacy systems software is modified and is installed on a different aircraft or engine where RTCA/DO-178B is not adopted as the means of compliance for software (see paragraph 10-3c of this chapter).

(5) Legacy systems software is modified and is installed on a different aircraft or engine where RTCA/DO-178B is adopted as the means of compliance for software (see paragraph 10-3d of this chapter).

(6) Legacy systems software is not modified but is installed on a different aircraft or engine where RTCA/DO-178B is adopted as the means of compliance for software (see paragraph 10-3e of this chapter).

d. Legacy systems, by definition, already have a recognized approval for installation or manufacturing through the TC, STC, ATC, ASTC, PMA, Production Certificate (PC), or TSO authorization processes. If there are no changes to the software of these systems, then the original approval of the software may still be valid, assuming an equivalence to the needed software level for the current installation can be ascertained (further discussed in paragraph 10-3 of this chapter) and the similarity of the system's use to the original approval is maintained. Before installation in an aircraft or engine, there should be an assessment that the legacy system will not be used in a significantly different manner than it was for the original installation approval.

e. The guidelines in this chapter may not be applicable to all TSO projects. AIR-100 Policy Memo, "Technical Standard Order (TSO), Software Approval Criteria," dated August 10, 1994, provides FAA policy regarding application of RTCA/DO-178B to TSO projects. Paragraphs b and c of the memo are particularly relevant to this chapter.

(1) Paragraph b states: "For TSOs that specify software guidelines, the ACO should conduct its review in accordance with those guidelines."

(2) Paragraph c states: "For TSOs that do not specify any software guidelines, the ACO should verify that the applicant's software development process and procedures meet the objectives of RTCA/DO-178B."

(3) Therefore, the guidelines in this chapter are applicable for TSOs that require RTCA/DO-178B (either in the TSO itself or because of the TC/STC/ATC/ASTC application) or that specify no software guidelines.

f. Systems with minor changes should be handled as changes under the original approval basis (that is, RTCA/DO-178B does not need to be applied to the changes). Examples of software changes that might be classified as minor include:

- Gain changes where the new gain is within a band of gain settings originally tested and approved,

- Changes to maintenance information formatting,

- Adding an output interface, or

- Changing data in a personality module that is within the set of options previously verified and approved.

(1) The certification authority and DER should be able to readily establish that these changes have been performed correctly under the original certification basis and software guidance. The normal data submittals appropriate to the revision of RTCA/DO-178 used for the original certification will still need to be evaluated to ensure that the changes are implemented correctly. If this cannot be done, then this is not a minor change.

(2) The determination of whether a change is minor cannot be made by considerations, such as metrics or a count of lines of code. Therefore, this determination will be based upon the software change impact analysis process outlined in chapter 11 of this order.

> **Note:** This process of allowing minor changes should not be followed, if the system is being used differently than it was for the original or subsequent installation approvals, or if the system has experienced in-service difficulties.

g. When changes are made to legacy systems beyond the minor changes, assurance that the changes have been correctly implemented and verified will be required. The following items should be considered:

(1) Earlier versions of RTCA/DO-178 do not contain well-defined acceptance criteria for several objectives and guidelines. One example is in the area of testing. RTCA/DO-178B guidelines indicate that testing be of sufficient rigor to provide specific structural coverage criteria and provide specific criteria for that rigor, whereas RTCA/DO-178A only indicates that testing exercise the logic and computations but does not specify any criteria for how extensively the structure should be exercised.

(2) Some newer technologies and tool qualification are not even addressed in the earlier versions of RTCA/DO-178. In all cases where ambiguities exist, use RTCA/DO-178B to provide a more exact interpretation.

(3) To be consistent with prior approvals, use RTCA/DO-178B to evaluate the processes used to make the change, the changed software components, and those components affected by the software changes, using the guidelines of chapter 11 of this order and Sections 12.1.1 through 12.1.6 of RTCA/DO-178B. Affected components should be identified by performing a change impact analysis of the software changes and identifying impacts on other components, interfaces, timing, and memory (for example, control coupling analysis, data coupling analysis, timing analysis, and memory usage analysis). These analyses should also identify the level and extent of regression testing needed to verify the change.

(4) The unaffected portions of the software already have an approval basis and could be accepted in accordance with paragraph 10-2d of this chapter. (Note that the unaffected portion is the software that neither changed nor was affected by the change as determined by control flow, data flow, memory usage, or timing analyses. The change impact analysis is used to determine the affected and unaffected portions.) In most cases, the risk of latent errors remaining in the software may be further mitigated by considering the benefit of service experience with the prior approval. RTCA/DO-178B, Section 12.3.5, "Product Service History," contains criteria that should be satisfied to allow the use of service experience. By virtue of the previous approval of the software, it may be assumed as already meeting many of the provisions of RTCA/DO-178B, Section 12.3.5. Little or no additional data may be needed from the applicant regarding service experience under Section 12.3.5, if the applicant has sufficient relevant service history data and no in-service problems with the system.

> **Note:** The note in Section 12.3.5g of RTCA/DO-178B does imply that additional data may be required to verify system safety objectives for software components and should be appropriately considered.

(5) Some TSOs require that DO-178[] and the appropriate level be specified on the nameplate. If a major change has been approved to RTCA/DO-178B and a majority of the software complies with RTCA/DO-178B, the nameplate may be marked with DO-178B and the appropriate software level.

(6) Once a DO-178B compliant change process is in place to address a major software change, that process should be applied to all subsequent changes to that software.

10-3. Procedures. For any project involving changes to a legacy system or a different installation for a legacy system, the certification authority and/or DER should follow the procedures listed in this paragraph.

a. The certification authority and/or DER should establish that there is equivalence between the legacy system's software level(s) and the proposed installation's software level using figure 10-1 below. Figure 10-1 illustrates the equivalence between DO-178/DO-178A and RTCA/DO-178B. Figure 10-1 is designed as a truth table asking the following question: "If the legacy system's software has a specific DO-178/DO-178A software level(s), can it be automatically considered "equivalent to" a certain RTCA/DO-178B level?" For example, if the legacy system's software is RTCA/DO-178A Level 2 software, it can be considered "equivalent to" Levels C, D, or E for an installation requiring RTCA/DO-178B.

(1) There are two entries in figure 10-1 that may require additional analysis before determining equivalency; these are shown by an "Analyze" in figure 10-1. There should be an agreement between the certification authority and applicant, when additional analysis is needed.

(2) If equivalency cannot be established by figure 10-1 (that is, a "NO" entry in the table), the provisions of RTCA/DO-178B, Section 12.1.4 should be applied to the software

application or partition to upgrade the software level. Procedures for applying Section 12.1.4 are not covered by this order. The remainder of this chapter assumes that equivalency has been established.

Figure 10-1. Software Level Equivalence

	Legacy System Software Level per RTCA/DO-178/DO-178A		
RTCA/DO-178B SW Level Required by the Installation	***Critical/Level 1***	***Essential/Level 2***	***Non-essential/Level 3***
A	YES/Analyze	NO	NO
B	YES	NO/Analyze	NO
C	YES	YES	NO
D	YES	YES	NO
E	YES	YES	YES

b. If the legacy system's software is unmodified and being reinstalled on the same aircraft or engine or a different aircraft or engine where RTCA/DO-178B is not required, then the original assurance process and associated data submittals may be accepted. This is only true if the system is being used in exactly the same way as originally certified, has no added functionality since the original or subsequent certification approvals, and has not experienced service difficulties (for example, Airworthiness Directives and Service Bulletins).

c. If the legacy system's software is modified and installed on the same aircraft or engine or on a different aircraft or engine where RTCA/DO-178B is not adopted as the means of demonstrating compliance for software, then either the compliance means of the original installation or the compliance means of the original legacy system may be used, providing the one with the latest revision is used. Again, this is only true if the system is being used in exactly the same way as originally certified, has no added functionality since the original certification, and has not experienced in-service difficulties. A change impact analysis as defined in chapter 11 of this order should be conducted to evaluate the software modifications and to apply appropriate regression testing.

d. If the legacy system software is modified and installed on a different aircraft or engine where RTCA/DO-178B is adopted as the means of demonstrating compliance, determine if the change is a minor change (per paragraph 10-2f of this chapter and the guidelines of chapter 11). Any changes determined to be minor changes may be handled the same as the not modified case discussed in paragraph 10-3b of this chapter. The determination of whether a change is a minor change is at the discretion of the certification authority and/or DER (if authorized), using the guidelines of chapter 11 of this order. Some representative, but not exhaustive, examples of minor changes are provided in paragraph 10-2f of this chapter.

(1) If the change is not a minor change, all changes to the software and all components affected by the change should be assured using RTCA/DO-178B (per paragraph 10-2g of this chapter). The change impact analysis is the normal means of determining affected

components. A description of change impact analysis is included in chapter 11. However, the project plans and processes and the change activities and evidences should be shown to meet the objectives of RTCA/DO-178B. For example, if the original software was not evaluated using the structural coverage criteria in RTCA/DO-178B, Section 6 and Annex A, then RTCA/DO-178B verification activities specified for the software level of the changed software will have to be completed and the coverage objectives satisfied.

(2) Additional affected, but unchanged, components may not have to be evaluated for internal structural coverage but should satisfy the objectives for data coupling and control coupling coverage (such as verify no changes to component interfaces with other components using integration testing), as well as requirements-based test coverage for those affected functions. Once this process is complete, the applicant may be allowed to claim that their legacy system software application or partition complies with RTCA/DO-178B, at the certification authority's discretion, depending on the significance of the modifications and evidence produced.

e. If the legacy system software is not modified but is installed on a different aircraft or engine (that is, different type certificate) where RTCA/DO-178B is adopted as the means of demonstrating assurance, then there should not be a separate compliance finding for the software. The original approval may serve as the installation approval of the software, unless the operational use of the system is expected to be significantly different (for example, an air data computer installed on piston-powered general aviation aircraft flying below 14,500 feet is now installed on a corporate jet flying at 50,000 feet). When the operational use is significantly different than the original or subsequent installation approvals, an assurance to RTCA/DO-178B guidance should be performed. The determination of the significance in change of the operational use is at the discretion of the certification authority and/or DER (if authorized).

f. All changes to software in legacy systems and the process used to approve those changes should be documented in the PSAC, SCI, and/or the SAS, as appropriate for the specific project. If service history is claimed for the legacy system, those data should be summarized in the SAS as well.

g. If any future changes are proposed, they should be addressed by using the criteria specified in this chapter.

This Page Intentionally Left Blank

Chapter 11. Oversight of Software Change Impact Analyses Used to Classify Software Changes as Major or Minor

11-1. General.

a. RTCA/DO-178B, Section 12.1.1, identifies analysis activities to be performed for proposed software changes. RTCA/DO-178B also states that reverification should be accomplished on all software changes and areas affected by those changes.

b. This chapter provides a standardized process to determine the impact of software changes on airborne systems, to assure that safety is not adversely impacted. This chapter also focuses on the change impact analysis to determine the extent of certification authority involvement in the review of changes, and to determine the significance of the software changes to the system.

c. The change impact analysis may be used by an applicant to provide justification for the classification of a change as it relates to 14 CFR §§ 21.93, 21.115, and 21.611. This chapter does not contain examples of minor or major changes, but it does offer guidelines for analyzing the impact of software changes. Changes analyzed as minor (using the guidelines of this order) for products previously approved under the TSO authorization process should be documented and verified by the applicant, but require no further oversight by the certification authority (per 14 CFR part 21). Likewise, changes analyzed as minor for products previously approved under the TC, STC, ATC, or ASTC process should be documented and verified by the applicant and may be implemented for the software applications without further oversight by the certification authority or DER (if authorized) per 14 CFR part 21. However, the substantiation and description of the change(s) should still be submitted to the certification authority in accordance with the regulations and delegation agreements.

11-2. Discussion.

a. The applicant should identify the software changes to be incorporated in the product and perform a change impact analysis. The change impact analysis should follow a defined process to determine the potential impact of the change on continued operational safety of the aircraft. For TSO authorized equipment, the analysis should identify the intended target aircraft environment that forms the basis for the analysis. This analysis also provides a basis for determining the extent of certification authority involvement. The following items should be addressed by the change impact analysis, as applicable:

(1) Traceability analysis identifies areas that could be affected by the software change. This includes the analysis of affected requirements, design, architecture, code, testing and analyses, as described below:

(a) ***Requirements and design analysis*** identifies the software requirements, software architecture, and safety-related software requirements impacted by the change.

Additionally, the analysis identifies any additional features and/or functions being implemented in the system, assures that added functions are appropriately verified, and assures that the added functions do not adversely impact existing functions.

(b) ***Code analysis*** identifies the software components and interfaces impacted by the change.

(c) ***Test procedures and cases analysis*** identifies specific test procedures and cases that will need to be reexecuted to verify the changes, identifies and develops new or modified test procedures and cases (for added functionality or previously deficient testing), and assures that there are no adverse effects as a result of the changes. The absence of adverse effects may be verified by conducting regression testing at the appropriate hierarchical levels (such as aircraft flight tests, aircraft ground tests, laboratory system integration tests, simulator tests, bench tests, hardware/software integration tests, software integration tests, and module tests), as appropriate for the software level(s) of the changed software.

(2) Memory margin analysis assures that memory allocation requirements and acceptable margins are maintained.

(3) Timing margin analysis assures that the timing requirements, central processing unit task scheduling requirements, system resource contention characteristics, interface timing requirements, and acceptable timing margins are maintained.

(4) Data flow analysis identifies changes to data flow and coupling between components and assures that there are no adverse impacts.

(5) Control flow analysis identifies changes to the control flow and coupling of components and assures that there are no adverse impacts.

(6) Input/output analysis assures that the change(s) have not adversely impacted the input and output (including bus loading, memory access, and hardware input and output device interfaces) requirements of the product.

(7) Development environment and process analyses identify any change(s), which may adversely impact the software application or product (for example, compiler options or versions and optimization change; linker, assembler, and loader instructions or options change; or software tool change).

(8) Operational characteristics analysis evaluates that changes (such as changes to gains, filters, limits, data validation, interrupt and exception handling, and fault mitigation) do not result in adverse effects.

(9) Certification maintenance requirements (CMR) analysis determines whether new or changed CMRs are necessitated by the software change.

(10) Partitioning analysis assures that the changes do not impact any protective mechanisms incorporated in the design.

Note: The above list is not all-inclusive and depends on the product for which the modification is being made.

b. The change impact analysis should determine whether the change could adversely affect safe operation of the system or product. The following are examples of areas that could have an adverse impact on safety or operation:

(1) Safety-related information is changed. For example:

(a) Previous hazards, identified by the system safety assessment, are changed.

(b) Failure condition categories, identified by the system safety assessment, are changed.

(c) Software levels are changed, particularly if the new software level is higher than the previous level.

(d) Safety-related requirements, identified by the system safety assessment, are changed.

(e) Safety margins are reduced.

(2) Changes to operational or procedural characteristics of the aircraft that could adversely affect flight safety. For example:

(a) Aircraft operational or airworthiness characteristics are changed.

(b) Flight crew procedures are changed.

(c) Pilot workload is increased.

(d) Situational awareness, warnings, and alerts are changed.

(e) Displayed information to make flight decisions is changed.

(f) Assembly and installation requirements are changed.

(g) Equipment interchangeability and/or interoperability with other equipment is changed.

(h) CMRs are changed or added.

(3) New functions or features are added to the existing system functions that could adversely impact flight safety.

(4) Processors, interfaces, and other hardware components or the environment are changed in such a way that safety could be adversely affected (see RTCA/DO-178B, Section 12.1.3).

(5) Software life cycle data (requirements, code, and architecture) is significantly changed in such a way that it could adversely affect safety. For example:

(a) Changes to software requirements, design, architecture, and code components (especially those affecting safety-related functions, partitioning, redundancy or safety monitors).

(b) Changes to code (source, object, and executable object) components that perform a safety-related function or changes to a component providing input to a component, which performs a safety-related function. (For this order, a safety-related function is one that could potentially induce or allow a major, hazardous, or catastrophic failure condition to go undetected).

(c) Changes to characteristics of the development environment impacting the executable object code.

(d) Changes to memory allocation requirements so that memory margins are adversely impacted (for example, less than 5 percent margin remaining).

(e) Changes to timing requirements so that timing margins are adversely impacted (for example, margins are unpredictable or less than 10 percent margin remains).

(f) Changes to input/output requirements (such as bus loading) so that input or output performance is adversely impacted (for example, less than 5 percent margin remains).

(g) Data and control coupling characteristics are adversely impacted (for example, to the extent that more than 50 percent of the coverage analysis must be redone).

(h) Changes to interface characteristics.

c. Additionally, the following items should be identified in the change impact analysis:

(1) Updates needed to assure that the software change(s) is incorporated in the appropriate software life cycle data, including requirements, design, architecture, source and object code, and traceability.

(2) Verification activities needed to verify the changes and that there are no adverse effects on the system. The change impact analysis should cover how changes that could adversely affect safe operation of the system or aircraft will be verified, so the changed and

unchanged software will continue to satisfy their requirements for safe operation. These verification activities may include reviews, analyses, regression testing, requirements-based testing, flight testing, and so on, including reevaluation of existing analyses, reexecution of existing tests, and new test procedures and cases (for added functionality or previously deficient testing).

11-3. Procedures. Each project involving software changes has different needs. This paragraph outlines procedures for the certification authority or DER (if authorized) to consider with the applicant when addressing software changes.

a. The applicant may define and follow a procedure for classifying software changes as major or minor and should seek certification authority review, feedback, and approval for that procedure. As a minimum, any such procedure should address the following before being implemented:

(1) The applicant's process for using the change impact analysis (see paragraph 11-2 of this chapter) to justify a minor or major change classification and the criteria used by the applicant to make the change classification.

> **Note 1:** The extensiveness and formality of the change impact analysis will vary by complexity, criticality, and extensiveness of the change. The change impact analysis may be in-depth for complex, highly critical systems but may be briefer and less rigorous for less complex or less safety critical systems or less extensive changes.
>
> **Note 2:** The applicant's documentation should address the categorization of the change as minor or major, per the appropriate regulations (for example, 14 CFR §§ 21.93, 21.115, and/or 21.611), to obtain FAA agreement on the change classification.

(2) The applicant's process to review and approve the change classification (such as DER review and approval).

(3) The process to be followed for a minor change determination (see paragraph 11-3c of this chapter).

(4) The process to be followed for a major change determination (see paragraph 11-3d of this chapter).

(5) The process for informing the FAA of all proposed software changes and their proposed classifications.

(6) The process for obtaining FAA concurrence with the proposed classifications.

Note: Once FAA approval of the software change classification procedure has been granted, the applicant should follow the procedure for all proposed software changes. Deviations from the approved process require ACO concurrence.

b. If the applicant does not have an FAA-approved software change classification procedure, the applicant should inform the FAA and/or DER that a software change is being planned. In these cases, the applicant should perform the following activities:

(1) Perform a change impact analysis, using paragraph 11-2 of this chapter.

(2) Propose a major or minor classification for the change (based on the change impact analysis and safety implications as stated in paragraph 11-2 of this chapter) and seek FAA feedback and concurrence on the classification.

(3) Support any proposed minor classification with rationale about the absence of safety impact and/or the limited scope of the change, and the proposed method of verifying the change. After the FAA has agreed to the applicant's data and rationale, the applicant may proceed without further FAA oversight for minor changes (see paragraph 11-3c of this chapter).

(4) Submit the appropriate documentation to the FAA for major changes (see paragraph 11-3d of this chapter).

c. For minor changes, the FAA oversight of the development process should involve approval and periodic review of the applicant's change impact analysis process and associated criteria for making a major/minor determination with respect to the relevant regulations. Once the change strategy and the change itself have been performed, the strategy and change impact analysis should be documented in the SAS. New, modified, and reused software life cycle data should also be identified in the SCI. For minor changes, submittals of the SAS and SCI to the cognizant ACO should be per agreement with the ACO.

Note 1: When applicable, DERs should be involved in the change classification procedure and oversight of the company's adherence to that procedure.

Note 2: Equipment containing changes classified by the manufacturer as minor but not yet concurred with by the certification authority or DER (when authorized) should not be installed on flight aircraft until the certification authority concurs with the classification.

d. For major changes, the certification authority and/or DER (if authorized) should review the applicant's PSAC or other summary of change impact analysis data and the applicant's proposed strategy for addressing the change issues. Once the change strategy and the change itself are completed, the certification authority and/or DER (if authorized) should ensure that the strategy and change impact analysis results are documented in the SAS. New, modified,

and reused software life cycle data should also be identified in the SCI and submitted to the certification authority and/or DER (if authorized to approve major changes).

> **Note:** In many cases, a change process may already be in place to address major, minor, significant, insignificant changes. The applicant's change impact analysis activities (in accordance with this order) should fit within the applicant's already existing framework to avoid unnecessary or inappropriate activities.

This Page Intentionally Left Blank

Chapter 12. Approving Reused Software Life Cycle Data

12-1. General. This chapter provides guidelines for determining if software life cycle data, produced and approved for one certification project, can be approved on a follow-on certification project. Approval for reuse could minimize the amount of rework while maintaining an equivalent level of design assurance.

12-2. Software Suitable for Reuse.

a. If properly planned and packaged, software life cycle data can be reused from one project to the next, with minimal rework. For example, the software plans, requirements, design, and other software life cycle data (as documented in a Software Configuration Index) for a Global Positioning System (GPS) may originally be approved on GPS #1 (the original certification project) and reused on GPS #2 (the subsequent certification project). Sample items suitable for reuse include:

(1) Software plans and standards. These include software undergoing non-substantive changes, such as:

- Program name,
- Name change due to consolidations or mergers, and
- Configuration changes for reasons other than design changes (for example, document format change, drawing modifications, or documentation system changes).

(2) Tool qualification data. The FAA can approve reuse, if the tool is used exactly as specified in the qualification approval as part of the original certification, and the applicant has access to the tool qualification data. This is true even if some of the features were qualified but not used during the original certification. The applicant should ensure that the same version of the tools is being used as that supported by the qualification data. The FAA will not approve reuse if the applicant uses additional or different tool functionality than was previously qualified.

(3) Software libraries. The FAA can approve library sets in the original certification project if the library set is used identically (that is, same library functions are used the same way).

(4) Software requirements, design, code, verification procedures, and verification results. The FAA may approve these for reuse after the applicant makes a thorough change impact analysis. This is to confirm that the requirements, design, code, procedures, and so forth are unaffected and unchanged from the previous certification effort.

(5) Configuration items. These may be approved for reuse in their entirety, if the certification authority and DERs use paragraphs 12-3 through 12-5 of this chapter to make the determination, and the configuration of the software life cycle data has not changed.

Configuration item requirements verified at a higher level (that is, system level) should be identified in the original configuration and reverified before reuse.

b. Projects not using RTCA/DO-178B may have additional considerations not documented in this chapter. Certification authorities should evaluate them on a case-by-case basis. The applicant should contact their local certification authority for guidance. The certification authority should coordinate with the CSTA for Aircraft Computer Software, the appropriate Directorate, and/or AIR-100, as necessary.

12-3. Safety Considerations. If the FAA finds software life cycle data acceptable for reuse, no further design approval is required. Figure 12-1 illustrates the considerations that govern whether the FAA will approve software reuse.

Figure 12-1. Reuse Approval Considerations

FAA may approve for reuse if:	1. There is no adverse effect on original system safety margins, **and** 2. There is no adverse effect on original operational capability UNLESS accompanied by a justifiable increase in safety.
FAA will NOT approve for reuse if the reuse:	1. Adversely affects safety, 2. Exceeds a pre-approved range of data or parameters, **or** 3. Exceeds an equipment performance characteristic.

12-4. Factors Affecting Reuse.

a. Any of the software life cycle data in Section 11, RTCA/DO-178B is suitable for reuse. To meet the guidelines in paragraph 12-5 of this chapter, the software life cycle data should be unchanged, and should apply to the project for which reuse is being considered.

b. In-service problems with previous applications can limit reuse. There may be Airworthiness Directives or a manufacturer's unresolved problem reports with the previously approved system. The applicant needs to analyze all open manufacturer's problem reports to ensure that the reusable portion of the new software is not affected. If the reusable portion of the new software is affected, changes to correct that software life cycle data should be made or the software should not be used.

c. Applicants should determine if the software data apply to the subsequent project's development by assessing the similarity of both the operational environment and the software development environment. They should:

(1) Assess the operational environment by evaluating the end-to-end performance requirements and the operational safety assessment.

(2) Refer to the Software Life Cycle Environment Configuration Index in Section 11.15, RCTA/DO-178B, when assessing the software development environment.

(3) Demonstrate that operational and development environments are the same, or demonstrated to produce identical results as the previous certification.

(4) Assess any outstanding problem reports.

12-5. Reuse Approval Guidelines.

a. The certification authority should ensure that the applicant has met the following guidelines before granting certification credit for reused software life cycle data:

(1) The software life cycle data have not changed since its previous approval.

(2) The software level of the software application(s) is equal to (or less than) the software level of the original certification effort.

(3) The range and data type of inputs to the configuration item are equivalent to its approved predecessor.

(4) The configuration item is embedded on the same target computer and is used the same way operationally as the original certification project.

(5) Equivalent software/hardware integration testing and system testing were conducted on the same target computer and system as in the original certification project.

(6) The applicant followed the safety considerations and reuse factors in paragraphs 12-3 and 12-4 of this chapter.

(7) The software life cycle data and the rationale for reuse of each item are documented in the "Additional Considerations" portion of the PSAC. The applicant's PSAC should include method of use, integration, and documentation for the reused configuration item. The PSAC should be submitted as early as possible in the development program. The applicant should also document all references to the project previously certified and the project number, as applicable, in the PSAC.

b. The certification authority responsible for the subsequent certification should review the PSAC and notify the applicant whether the proposal is acceptable or not (with appropriate rationale).

This Page Intentionally Left Blank

Chapter 13. Properly Overseeing Suppliers

13-1. When To Apply This Chapter. This policy applies when an applicant uses suppliers and sub-tier suppliers to perform system and software development, verification, and certification activities. The degree to which you use this policy may depend on the size and complexity of a particular certification project. Because it's impractical to cover all situations or conditions that may arise, supplement this policy with good judgment in handling the situation or condition. Confer with FAA system and software specialists as required.

13-2. Contemporary Issues.

a. Many TC/STC/TSOA applicants have shifted system and software development, verification, and certification activities onto their aircraft system suppliers and sub-tier suppliers. In the past, these suppliers participated in compliance activities only at their respective system, subsystem, or component levels. With airborne systems becoming increasingly more complex and integrated, and suppliers and sub-tier suppliers accepting these new responsibilities, we are concerned that their lack of expertise could result in incomplete or deficient certification activities.

b. Each responsibility that the applicant delegates to a supplier creates an interface with that supplier that needs to be validated and verified to ensure that the transition from the supplier's processes to the applicant's processes (or vice-versa) is accomplished correctly and accurately. Lack of proper validation and verification of life cycle data at the transition point has resulted in issues with regard to requirements, problem reporting, changes, etc.

c. Some certification tasks and activities may be performed in a foreign country. We can review the bilateral agreement with that country to determine if the certification authority may be able to help us in making a determination of compliance to the applicable FAA regulations. We can't, however, request the certification authority of a country with which we do not have a bilateral agreement in place to assist us in making a determination of compliance to FAA regulations. We would consider it an undue burden on us if we were required to oversee compliance activities at foreign supplier facilities in non-bilateral countries (including conducting on-site reviews). You may contact the International Policy Office, AIR-40, for additional information regarding bilateral agreements.

d. Finally, retention of substantiating data, such as software life cycle data and other certification and compliance data, is a critical part of the certification process. When this data is retained by a foreign supplier, it may not be readily available to us. This may also affect the continued operational safety of the aircraft and its systems, especially with regard to in-service problems (service difficulties), problem resolution (service bulletins), and mandatory corrections (airworthiness directives).

13-3. Supplier Oversight Plans and Procedures

a. The applicant should create oversight plans and procedures that will ensure all suppliers and sub-tier suppliers will comply with all regulations, policy, guidance, agreements, and standards that apply to the certification program. The applicable publications include, but are not limited to:

(1) 14 CFR;

(2) ACs;

(3) FAA orders and notices;

(4) Issue papers;

(5) Special conditions;

(6) Applicant designee procedures, partnership for safety plans, memoranda of agreement;

(7) Applicant standards for system, hardware, and software development (including requirements, design, and coding standards);

(8) Applicant quality assurance plans, procedures, and processes;

(9) Applicant configuration management plans, procedures, and processes;

(10) System supplier standards, plans, procedures and processes; and

(11) Applicant process for software change impact analysis.

b. The applicant's planning documents, such as certification plans and PSACs, should describe how the applicant will have visibility into their suppliers' and sub-tier suppliers' activities. This includes commercial off-the-shelf software component suppliers and vendors. The applicant should submit these plans for your review and approval, preferably early in the program. The applicant should avoid making changes to the plans late in the program. If late changes are unavoidable, the applicant must allow adequate time for your review and consideration.

13-4. Supplier Oversight: Review the Applicant's Plans.

a. The applicant should address the following concerns in a supplier management plan or other suitable planning documents. As a project engineer, you review the plan(s) and see that the following areas are addressed to your satisfaction:

(1) Visibility into compliance with regulations, policy, plans, standards, and agreements. The plan should address how the applicant will ensure that all applicable regulations, policy, plans, standards, issue papers, partnership for safety plans, and memoranda of agreement are conveyed to, coordinated with, and complied with by prime and sub-tier suppliers.

(2) Integration management. The plan should address how the system components will be integrated, and who will be responsible for validating and verifying the software and the integrated system. The plan should address:

(a) How requirements will be implemented, managed, and validated; including safety requirements, derived requirements, and changes to requirements;

(b) How the design will be controlled and approved;

(c) How the integration test environment will be controlled;

(d) How the software build and release process will be controlled (reconcile any differences between the supplier's and the applicant's release strategies);

(e) What product assurance activities that support the certification requirements will be conducted and who will be conducting them; and

(f) The applicant's strategy for integrating and verifying the system, including requirements-based testing and structural coverage analysis.

(3) Designee tasks and responsibilities. The plan should identify who the designees are and what their responsibilities are, who the focal points are, and how their activities will be coordinated and communicated. It should identify who will approve or recommend approval of software life cycle data.

(4) Problem reporting and resolution. The plan should establish a system to track problem reports. It should describe how problems will be reported between the applicant and all levels of suppliers. The problem reporting system should ensure that problems are resolved, and that reports and the resulting changes are recorded in a configuration management system. The plan should describe how the designee(s) will oversee problem reporting.

(5) Integration verification activity. The plan should identify who will be responsible for ensuring that all integration verification activities between all levels of suppliers comply with applicable guidance. It should describe how the designee(s) will oversee the verification process.

(6) Configuration management. The plan should describe the procedures and tools to aid configuration management of all software life cycle data. It should describe how configuration control will be maintained across all sub-tier suppliers, including those in foreign locations, and how designees will oversee configuration management.

(7) Compliance substantiation and data retention. The plan should describe how the applicant will ensure that all supplier and sub-tier supplier compliance findings are substantiated and retained for the program. The plan should address, at minimum, the following certification data:

(a) Evidence that compliance has been demonstrated;

(b) Verification and validation data; and

(c) Software life cycle data.

b. The applicant's supplier management plan (or equivalent plans) should address the concern identified in paragraph 13-2.b. regarding the transition of life cycle data between the applicant's processes and the suppliers' processes. The plan should address the validation and verification of data with regard to all processes, including requirements management, problem reporting, use of standards, change impact, reviews, etc.

c. The plans should state that certification data will be retained at a facility in the United States, and that the data will be in English, since non-English certification data may create ambiguities when translated to English. Data located in a facility outside the United States may present an undue burden on us.

Chapter 14. Software Problem Reporting

14-1. When to Apply This Chapter. This policy applies when an applicant's suppliers and sub-tier suppliers will be responsible for managing problems detected during the development of aircraft systems implemented with software. This chapter also discusses your involvement with assessing unresolved problems before certification. The degree to which you use this policy may depend on the size and complexity of a particular certification project. Because it's impractical to cover all situations or conditions that may arise, supplement this policy with good judgment in handling the situation or condition. Confer with FAA system and software specialists as required.

14-2. Supplier Involvement in Problem Reporting.

a. The software development and verification phases of complex and highly integrated systems are likely to result in a large number of problem reports produced by the applicant and their suppliers. This brings about the following concerns:

(1) The applicant's suppliers and sub-tier suppliers may not have the expertise to determine whether problems with their component(s) will have safety, functional, or operational impacts on the aircraft or airborne system in which they are used;

(2) The applicant may not have adequate visibility into supplier and sub-tier supplier problem reporting processes; and

(3) There may be a large number of open problem reports, indicating a lack of software maturity and assurance at TIA or certification.

b. Due to these concerns, the applicant will need to actively participate in the oversight of problem reporting processes to ensure that problems are properly identified, reported, and resolved.

c. RTCA/DO-178B, sections 7.2.3 through 7.2.7 and Table 7-1, provide guidance on problem reporting and resolution. Additionally, section 11.20 (j) states that the Software Accomplishment Summary should contain a summary of problem reports unresolved at the time of certification, including a statement of functional limitations.

14-3. Oversight of Problem Reporting.

a. In order to ensure that software problems are consistently reported and resolved, and that software development assurance is accomplished before certification, the applicant should discuss in their Software Configuration Management Plan, or other appropriate planning documents, how they will oversee their supplier's and sub-tier supplier's software problem reporting process. As a project engineer, you review the plans and verify that they address the following to your satisfaction:

(1) The plans should describe each of the applicant's supplier's and sub-tier supplier's problem reporting processes that will ensure problems are reported, assessed, resolved, implemented, re-verified (regression testing and analysis), closed, and controlled. The plans should consider all problems related to software, databases, data items, and electronic files used in any systems and equipment installed on the aircraft.

(2) The plans should establish how problem reports will be categorized so that each problem report can be classified as follows:

(a) Categories should identify problems with a potential impact on safety, functionality, performance, operation, or design assurance;

(b) Categories should identify problems that should be resolved before certification, and problems that could be deferred beyond certification; and

(c) Each category should define the criteria for which deferring the problem is acceptable.

(3) The plans should describe how the applicant's suppliers and sub-tier suppliers will notify the applicant of any problems that could impact safety, performance, functional or operational characteristics, software assurance, or compliance.

(a) The supplier may enter such problems into their own problem reporting and tracking system, and then transfer them to the applicant's problem reporting system. If so, the plan needs to describe how this is accomplished. If the supplier's problem reporting system is not directly compatible with the applicant's system, the plan needs to describe a process for verifying the translation between problem reporting systems.

(b) The applicant may allow their suppliers and sub-tier suppliers to have access to the applicant's problem reporting system. Doing so may help the applicant ensure that they will properly receive and control their supplier's problem reports. If the applicant does allow the supplier to have access to their system, they should restrict who within the supplier's organization has such access in order to maintain proper configuration control, and these individuals should be trained on the proper use of the applicant's problem reporting system.

(c) The plans should describe any tools that the applicant's suppliers or sub-tier suppliers plan to use for the purpose of recording action items or observations for the applicant to review and approve prior to entering them into the applicant's problem reporting system.

(d) The plans should state that suppliers will have only one problem reporting system in order to assure that the applicant will have visibility into all problems and that no problems are hidden from the applicant.

(e) Any problems that may influence other applications, or that may have system-wide influence should be made visible to the appropriate disciplines.

(4) The plans should describe how flight test, human factors, systems, software, and other engineers of the appropriate disciplines will be involved in reviewing each supplier's and sub-tier supplier's problem report resolution process. They should also describe how these engineers will participate in problem report review boards and change control boards.

(5) The plans should establish the criteria that problem report review boards and change control boards will use in determining the acceptability of any open problem reports that the applicant will propose to defer beyond certification.

(a) These boards should carefully consider the potential impacts of any open problem reports on safety, functionality, and operation.

(b) Since a significant number of unresolved problem reports indicate that the software may not be fully mature and its assurance questionable, the applicant should describe a process for establishing an upper boundary or target limit on the number of problem reports allowed to be deferred until after type certification.

(c) The plan should establish a means of determining a time limit that unresolved problem reports deferred beyond certification will be resolved. This applies to problem reports generated by the applicant, suppliers, and sub-tier suppliers.

b. As a project engineer, you should be involved in certain decisions related to open problem reports prior to TIA and certification. You should:

(1) Review, as appropriate, any problem reports that are proposed for deferral beyond certification. This review may require FAA flight test, systems, and other specialists. You may need to ask for more information to make your assessment. If you have concerns that safety might be impacted, you can disallow the deferral of specific problem reports.

(2) If the applicant is using previously developed software, ensure that the applicant has reassessed any open problem reports for their potential impact on the aircraft or system baseline to be certified.

(3) Ensure that the applicant has considered the inter-relationships of multiple open problem reports and assessed whether any open problem report has become more critical when considered in conjunction with another related problem report.

(4) Ensure that the applicant has reviewed any open problem reports related to airworthiness directives, service bulletins, or operating limitations and other mandatory corrections or conditions. The applicant may need your help to determine which problems to resolve before certification.

(5) Review any open problem reports with potential safety or operational impact to determine if operational limitations and procedures are required before FAA test pilots participate in test flights. You may need to involve technical experts in making your determination.

(6) Ensure that the applicant has complied with DO-178B, section 11.20 (j).

Chapter 15. Assuring Airborne System Databases and Aeronautical Databases

15-1.When to Apply This Chapter. This policy applies when the applicant's airborne systems and equipment is utilizing aeronautical databases or airborne system databases. The degree to which you use this policy may depend on the size and complexity of a particular certification project. Because it's impractical to cover all situations or conditions that may arise, supplement this policy with good judgment in handling the situation or condition. Confer with FAA system and software specialists as required.

15-2. Databases and Their Design Assurance. There are three distinct types of databases used in airborne systems and equipment:

a. Aeronautical databases, which are used by an airborne system and whose development processes are typically approved using the guidance of RTCA/DO-200A, AC 20-153A, and Order 8110.55.

(1) Aeronautical databases should be demonstrated to comply with RTCA/DO-200A or other acceptable means. RTCA/DO-200A defines requirements and an acceptable means of compliance for participants processing aeronautical databases. If followed, it provides assurance that the production of aeronautical databases meets the integrity requirements for intended function, based on design assurance levels or software levels. It addresses specifics of the aeronautical data process, and assumes that participating organizations have an acceptable quality management system.

(2) AC 20-153A applies to navigation, terrain, obstacle, and airport map databases, and provides criteria for organizations to apply for a letter of acceptance (LOA) for their aeronautical data process. The LOA identifies organizations within the aeronautical data chain that demonstrate acceptable data processes, and formally documents that a supplier's databases are being produced according to RTCA/DO-200A.

(3) Order 8110.55 explains how you can evaluate and accept aeronautical data processes of a database supplier who complies with AC 20-153A and issue them an LOA.

b. Airborne system databases, which are used by an airborne system and approved as part of the type design of the aircraft or engine. These databases may influence paths executed through the executable object code, be used to activate or deactivate software components and functions, adapt the software computations to the aircraft configuration, or be used as computational data.

(1) Airborne system databases may consist of script files, interpretive languages, data structures, or configuration files (including registries, software options, operating program configuration, aircraft configuration modules, and option-selectable software).

(2) Assurance of these databases is typically achieved in the context of RTCA/DO-178B airborne system and equipment software processes.

c. Other applications and databases, which are not part of the type design of the aircraft or engine, and which are operationally approved by Flight Standards. This includes applications and databases defined as Type A and Type B in AC 120-76A, and electronic checklists addressed in AC 120-64. User-Modifiable Software is also in this category (refer to section 2.4 of DO-178B and chapter 7 of this order). These applications and databases have no design assurance requirements and therefore are not addressed in this chapter.

15-3. Assuring Aeronautical Databases. To ensure that the applicant and their airborne system suppliers have complied with all applicable regulations and FAA guidance for aeronautical databases, you should:

a. Ensure that the applicant has followed the guidance provided in AC 20-153A, or other acceptable means for aeronautical databases that comply with the requirements of RTCA/DO-200A. A current Type 2 LOA (refer to AC 20-153A) provides evidence that the aeronautical database complies with DO-200A in support of installation eligibility and operational authorization for use.

b. Ensure that any aeronautical databases meet the appropriate assurance level requirements using RTCA/DO-200A (Appendix B), AC 20-153A, or other acceptable means (refer to Order 8110.55).

15-4. Assuring Airborne System Databases. To ensure that the applicant and their airborne system suppliers have complied with all applicable regulations and FAA guidance for airborne system databases, you should:

a. Review the applicant's aircraft and system safety assessment(s) and verify that for each airborne system database:

(1) They have considered possible database errors and corruption for each system that will use each database;

(2) They have assigned appropriate software levels to each database (refer to AC xx.1309, AC 33.28, ARP 4754a, and ARP4761);

(3) They have based assigned database software levels on the worst-case potential hazard effect that errors or corruption could cause for the system and aircraft or engine; and

(4) You concur with the identified hazards and assigned software levels.

b. Ensure that each database is assured to the appropriate software level using RTCA/DO-178B or other acceptable means, and that they are verified in the context of the functional software, the system, and the overall aircraft use.

(1) A level of verification coverage appropriate for the database software level should be achieved. This may be achieved by a combination of requirements-based testing, data coupling analyses for data items that provide data only, and control coupling analyses for data items that influence software execution.

(2) Review the applicant's proposed verification coverage criteria for each database and either concur or provide rationale if you do not concur.

(3) Ensure that the applicant has applied robustness test conditions for databases, including those that influence software execution.

15-5. Actions Applicable to Aeronautical and Airborne System Databases.

a. Review any field-loadable software loading procedures for each database. Ensure that safeguards are established to detect database transmission and media errors, loading and content errors, mismatches between database part numbers and the aircraft systems or embedded software, and corruption of database contents or memory during use. Refer to chapters 5 and 6 of this order for more guidance on approving field-loadable software.

b. Ensure that maintenance instructions and appropriate limitations are provided for database updates if the contents of the database are valid for use only within a specified time.

c. Ensure that the applicant has provided a process for updating each database. The process should include a means for obtaining airworthiness approval and/or operational authorization for use, such as STC, minor modification (mod level change), system part number roll, or software part number roll, as appropriate. The process should address databases with their own part number assigned, as well as databases considered part of the operational software.

This Page Intentionally Left Blank

Chapter 16. Managing the Software Development or Verification Environment

16-1. When to Apply This Chapter. This policy applies when the applicant is using a software development or verification environment that may not be completely representative of the target computer. In this chapter, we show you how to ensure that the applicant establishes and maintains configuration control of the software development and verification environment, and implements a structured problem reporting system for the environment. The degree to which you use this policy may depend on the size and complexity of a particular certification project. Because it's impractical to cover all situations or conditions that may arise, supplement this policy with good judgment in handling the situation or condition. Confer with FAA system and software specialists as required.

16-2. How Representative is the Environment? RTCA/DO-178B requires that the verification test activities take place on the target computer, a target emulator, or a host computer simulator. Software development and verification teams typically utilize an environment designed specifically to emulate the target computer to satisfy this requirement. Because the environment may go through several iterations during software development and verification, it may not be clear how representative the environment is of the actual production hardware at any point in time in the verification process. Additionally, the environment may not be identical to the final production version of the hardware to be installed in the aircraft. Therefore, the applicant should establish and maintain configuration control of the environment, and implement a structured problem reporting system for the environment available to users of the environment.

16-3. Controlling the Development and Verification Environment. The applicant should address the following aspects in their Software Development Plan, Software Verification Plan, and Software Configuration Management Plan as applicable. The applicant should convey these aspects to all participating software suppliers, and ensure that they comply with them. As a project engineer, you review these plans and assess their adequacy.

a. The Software Development Plan and Software Verification Plan should include:

(1) A description of the software development or verification environment, and an explanation of the differences between it and the production version of the system hardware and software to be installed on the aircraft.

(2) An explanation of how the software development or verification environment will be used by system software suppliers and what RTCA/DO-178B objectives it will be used to show compliance with.

(3) An explanation of how the software development or verification environment will be used to show compliance with RTCA/DO-178B objectives that involve verification of the software executable object code. This should address the entire executable object code, not just individual functional software components. If development tools are being used in the integrated environment, then verification should also be performed in the integrated environment.

(4) A process for analyzing completed verification activities and assessing the need to repeat any of those activities after changes are made to the software development and verification environment. The process should ensure that all affected verification activities will be repeated, or ensure that a documented analysis is conducted showing why retesting is not required.

b. The Software Configuration Management Plan should include:

(1) A description of the configuration control system to be used for the software development and verification environment. The plan should identify the person who is responsible for administering this system.

(2) A problem reporting and assessing system for the software development and verification environment that is available to all users of the environment (refer to chapter 14 of this order).

Appendix A. Level of Involvement Worksheets

Appendix A contains three worksheets that may be used to help the certification authority or designee determine an appropriate level of involvement in software projects. The worksheets are provided as examples only and their use, individually or in combination, is not mandatory. Worksheet 1 indicates a level of involvement based on the software level of the project. Worksheet 2 allows for additional refinement of involvement based on more specific criteria. Worksheet 3 uses the total score result from Worksheet 2 to indicate a level of involvement.

Worksheet 1: Level of Involvement Based on Software Level

RTCA/DO-178B/C Software Level	Level of Involvement
D	LOW
C	LOW or MEDIUM
B	MEDIUM or HIGH
A	MEDIUM or HIGH

Worksheet 2: Level of Involvement Based on Other Relevant Project Criteria

	Criteria	Scale	MIN.		MAX.	Score
1.	**Applicant/Developer Software Certification Experience**					
1.1	Experience with civil aircraft or engine certification.	Scale: # projects:	0 0	5 3-5	10 6+	
1.2	Experience with RTCA/DO-178B/C.	Scale: # projects:	0 0	5 2-4	10 5+	
1.3	Experience with RTCA/DO-178 or RTCA/DO-178A.	Scale: # projects:	0 0	3 4-6	5 7+	
1.4	Experience with other software standards (other than RTCA/DO-178 [ ]).	Scale: # projects:	0 0	2 4-6	4 7+	
2.	**Applicant/Developer Demonstrated Software Development Capability**					
2.1	Ability to consistently produce RTCA/DO-178B/C software products.	Scale: Ability:	0 Low	5 Med	10 High	
2.2	Cooperation, openness, and resource commitments.	Scale: Ability:	0 Low	5 Med	10 High	
2.3	Ability to manage software development and sub-contractors.	Scale: Ability:	0 Low	5 Med	10 High	
2.4	Capability assessments (for example, Software Engineering Institute Capability Maturity Model, ISO 9001-3).	Scale: Ability:	0 Low	2 Med	4 High	
2.5	Development team average based on relevant software development experience.	Scale: Ability:	0 < 2 yrs	5 2-4 yrs	10 > 4 yrs	
3.	**Applicant/Developer Software Service History**					
3.1	Incidents of software-related problems (as a % of affected products).	Scale: Incidents:	0 > 25%	5 > 10%	10 None	
3.2	Company management's support of designees.	Scale: Quality:	0 Low	5 Med	10 High	
3.3	Company software quality assurance organization and configuration management process.	Scale: Quality:	0 Low	5 Med	10 High	
3.4	Company stability and commitment to safety.	Scale: Stability:	0 Low	3 Med	6 High	
3.5	Success of past company certification efforts.	Scale: Success:	0 None	3 > 50%	6 All	

	Criteria	Scale	MIN.		MAX.	Score
4.	**The Current System and Software Application**					
4.1	Complexity of the system architecture, functions, and interfaces.	Scale: Complex:	0 High	5 Med	10 Low	
4.2	Complexity and size of the software and safety features.	Scale: Complex:	0 High	5 Med	10 Low	
4.3	Novelty of design and use of new technology.	Scale: Newness:	0 Much	5 Some	10 None	
4.4	Software development and verification environment.	Scale: Environ:	0 None	3 Older	6 Modern	
4.5	Use of alternative methods or additional considerations.	Scale: Standard:	0 Much	3 Little	6 None	
5.	**Designee Capabilities**					
5.1	Experience of designee(s) with RTCA/DO-178B/C.	Scale: Projects:	0 < 5	5 5-10	10 > 10	
5.2	Designee authority, autonomy, and independence.	Scale: Autonomy:	0 None	5 Self-starter	10 Outgoing	
5.3	Designee cooperation, openness, and issue resolution effectiveness.	Scale: Effectiveness:	0 Non-Responsive	5 Responsive	10 Cooperative/Open	
5.4	Relevance of assigned designees' experience.	Scale: Related:	0 None	5 Somewhat	10 Exact	
5.5	Designees' current workload.	Scale: Workload:	0 High	5 Medium	10 Low	
5.6	Experience of designees with other software standards (other than RTCA/DO-178[]).	Scale: Projects:	0 < 5	3 5-10	5 > 10	

Total Score Result (TSR): _____

Worksheet 3: Level of Involvement Combining Results of Worksheet 2 with Software Level

Total Score Result (TSR)	**Software Level A**	**Software Level B**	**Software Level C**	**Software Level D**
TSR $\leq$ 80	HIGH	HIGH	MEDIUM	LOW
80 < TSR $\leq$ 130	HIGH	MEDIUM	MEDIUM	LOW
130 < TSR	MEDIUM	MEDIUM	LOW	LOW

Appendix B. Directive Feedback Information

Directive Feedback Information

Please submit any written comments or recommendation for improving this directive, or suggest new items or subjects to be added to it. Also, if you find an error, please tell us about it.

Subject: Order 8110.49_Change 2

To: Directive Management Officer, 9-AWA-AVS-AIR-DMO@faa.gov

(Please check all appropriate line items)

☐ An error (procedural or typographical) has been noted in paragraph _____ on page _____ .

☐ Recommend paragraph _____ on page _____ be changed as follows: *(attach separate sheet if necessary)*

☐ In a future change to this order, please include coverage on the following subject *(briefly describe what you want added):*

☐ Other comments:

☐ I would like to discuss the above. Please contact me.

Submitted by: _____ Date:_____

Telephone Number: _____ Routing Symbol: _____

FAA Form 1320-19 (10-98)