RTCA, Inc.
1150 18th Street, NW, Suite 910
Washington, DC 20036-3816 USA

Software Considerations in Airborne Systems and Equipment Certification

RTCA DO-178C
December 13, 2011

Prepared by: SC-205

Copies of this document may be obtained from

RTCA, Inc.

Telephone: 202-833-9339
Facsimile: 202-833-9434
Internet: www.rtca.org

Please visit the RTCA Online Store for document pricing and ordering information.

FOREWORD

This report was prepared by RTCA Special Committee 205 (SC-205) and EUROCAE Working Group 71 (WG-71) and approved by the RTCA Program Management Committee (PMC) on December 13, 2011.

RTCA, Incorporated is a not-for-profit corporation formed to advance the art and science of aviation and aviation electronic systems for the benefit of the public. The organization functions as a Federal Advisory Committee and develops consensus-based recommendations on contemporary aviation issues. RTCA's objectives include but are not limited to:

- coalescing aviation system user and provider technical requirements in a manner that helps government and industry meet their mutual objectives and responsibilities;

- analyzing and recommending solutions to the system technical issues that aviation faces as it continues to pursue increased safety, system capacity, and efficiency;

- developing consensus on the application of pertinent technology to fulfill user and provider requirements, including development of minimum operational performance standards for electronic systems and equipment that support aviation; and

- assisting in developing the appropriate technical material upon which positions for the International Civil Aviation Organization and the International Telecommunication Union and other appropriate international organizations can be based.

The organization's recommendations are often used as the basis for government and private sector decisions as well as the foundation for many Federal Aviation Administration Technical Standard Orders.

Since the RTCA is not an official agency of the United States Government, its recommendations may not be regarded as statements of official government policy unless so enunciated by the U.S. government organization or agency having statutory jurisdiction over any matters to which the recommendations relate.

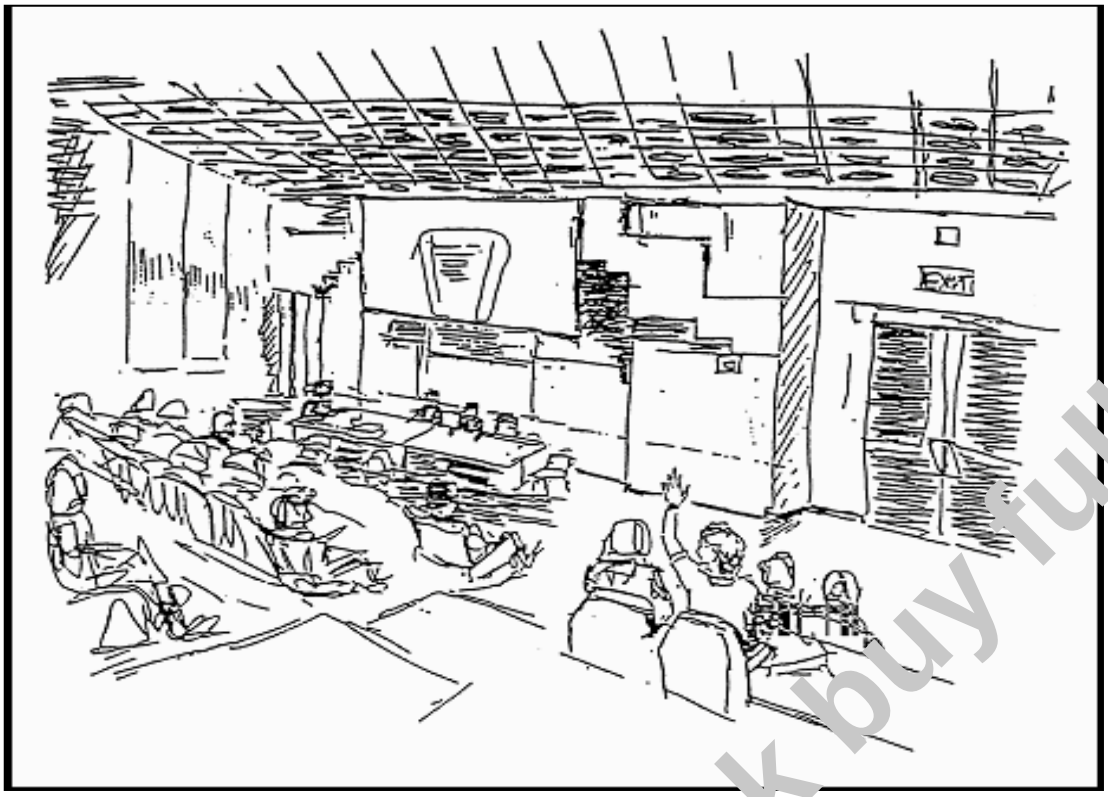

Illustration provided by Pat Neilan, UK CAA

CONSENSUS n. Collective opinion or concord; general agreement or accord. [Latin, from consentire, to agree]

TABLE OF CONTENTS

1.0 INTRODUCTION 1

1.1 Purpose 1
1.2 Scope 1
1.3 Relationship to Other Documents 2
1.4 How to Use This Document 2
1.5 Document Overview 4

2.0 SYSTEM ASPECTS RELATING TO SOFTWARE DEVELOPMENT 7

2.1 System Requirements Allocation to Software 7
2.2 Information Flow Between System and Software Life Cycle Processes 10
2.2.1 Information Flow from System Processes to Software Processes 10
2.2.2 Information Flow from Software Processes to System Processes 10
2.2.3 Information Flow between Software Processes and Hardware Processes 11
2.3 System Safety Assessment Process and Software Level 11
2.3.1 Relationship between Software Errors and Failure Conditions 12
2.3.2 Failure Condition Categorization 13
2.3.3 Software Level Definition 14
2.3.4 Software Level Determination 14
2.4 Architectural Considerations 15
2.4.1 Partitioning 15
2.4.2 Multiple-Version Dissimilar Software 16
2.4.3 Safety Monitoring 16
2.5 Software Considerations in System Life Cycle Processes 17
2.5.1 Parameter Data Items 17
2.5.2 User-Modifiable Software 18
2.5.3 Commercial-Off-The-Shelf Software 19
2.5.4 Option-Selectable Software 19
2.5.5 Field-Loadable Software 19
2.5.6 Software Considerations in System Verification 20
2.6 System Considerations in Software Life Cycle Processes 20

3.0 SOFTWARE LIFE CYCLE 21

3.1 Software Life Cycle Processes 21
3.2 Software Life Cycle Definition 21
3.3 Transition Criteria Between Processes 22

4.0 SOFTWARE PLANNING PROCESS 25

4.1 Software Planning Process Objectives 25
4.2 Software Planning Process Activities 25
4.3 Software Plans 26
4.4 Software Life Cycle Environment Planning 27
4.4.1 Software Development Environment 28
4.4.2 Language and Compiler Considerations 28
4.4.3 Software Test Environment 29
4.5 Software Development Standards 29
4.6 Review of the Software Planning Process 30

5.0 SOFTWARE DEVELOPMENT PROCESSES 31
5.1 Software Requirements Process 32
5.1.1 Software Requirements Process Objectives 32
5.1.2 Software Requirements Process Activities 32
5.2 Software Design Process 33
5.2.1 Software Design Process Objectives 33
5.2.2 Software Design Process Activities 33
5.2.3 Designing for User-Modifiable Software 34
5.2.4 Designing for Deactivated Code 34
5.3 Software Coding Process 35
5.3.1 Software Coding Process Objectives 35
5.3.2 Software Coding Process Activities 35
5.4 Integration Process 35
5.4.1 Integration Process Objectives 36
5.4.2 Integration Process Activities 36
5.5 Software Development Process Traceability 37

6.0 SOFTWARE VERIFICATION PROCESS 39
6.1 Purpose of Software Verification 39
6.2 Overview of Software Verification Process Activities 40
6.3 Software Reviews and Analyses 41
6.3.1 Reviews and Analyses of High-Level Requirements 41
6.3.2 Reviews and Analyses of Low-Level Requirements 42
6.3.3 Reviews and Analyses of Software Architecture 42
6.3.4 Reviews and Analyses of Source Code 43
6.3.5 Reviews and Analyses of the Outputs of the Integration Process 44
6.4 Software Testing 44
6.4.1 Test Environment 46
6.4.2 Requirements-Based Test Selection 46
6.4.2.1 Normal Range Test Cases 46
6.4.2.2 Robustness Test Cases 47
6.4.3 Requirements-Based Testing Methods 47
6.4.4 Test Coverage Analysis 49
6.4.4.1 Requirement-Based Test Coverage Analysis 49
6.4.4.2 Structural Coverage Analysis 49
6.4.4.3 Structural Coverage Analysis Resolution 50
6.4.5 Reviews and Analyses of Test Cases, Procedures, and Results 51
6.5 Software Verification Process Traceability 51
6.6 Verification of Parameter Data Items 51

7.0 SOFTWARE CONFIGURATION MANAGEMENT PROCESS 53
7.1 Software Configuration Management Process Objectives 53
7.2 Software Configuration Management Process Activities 54
7.2.1 Configuration Identification 54
7.2.2 Baselines and Traceability 55
7.2.3 Problem Reporting, Tracking, and Corrective Action 55
7.2.4 Change Control 56
7.2.5 Change Review 56
7.2.6 Configuration Status Accounting 56

7.2.7 Archive, Retrieval, and Release 57
7.3 Data Control Categories 58
7.4 Software Load Control 58
7.5 Software Life Cycle Environment Control 59

8.0 SOFTWARE QUALITY ASSURANCE PROCESS 61

8.1 Software Quality Assurance Process Objectives 61
8.2 Software Quality Assurance Process Activities 61
8.3 Software Conformity Review 62

9.0 CERTIFICATION LIAISON PROCESS 65

9.1 Means of Compliance and Planning 65
9.2 Compliance Substantiation 65
9.3 Minimum Software Life Cycle Data Submitted to Certification Authority 66
9.4 Software Life Cycle Data Related to Type Design 66

10.0 OVERVIEW OF CERTIFICATION PROCESS 67

10.1 Certification Basis 67
10.2 Software Aspects of Certification 67
10.3 Compliance Determination 67

11.0 SOFTWARE LIFE CYCLE DATA 69

11.1 Plan for Software Aspects of Certification 70
11.2 Software Development Plan 71
11.3 Software Verification Plan 71
11.4 Software Configuration Management Plan 72
11.5 Software Quality Assurance Plan 73
11.6 Software Requirements Standards 74
11.7 Software Design Standards 74
11.8 Software Code Standards 75
11.9 Software Requirements Data 75
11.10 Design Description 75
11.11 Source Code 76
11.12 Executable Object Code 76
11.13 Software Verification Cases and Procedures 76
11.14 Software Verification Results 77
11.15 Software Life Cycle Environment Configuration Index 77
11.16 Software Configuration Index 77
11.17 Problem Reports 78
11.18 Software Configuration Management Records 78
11.19 Software Quality Assurance Records 79
11.20 Software Accomplishment Summary 79
11.21 Trace Data 80
11.22 Parameter Data Item File 80

12.0 ADDITIONAL CONSIDERATIONS 81

12.1 Use of Previously Developed Software 81
12.1.1 Modifications to Previously Developed Software 81
12.1.2 Change of Aircraft Installation 81

12.1.3 Change of Application or Development Environment 82
12.1.4 Upgrading a Development Baseline 83
12.1.5 Software Configuration Management Considerations 84
12.1.6 Software Quality Assurance Considerations 84
12.2 Tool Qualification 84
12.2.1 Determining if Tool Qualification is Needed 84
12.2.2 Determining the Tool Qualification Level 85
12.2.3 Tool Qualification Process 85
12.3 Alternative Methods 86
12.3.1 Exhaustive Input Testing 86
12.3.2 Considerations for Multiple-Version Dissimilar Software Verification 86
12.3.2.1 Independence of Multiple-Version Dissimilar Software 87
12.3.2.2 Multiple Processor-Related Verification 88
12.3.2.3 Multiple-Version Source Code Verification 88
12.3.2.4 Tool Qualification for Multiple-Version Dissimilar Software 88
12.3.2.5 Multiple Simulators and Verification 88
12.3.3 Software Reliability Models 89
12.3.4 Product Service History 89
12.3.4.1 Relevance of Service History 90
12.3.4.2 Sufficiency of Accumulated Service History 91
12.3.4.3 Collection, Reporting, and Analysis of Problems Found During Service History 91
12.3.4.4 Service History Information to be Included in the Plan for Software Aspects of Certification 92

ANNEX A – PROCESS OBJECTIVES AND OUTPUTS BY SOFTWARE LEVEL 95

ANNEX B – ACRONYMS AND GLOSSARY OF TERMS 107

APPENDIX A – BACKGROUND OF DO-178C/ED-12C DOCUMENT A-1

APPENDIX B – COMMITTEE MEMBERSHIP B-1

LIST OF FIGURES

Figure 1-1 Document Overview 5
Figure 2-1 Information Flow Between System and Software Life Cycle Processes 9
Figure 2-2 Sequence of Events for Software Error Leading to a Failure Condition 12
Figure 3-1 Example of a Software Project Using Four Different Development Sequences 22
Figure 6-1 Software Testing Activities 45

LIST OF TABLES

Table 2-1 Failure Condition Category Descriptions 13
Table 7-1 SCM Process Activities Associated with CC1 and CC2 Data 58
Table 12-1 Tool Qualification Level Determination 85
Table A-1 Software Planning Process 96
Table A-2 Software Development Processes 97
Table A-3 Verification of Outputs of Software Requirement Process 98
Table A-4 Verification of Outputs of Software Design Process 99
Table A-5 Verification of Outputs of Software Coding & Integration Processes 100
Table A-6 Testing of Outputs of Integration Process 101
Table A-7 Verification of Verification Process Results 102
Table A-8 Software Configuration Management Process 103
Table A-9 Software Quality Assurance Process 104
Table A-10 Certification Liaison Process 105

This Page Intentionally Left Blank

1.0 INTRODUCTION

The rapid increase in the use of software in airborne systems and equipment used on aircraft and engines in the early 1980s resulted in a need for industry-accepted guidance for satisfying airworthiness requirements. DO-178, "Software Considerations in Airborne Systems and Equipment Certification", was written to satisfy this need.

This document, now revised in the light of experience, provides the aviation community with guidance for determining, in a consistent manner and with an acceptable level of confidence, that the software aspects of airborne systems and equipment comply with airworthiness requirements. As software use increases, technology evolves, and experience is gained in the application of this document, this document will be reviewed and revised. Appendix A provides the background of this document.

1.1 Purpose

The purpose of this document is to provide guidance for the production of software for airborne systems and equipment that performs its intended function with a level of confidence in safety that complies with airworthiness requirements. This guidance includes:

- Objectives for software life cycle processes.
- Activities that provide a means for satisfying those objectives.
- Descriptions of the evidence in the form of software life cycle data that indicate that the objectives have been satisfied.
- Variations in the objectives, independence, software life cycle data, and control categories by software level.
- Additional considerations (for example, previously developed software) that are applicable to certain applications.
- Definition of terms provided in the glossary.

In addition to guidance, supporting information is provided to assist the reader’s understanding.

1.2 Scope

This document discusses those aspects of certification that pertain to the production of software for airborne systems and equipment used on aircraft, engines, propellers and, by region, auxiliary power units. In discussing those aspects, the system life cycle and its relationship with the software life cycle is described to aid in the understanding of the certification process. A complete description of the system life cycle processes, including the system safety assessment and validation processes, or the certification process is not intended.

The guidance contained in this document does not define or imply the level of involvement of a certification authority in a certification process. To understand certification authority involvement, the applicant should refer to applicable regulations and guidance material issued by the relevant certification authority.