

Executive Summary

Towards Common Criteria Certification for DO-178B Compliant Airborne Software Systems¹

*Carol Taylor, Jim Alves-Foss, and Bob Rinker
Center for Secure and Dependable Systems
University of Idaho*

Introduction

This document summarizes the previous document, *Towards Common Criteria Certification for DO-178B Compliant Airborne Software Systems* presented in March, 2001 at the meeting between the University of Idaho, Rockwell Collins, and the NSA. Included are introductions to each of the reviewed documents, DO-178B, *Software Considerations in Airborne Systems and Equipment Certification* and the Common Criteria (CC), *Common Criteria for Information Technology Security Evaluation*. A high-level mapping between the Assurance Requirements Classes of the CC and the development processes of the DO-178B will be included. Additionally, a low-level mapping between the specific components of the CC Assurance Classes and the DO-178B requirement processes will be done. This will demonstrate where the two documents overlap and where requirements will have to be added from either the CC or the DO-178B in order to meet both DO-178B and CC requirements. The purpose of this document is to enable developers to map DO-178B compliance efforts into the CC.

Overview of DO-178

Requirements for software development are divided into several main processes within DO-178B. These processes include: Software Planning, Software Development, Software Verification, Software Configuration Management and Software Quality Assurance. Each of these processes will be summarized briefly to promote understanding of the DO-178B development requirements. Additionally, the five software levels relating to potential software failure in aircraft are briefly outlined as presented in DO-178B

Software Planning Process

¹ Not releasable to the Defense Technical Information Center per DoD directive 3200.12. This material is based upon work supported by the DOD under Contract. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the DoD.

The purpose of software planning is to define the means of producing software which will satisfy the system requirements and provide the confidence which is consistent with air worthiness requirements.

The software planning process objectives include:

- Definition of software life cycle processes and activities
- Software life cycle environment is selected
- Software development standards are defined
- Software plans are produced

Development of software plans are an important output of the planning process and define the means of satisfying the objectives of this document. Briefly, the software plans are described as follows:

Plan for Software Aspects of Certification - defines proposed development methods for the certification authority and defines means of compliance with DO-178B.

Software Development Plan – defines software life and development environment

Software Verification Plan – means by which software verification process objectives will be satisfied

Software Configuration Management Plan – means by which software configuration management objectives will be satisfied

Software Quality Assurance Plan - means by which software quality assurance objectives will be satisfied

Software Development Process

Under software, several sub-processes are defined each of which is part of the overall development process.

Software Requirements Process

Uses outputs of the system life cycle process to develop software high-level requirements including functional, performance, interface and safety-related requirements.

Software Design Process

High-level requirements are refined through the software design process to develop the software architecture and low-level requirements that can then be used to implement source code.

Software Coding Process

Source code is implemented from the software architecture and low-level requirements.

Integration Process

Consists of software integration and hardware/software integration

Software Verification Process

Verification is a technical assessment of the results of both the software development processes and the software verification processes. Verification is more than testing. Testing cannot show absence of errors but verification is typically a combination of reviews, analysis and test. The purpose of software verification is to detect and report errors that may have been introduced during the software development process. Software verification process objectives are satisfied through a combination of reviews, analysis and development of test cases and procedures. Execution of the test procedures provides a demonstration of compliance with the requirements.

Software Configuration Management Process

The objectives of the Software Configuration Management Process include:

- controlled configuration of the software throughout the software life cycle
- ensures consistency and repeatability of process activities
- ensures that secure archiving, recovery and control are maintained for configuration items

The Software Configuration Management process includes the activities of configuration identification, change control, baseline establishment, and archiving of the software product.

Software Quality Assurance Process

The Software Quality Assurance (SQA) assesses the software life cycle processes and their outputs to obtain assurance that the objectives are satisfied, that deficiencies are detected and resolved and that software product and life cycle data conform to certification requirements.

The SQA process objectives are to provide confidence that the software life cycle processes produce software that conforms to its requirements by assuring that these processes are performed in compliance with approved software plans and standards.

Software Levels in DO-178B

In addition to the specifications for software development, DO-178B describes requirements relating to five software levels. The software levels are correlated to the contribution of the software to potential failure conditions in aircraft. The five levels range from Level A, the most failure critical, through Level E, which has no effect on aircraft safety. The five levels along with the DO-178B compliance requirements are described in the following paragraphs.

Level A is the most critical failure level and a software failure here would result in a catastrophic failure condition for an aircraft.

Level B would cause or contribute to hazardous/severe major failure condition for an aircraft.

Level C would cause or contribute to major failure condition for an aircraft.

Level D would cause or contribute to minor failure condition for an aircraft.

Level E would have no effect on an aircraft.

The differences between the levels in regards to precautions against failure focus primarily on two issues:

1. Data required to prove compliance with development standards and software configuration management
2. Independent vs. non-independent assessment of compliance with DO-178B requirements².

The first difference between the software levels concerns the configuration management data required throughout the software development process. Configuration management practices are used throughout the software life cycle and differ between the software levels A, B, C, and D³ in the controls placed on the software life cycle data. There are two control processes under which data can be classified: Control Category 1 (CC1) and Control Category 2 (CC2). CC2 is a subset of CC1 requirements and is less stringent in terms of software life cycle data management. Features of these control processes include requirements for baselines, traceability, change control, change review, unauthorized changes protection, release and data retention among others.

Each software development process item is evaluated separately with regards to control category. Some items have higher requirements for data control (CC1) than others within the same Software Level. For example, under the Software Planning Process for Level A, definition of the software development processes requires that data satisfy the objectives of CC1, while coordination between the software plans only requires data to satisfy CC2.

The second area of difference between the software levels concerns independent assessment of the items within the major areas of DO-178B. The three levels of assessment in decreasing levels of rigor are:

² Annex A, of DO-178B summarizes each of the software life cycle processes and includes tables of each process by software level showing the requirements needed for the different levels.

³ Level E is “no effect” and is not further discussed in DO-178B

- Objective should be satisfied with independence.
- Objective should be satisfied
- Satisfaction of the objective is at applicant's discretion

At Level A, most of the items only require that the objective be satisfied not the most stringent of the assessments requirements. See Annex A – DO178B for a complete breakdown of these assessments by process item and software level.

Overview of the Common Criteria (CC)

The CC is designed to specifically address security functionality and assurance of that functionality. As such, it is much more prescriptive with regards to details regarding security functionality and less prescriptive about other issues compared to DO-178B.

The CC organizes assurance requirements into *classes* which identifies the general topic covered by that class. Each class then contains subtopics called *families*, which in turn have components that specify the individual requirements. The assurance classes that we have included in the comparison are:

ACM - Configuration Management
ADO – Delivery and Operation
ADV- Development
AGD – Guidance Documents
ALC – Life Cycle Support
ATE – Tests
AVA – Vulnerability Assessment

Security requirements for software and systems can vary depending on the purpose of the system. The CC thus offers different levels of assurance, termed Evaluation Assurance Levels (EALs). The EALs range from EAL1, the lowest level of assurance to EAL7, the highest security assurance level. A comparison of the different EALs is included.

Common Criteria Assurance Classes

ACM – Configuration Management

Configuration management is one method for establishing that the functional requirements and specification are implemented in the Target of Evaluation (TOE). Configuration management requires discipline and control in the processes of refinement and modification of the TOE. By requiring configuration management, methods are established to track changes and ensure that all changes are authorized.

ADO – Delivery and Operation

Delivery and operation provides requirements for correct delivery, installation, generation, and start-up of the TOE.

ADV – Development

The development class includes requirements for representing the security functionality at various levels of abstraction from the functional interface to the implementation. Requirements are also included for a correspondence mapping between the various representations. This class also includes requirements on the internal structure of the security functionality which covers aspects such as modularity, layering and minimization of complexity.

AGD – Guidance Documents

Guidance documents provides requirements for both user and administrator guidance documentation. These documents refer to written material that is intended to be used by those persons responsible for configuring, maintaining, and administering the TOE for maximum security.

ALC – Life Cycle Support

Life-cycle support requirements establish discipline and control in the TOE refinement during its development and maintenance. This class outlines how security analysis and production of evidence should be part of the development and maintenance activities.

ATE – Tests

Testing helps to establish that the TOE security functional requirements are met and provides assurance that the TOE satisfies the TOE security functional requirements. Emphasis for this class is on confirmation that the TOE security functions operate according to its specification. Both positive testing based on functional requirements and negative testing to check that undesirable behavior is absent is included. Penetration testing which identifies vulnerabilities in the design and implementation of the TOE security functions is addressed separately as an aspect of vulnerability assessment.

AVA – Vulnerability Assessment

This class includes requirements to address the existence of exploitable covert channels, deliberate misuse and the potential exploitable vulnerabilities introduced in the development and operation of the TOE.

Differences Between the Common Criteria Evaluation Assurance Levels

EALs are predefined assurance packages that define a consistent set of assurance requirements. Together the EALs form an ordered set that is the predefined assurance scale of the CC. The EALs are hierarchically ordered in that each EAL represents more assurance than all lower EALs. Substituting a higher assurance component from the same family or adding a component from other families achieves an increase in assurance.

The following sections describe the CC EALs and were extracted from Part 3, Section 6 of the CC documentation.

EAL1 – functionally tested

Provides a basic level of assurance by an analysis of the security functions using a functional and interface specification and guidance documentation to understand the security behavior.

EAL2 – structurally tested

Provides assurance by an analysis of the security functions, using a functional and interface specification, guidance documentation and the high-level design of the target of evaluation (TOE), to understand security behavior.

EAL2 represents an increase in security over EAL1 by requiring developer testing, a vulnerability analysis and independent testing based upon a more detailed TOE specification.

EAL3 – methodically tested and checked

Provides assurance by an analysis of the security functions, using a functional and interface specification, guidance documentation and the high-level design of the TOE, to understand security behavior. Independent testing of the TOE security functions supports analysis, evidence of developer testing based on the functional specification and high-level design, selective independent confirmation of developer test results, strength of function analysis and evidence of developer search for obvious vulnerabilities.

EAL3 is an increase in assurance over EAL2 by requiring more complete testing coverage of the security functions and mechanisms and/or procedures that provide some confirmation that the TOE will not be tampered with during development.

EAL4 – methodically designed, tested and reviewed

Provides assurance by an analysis of the security functions, using a functional and interface specification, guidance documentation, both high-level and low-level design of the TOE, and a subset of the implementation to understand security

behavior. Assurance is gained through an informal model of the TOE security policy.

EAL4 is an increase in assurance over *EAL3* because it requires more design description, a subset of the implementation and improved mechanisms that provide confidence that the TOE will not be tampered with during development or delivery.

EAL5 – semi-formally designed and tested

Provides assurance by an analysis of the security functions, using a functional and interface specification, guidance documentation both high-level and low-level designs of the TOE, and all of the implementation to understand security behavior. Additional assurance is gained through a formal model of the TOE security policy and a semiformal presentation of the functional specifications and high level design and a semiformal demonstration of the correspondence between them.

EAL5 offers increased assurance over *EAL4* in that a semiformal design description is required over the entire implementation, more structured architecture, covert channel analysis, and improved mechanisms and or procedures that provide confidence that the TOE will not be tampered with during development.

EAL6 – semiformal verified design and tested

Provides assurance by an analysis of the security functions, using a functional and interface specification, guidance documentation both high level and low level design of the TOE, and all of the implementation to understand security behavior and a structured presentation of the implementation. Assurance is gained through a formal model of the TOE security policy and a semiformal presentation of the functional specifications and high level design and a semiformal demonstration of the correspondence between them. A modular and layered TOE design is also required at this level.

EAL6 offers increased assurance over *EAL5* by requiring semiformal design descriptions, a structured representation of the implementation which is more analyzable, system covert channel identification and improved configuration management and development environment controls

EAL7 – formally verified design and tested

Provides assurance by an analysis of the security functions, using a functional and interface specification, guidance documentation both high-level and low-level design of the TOE, and all of the implementation to understand security behavior and a structured presentation of the implementation. Assurance is gained through

a formal model of the TOE security policy and a formal presentation of the functional specifications and high level design, semiformal presentation of the low-level design and a formal and semiformal demonstration of the correspondence between them. Modular, layered and simple design is required.

EAL7 offers more assurance than *EAL6* by requiring comprehensive analysis using formal representations and formal correspondence plus comprehensive testing

High-Level Comparison Between DO-178B Processes and CC Classes

There is a general correspondence between many of the DO-178B Processes and the CC Classes. This next section relates where these two documents have sections that offer comparable requirements by subject.

<u>CC Assurance Class</u>	<u>DO-178B Area</u>
ACM Configuration Management	Software Configuration Management
ADO Deliver and Operation	<i>(No Correspondence)</i>
ADV Development	Software Development Process
AGD Guidance Documents	<i>(No Correspondence)</i>
ALC Life Cycle Support	Software Planning Process
ATE Tests	Software Verification Process
AVE Vulnerability Assessment	<i>(No Correspondence)</i>
<i>(No Correspondence)</i>	Software Quality Assurance

Low-Level Comparison Between DO-178B Processes and Common Criteria Classes

At a more detailed level, comparisons can be made between individual components within the CC classes and the DO-178B areas. Also, sections in DO-178B that describe the data needed for these processes are included in the cross reference to the CC class

components. The following table, Table 1, details this cross-reference between CC class components and DO-178B areas.

Table 1. Cross-reference between CC class components and DO-178B sections

ACM – Configuration Management ACM_Aut CM Automation ACM_CAP Advanced support ACM_SCP Development Tools	<table border="1"> <thead> <tr> <th colspan="2">Software Configuration Management</th> </tr> <tr> <th>Activities</th> <th>Data Control Processes</th> </tr> </thead> <tbody> <tr> <td>X</td> <td></td> </tr> <tr> <td>X</td> <td></td> </tr> <tr> <td>X</td> <td></td> </tr> </tbody> </table>	Software Configuration Management		Activities	Data Control Processes	X		X		X		Life Cycle Data Sect. 11.4 Sect. 11.18 Sect. 11.4, 11.18																									
Software Configuration Management																																					
Activities	Data Control Processes																																				
X																																					
X																																					
X																																					
ADO – Delivery and Operation ADO_Del Prevention Modification ADO_IGS Installation and Start-up	(No correspondence with DO-178B Areas) (No correspondence with DO-178B Areas)																																				
ADV – Development ADV_FSP Functional Specification ADV_HLD High Level Design ADV_IMP Implementation of TSF ADV_INT Minimization Complexity ADV_LLD Low Level Design ADV_RCR Correspondence Demo ADV_SPM Security Policy Model	<table border="1"> <thead> <tr> <th colspan="5">Software Development Process</th> </tr> <tr> <th>Requirements</th> <th>Design</th> <th>Code</th> <th>Integrate</th> <th>Trace</th> </tr> </thead> <tbody> <tr> <td>X</td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td></td> <td>X</td> <td></td> <td>X</td> <td></td> </tr> <tr> <td></td> <td></td> <td></td> <td>X</td> <td></td> </tr> <tr> <td></td> <td></td> <td></td> <td>X</td> <td></td> </tr> <tr> <td>X</td> <td></td> <td></td> <td></td> <td></td> </tr> </tbody> </table>	Software Development Process					Requirements	Design	Code	Integrate	Trace	X						X		X					X					X		X					Life Cycle Data Sect. 11.6, 11.9, 11.14 Sect. 11.7, 11.10, 11.14 Sect. 11.14, 11.11 Sect. 11.7, 11.10 Sect. 11.14 Sect 11.9, 11.14
Software Development Process																																					
Requirements	Design	Code	Integrate	Trace																																	
X																																					
	X		X																																		
			X																																		
			X																																		
X																																					
AGD – Guidance Documents AGD_ADM Administrative AGD_USR User	(No Correspondence with DO-178B Areas) (No Correspondence with DO-178B Areas)																																				
ALC – Life Support ALC_DVS Sufficiency Security ALC_LCD Measurable Model ALC_TAT Compliance Standards	<table border="1"> <thead> <tr> <th rowspan="2">Activ.</th> <th colspan="3">Software Planning</th> <th>Review</th> </tr> <tr> <th>Plans</th> <th>Life/Cycle/Env. Stand</th> <th>Plans</th> <th>Plans</th> </tr> </thead> <tbody> <tr> <td>X</td> <td></td> <td>X</td> <td></td> <td>X</td> </tr> <tr> <td></td> <td></td> <td>X</td> <td>X</td> <td>X</td> </tr> </tbody> </table>	Activ.	Software Planning			Review	Plans	Life/Cycle/Env. Stand	Plans	Plans	X		X		X			X	X	X	Life Cycle Data Sect. 11.2 Sect. 11.2																
Activ.	Software Planning			Review																																	
	Plans	Life/Cycle/Env. Stand	Plans	Plans																																	
X		X		X																																	
		X	X	X																																	
Software Verification Process ATE – Tests ATE_Cov Coverage ATE_DPT Impement. Represent. ATE_FUN Functional Testing ATE_IND Independent Testing	<table border="1"> <thead> <tr> <th rowspan="2">Activities</th> <th colspan="2">Life Cycle Data</th> </tr> <tr> <th>Review and Analysis</th> <th>Testing</th> </tr> </thead> <tbody> <tr> <td></td> <td>X</td> <td>X</td> </tr> <tr> <td></td> <td>X</td> <td>X</td> </tr> <tr> <td></td> <td></td> <td>X</td> </tr> <tr> <td></td> <td></td> <td>X</td> </tr> </tbody> </table>	Activities	Life Cycle Data		Review and Analysis	Testing		X	X		X	X			X			X	Sect. 11.3, 11.13, 11.14																		
Activities	Life Cycle Data																																				
	Review and Analysis	Testing																																			
	X	X																																			
	X	X																																			
		X																																			
		X																																			
AVA – Vulnerability Assessment AVA_CCA Covert Channel AVA_MSU Insecure States AVA_SOF Functional Eval AVA_VLA Highly Resistant	(No Correspondence with DO-178B Areas) (No Correspondence with DO-178B Areas) (No Correspondence with DO-178B Areas) (No Correspondence with DO-178B Areas) (No Correspondence with DO-178B Areas)																																				

Summary of Non-Mapping CC Classes

While most of the classes within the Common Criteria could be mapped to a one of the DO-178B processes, several Common Criteria classes relating to just security issues had no mapping. These included the AGD – Guidance Documents, ADO- Delivery and Operation and AVA-Vulnerability Assessment CC classes. Guidance documents refer to documents specifically related to the security aspects of administration and operation of the software. DO-178B does not address documentation relating to just the security aspects of system operation or for that matter any specific operation. The Delivery and Operation class seeks to insure that the software was delivered without interference or tampering and that the software is installed and initially started securely and correctly. Vulnerability Assessment deals specifically with covert channel analysis, deliberate misuse and other security function assessments that are completely absent in DO-178B.

DO-178B is a general software developers guide for ensuring that software development practices meet FAA safety and reliability standards but not specifically aimed at security. Consequently, those CC classes that deal only with security issues will not match any of the DO-178B process requirements and will thus have to be added to a set of requirements intended to conform to both CC and DO-178B requirements.

Considerations in Mapping DO-178B to the Common Criteria

Aside from the previously identified CC classes, ADV, ADO and ADG that do not correspond to DO-178B development processes, most of the security class components map at some level to the DO-178B process requirements. Details of the mapping are discussed in the document, *Towards Common Criteria Certification for DO-178B Compliant Airborne Software Systems*. However, differences between the intended purposes of the two documents needs to be stated since the differences are important to the outcome of merging requirements from both documents.

DO-178B is intended to certify that software used in aircraft is developed with "best known" practices and does not contribute to aircraft safety hazards. Software is not ever certified as a standalone component but only as a part of aircraft or engine type. Emphasis in DO-178B is in outlining general policies and procedures to produce safe software in terms of airworthiness requirements and to produce documentation to substantiate that the development requirements have been met. Thus, language and content is high-level and abstract leaving a lot of compliance decisions up to the developer.

The Common Criteria (CC) is intended to specify security requirements that a system, hardware or software must satisfy in order to achieve a specific level of assurance. The CC only deals with security functionality of systems and does not address overall development issues except where they affect security. The CC is a much more detailed document in terms of specifying how compliance is achieved for an intended product.

Each component of each assurance class has specific action elements and evidence of compliance for both developers and evaluators.

Merging CC security requirements into DO-178B will need to address the differences in detail so that none of the CC functionality is lost.

Another important consideration is the addition of security requirements to other FAA documents needed for aircraft certification. In researching the FAA certification process, several additional documents appear to be as important as the DO-178B in guiding airborne system development. These documents are the ARP4754, *Certification Considerations for Highly-Integrated or Complex Aircraft Systems* and DO-254, *Design Assurance Guidance for Airborne Electronic Hardware*. Review of these documents is beyond the scope of this mapping task, but it is possible that CC classes should be mapped to one or both of these documents. Integrating security functionality into the FAA certification process needs to be addressed for the total system being evaluated, not just the software that will be integrated into the aircraft since the CC's scope encompasses entire systems. Consequently, a mapping between DO-178B and the CC will only constitute part of the process for achieving certification by the FAA for a high safety level system and by the NSA for a high-level assurance system. In conclusion, the mapping and integration of CC requirements will need to be extended beyond DO-178B to satisfy CC certification.