EUROPEAN ORGANISATION FOR THE SAFETY OF AIR NAVIGATION

EUROCONTROL

ANS

SOFTWARE

LIFECYLE

SAF.ET1.ST03.1000.REP-01-00

Edition	:	3.0
Edition Date	:	21/12/2005
Status	:	Released Issue
Class	:	General Public

EUROPEAN AIR TRAFFIC MANAGEMENT

DOCUMENT IDENTIFICATION SHEET

DOCUMENT DESCRIPTION

Document Title

ANS Software Lifecycle

EWP DELIVERABLE REFERENCE NUMBER	
PROGRAMME REFERENCE INDEX	**EDITION :** 3.0
SAF.ET1.ST03.1000.REP-01-00	**EDITION DATE :** 21/12/2005

Abstract

This document provides guidance material for defining an ANS software lifecycle. It also provides references to five existing standards (ED109, IEC12207, IEC61508, ED12B/DO178B and CMMi) and how these standards cover ANS needs.

Keywords

Software	Safety Assurance	Software Safety Assurance System
Standards	Quality Assurance	SWAL
Safety Assessment	Assurance Level	

CONTACT PERSON : P.MANA	**TEL :** 93295	**DIVISION :** DAP/SAF

DOCUMENT STATUS AND TYPE

STATUS		CATEGORY		CLASSIFICATION	
Working Draft	□	Executive Task	□	General Public	☑
Draft	□	Specialist Task	☑	EATMP	□
Proposed Issue	□	Lower Layer Task	□	Restricted	□
Released Issue	☑				

ELECTRONIC BACKUP

INTERNAL REFERENCE NAME :

HOST SYSTEM	MEDIA	SOFTWARE(S)
Microsoft Windows	Type : Hard disk	
	Media Identification :	

DOCUMENT APPROVAL

THE FOLLOWING TABLE IDENTIFIES ALL MANAGEMENT AUTHORITIES WHO HAVE SUCCESSIVELY APPROVED THE PRESENT ISSUE OF THIS DOCUMENT.

AUTHORITY	NAME AND SIGNATURE	DATE
Chairman of the EATMP Software Task Force	P.MANA	21/12/2005
Chairman of the Safety Assessment Methodology Task Force	P.MANA	21/12/2005
Chairman of the Safety Team	E.MERCKX	21/12/2005
DAP Director	G.PAULSON	21/12/2005

DOCUMENT CHANGE RECORD

THE FOLLOWING TABLE RECORDS THE COMPLETE HISTORY OF THE SUCCESSIVE EDITIONS OF THE PRESENT DOCUMENT.

EDITION	DATE	REASON FOR CHANGE	SECTIONS PAGES AFFECTED
0.1	11/06/1999	First Issue	All
0.2	10/12/1999	Second Working Draft Issue, after review by the Task Force	All
0.3	03/03/2000	Third Working Draft Issue, after review by the Software Task Force	All
1.0	29/10/2001	Proposed Issue, after review by Software Task Force	All
1.1	23/11/201	Proposed Issue after SWTF/4 meeting. The document title changed (old title: Software Standards Analysis)	Introduction & Part I
2.0	25/04/2002	Released Issue, after final review	Part I
3.0	21/12/2005	Second Released Issue, consistency with SAM V2.0, Recommendations for ANS SW V1.0, ESARR6 V1.0	Part I

TABLE OF CONTENTS

CHAPTER 1- GENERAL INTRODUCTION

CHAPTER 2- SOFTWARE STANDARDS OVERVIEW

1

GENERAL INTRODUCTION

1 PURPOSE

An increasing proportion of ANS (Air Navigation System) functions is implemented by software and these functions are becoming more safety-critical. It is therefore necessary to define guidance on how assurance may be provided for software.

To complement the EATMP Air Navigation System Safety Assessment Methodology, initial material is needed for establishing such guidance and recommendations on the major activities required providing the appropriate safety and quality assurance level for software in Air Navigation Systems.

A system throughout this document is composed of: people, procedure and equipment (Software, Hardware, Human Machine Interface (HMI)).

However today, no ANS software-related standard exists which neither fulfils ANS specificities (especially for ground part of ANS), nor is widely spread and extensively used by ANS community (at least not enough to become a de facto standard).

Consequently, some standards have been chosen on which to base recommendations.

The objective of this document is not to promote any standard or to rank them. It just intends to identify the objectives/activities/tasks required by each standard and to describe their commonalities and differences.

The main objectives of this document are:

- **to define an ANS software lifecycle**
- **to allow these different organisations to assess their own practices with respect to this recommended software lifecycle and to these standards**.

The purposes of this document are:

- To define a recommended software lifecycle that matches ANS needs (Part I);
- To refer to existing standards developed for other domains of application (Part I);
- To assess the suitability of these standards for the definition, development, operation and maintenance of Air Navigation System software (Part II);
- To provide compatibility/traceability matrix between standards. For each process/activity of this recommended ANS software lifecycle, a reference will be provided to standards paragraphs that cover it either fully or partially (Part II);
- To provide for each of the five standards a coverage matrix, which identifies which processes/objectives of each standard are part of this recommended ANS software life (Part II);
- To provide the main omissions of these five standards as far as ANS needs are concerned.

2 SCOPE

The software lifecycle described in this document applies to Air Navigation System Software.

3 APPROACH

As no safety and/or quality standard dedicated to ANS exists so far, the approach has been to perform a survey of existing software related standards what ever their domain of application.

Some safety-oriented standards exist such as ED12B/DO178B but which deals with airborne software in a certification environment or IEC 61508: a generic standard, which first requires to be tailored to a domain of application (this has not yet been done for ANS).

The selection of international standards is the following:

- **ISO/IEC 12207**
  Information Technology - Software Engineering - Software Life-Cycle Processes (November 1995).
- **ED109**

  Guidelines for Communication, Navigation, Surveillance, and Air Traffic Management (CNS/ ATM) Systems Software Integrity Assurance (March 2002)
- **IEC 61508-3**
  Functional safety of electrical/electronic/programmable electronic safety-related systems
  Part 3: Software Requirements (Draft Standard)
- **ED12B/DO178B**
  Software Considerations in Airborne Systems and Equipment Certification (December 1992)
- **CMMI**

Capability Maturity Model Integration (V1.1 March 2002)

Then an analysis of these standards and the identification of ANS specificities led to the definition of a ANS software lifecycle.

This ANS software lifecycle covers ***quality*** and ***safety*** related activities from the beginning of the system definition till decommissioning.

The intent of this document is not to define a new standard but to establish a reference against which to assess own practices.

The approach elaborated relies on the analysis of best practices both from other domains using dedicated standards and also from ANS using the feedback of ANS stakeholders (regulatory bodies, ATS providers, industry, consultants, …).

4 STRUCTURE OF THE CURRENT ISSUE

This current issue includes:

- **Introduction**:
  - A general introduction identifying the purpose, scope, approach and content of this document (Chapter 1)
  - A brief description of the five selected quality and safety standards for software development (Chapter 2).
- **Part I**: ANS Software Lifecycle Definition
  - An ANS software lifecycle is defined. This ANS software lifecycle is based on IEC/ISO 12207, but this does neither mean that this standard best fits ANS needs nor that it is the recommended one.
  - A reference to those standards is provided. The purpose of this reference is to provide compatibility/traceability matrix between a recommended ANS software lifecycle and these standards.
  - The definition of the recommended ANS software lifecycle includes the following:
    - Software Safety Assurance System (Chapter 1)
    - Primary Lifecycle Processes (Chapter 2)
    - Supporting Lifecycle Processes(Chapter 3)
    - Organisational Lifecycle Processes (Chapter 4)
    - Additional Software Lifecycle Objectives (Chapter 5)
- **Part II**: Software Standards Coverage
  - This part identifies how each of the five standards is covered by the recommended ANS software lifecycle. Each standard paragraph or

clause, which has been integrated in the ANS recommendations, is identified as such and a reference to the ANS recommendations paragraph is provided.

- This part also identifies the main omissions of these five standards as far as ANS particularities and the width of our scope are concerned.

5 APPLICABILITY OF THE DOCUMENT

This document recognises that the guidelines herein are not mandated by law, but represent a consensus of the ANS community on what are or should be the best practices.

It also recognises that alternative methods to the methods described herein may be available to the stakeholders. For these reasons, the use of words such as “shall’ and ‘must” is avoided, therefore all statements are using “should”.

6 TARGET AUDIENCE

This document is specifically targeted at:

Safety practitioners: Correct process in a methodologically correct way.

They are responsible for:

the link between the programme/project and the safety assessment process, the methodological support to the different steps of the safety assessment process and the integration within the organisation Safety Management System (SMS).

For example, the safety practitioners have to ensure that SWAL is allocated in accordance with Chapter 2, and that SWAL is validated.

Software Team: Application in their domain knowledge.

They use “ANS Software Lifecycle” to apply “Recommendations for ANS SW” for a specific software.

For example, software team is responsible for the implementation of objectives of the allocated SWAL and for the verification & validation of their satisfaction.

Project/Programme Manager or Safety Manager.

7 READERSHIP

The following table suggests a minimum reader's attention to this document.

	Software Team	Safety Practitioner	Other roles (Programme/project Manager, Safety Manager, ..)
Cover - Chapter 1 General Introduction	🕮	🕮	✓
Cover - Chapter 2 – SW Standards overview	✓	🕮	N/A
Part I - Introduction	🕮	🕮	✓
Part I - Chapter 1 – Software Safety Assurance System	🕮	🕮	🕮
Part I - Chapter 2 – Primary lifecycle	🕮	🕮	✓
Part I - Chapter 3 – Supporting Lifecycle	🕮	🕮	✓
Part I - Chapter 4 – Organisational Lifecycle	✓	🕮	✓
Part I - Chapter 5 – Additional Lifecycle	🕮	🕮	✓
Part II -Introduction	✓	🕮	N/A
Part II – Chapter 1 – ED12B/DO178B	✓*	🕮*	✓*
Part II - Chapter 2 – IEC61508	✓*	🕮*	✓*
Part II - Chapter 3 – ISO/IEC 12207	✓*	🕮*	✓*
Part II - Chapter 4 – ED109/DO278	✓*	🕮*	✓*
Part II – Chapter 5 – CMMi	✓*	🕮*	✓*

*: valid only for the standard being used by the organisation. Otherwise: N/A.

🕮: Detailed knowledge;
✓: Aware;
N/A: Not Applicable.

8 HOW TO USE THIS DOCUMENT

This document can be used for the following purposes (see table example):

1. **Identification of the ANS software lifecycle.** The document user will have access to this reference lifecycle and its activities in the columns:
   - N°: activity Number;
   - Activity Title: Reference name of the activity;
   - Activity: Detailed description of the activity.
2. **Assessment of its own practices.** If the document user applies one of the five pre-assessed standards (ISO/IEC 12207, ED109, ED12B/DO178B, IEC61508 and CMMI), he/she will be able to compare directly his own practices with the reference lifecycle and its associated activities by reading the relevant column of the selected standard (the exact reference is provided):
   - ● (means fully covered: this standard proposes an equivalent activity);
   - **P** (partially covered: this standard does not fully provide an equivalent activity);
   - blank (missing: this standard does not provide an equivalent activity);
3. **Identification of activities (how) to satisfy an objective (what) listed in "Recommendations for ANS Software".** The document user will search the objective (Column Obj) to be satisfied for a specific SWAL and will find the proposed activities (Activity) to contribute to satisfy the objective (many activities could be necessary to satisfy one objective) and also how one of the 5 standards proposes to achieve this activity.

N°	Obj	Activity Title	Activity	ISO/IEC 12207	ED109	ED-12B/ DO 178B	IEC 61508	CMMI
2	4.3.4	Assurance Level Related Requirements	Software requirements are commensurate with the allocated Assurance Level.		● (Ref: 3 A2.1, A2.2)	● (Ref: 5.1.2, 11.9)	● (Ref: 7.2.2)	P (Ref: RD 3.3)
3	4.3.4, 4.3.15	Software Requirements Definition Criteria	The developer should specify & document the software requirements considering the criteria listed below. a) Traceability to system requirements and system design; b) External consistency with system requirements; c) Internal consistency; d) Testability; e) Feasibility of software design; f) Feasibility of operation and maintenance.	● (Ref: 5.3.4.2)	● (Ref: 3.3. Table A2.1, A2.2 , A-3 line 6)	● (Ref: 5.5, 11.6, 11.9)	P (Ref: 7.2.2.1, 7.2.2.2, 7.2.2.6)	● (Ref: RD 3.3 ReqM 1.4)
4	4.3.9, 4.3.10 4.3.11 4.3.12	Software Requirements Standards	Definition of methods, rules and tools to be used to develop software requirements.		● (Ref: 3.2 Tables A2.1, A2.2)	● (Ref: 11.6)	● (Ref: 7.2.2.4, 7.2.2.6)	● (Ref: RD GP 2.2, 2.3, 3.1)

2

SOFTWARE STANDARDS OVERVIEW

1 INTRODUCTION

The purpose of this chapter is to provide a brief description of major software quality and/or safety standards.

This description intends to identify:

- the organisation, which has defined the standard,
- the scope of the standard, i.e. the list of processes and objectives or activities, which are to be performed during the lifecycle of the software development,
- the status of their use (industry, domain, …) and of their issue (recent, to be updated, …)
- safety-related considerations.

2 GENERAL PRESENTATION OF STANDARDS

Five standards are considered in this document:

- **ISO/IEC 12207**
  Information Technology - Software Engineering - Software Life-Cycle Processes (November 1995).

- **ED109/DO278**

  Guidelines for Communication, Navigation, Surveillance, and AIR TRAFFIC MANAGEMENT (CNS/ ATM) Systems Software Integrity Assurance (March 2002)

- **IEC 61508-3**
  Functional safety of electrical/electronic/programmable electronic safety-related systems
  Part 3: Software Requirements

- **ED12B/DO178B**
  Software Considerations in Airborne Systems and Equipment Certification (December 1992)

- **CMMI**

  Capability Maturity Model Integration (V1.1 March 2002)

Standards scope and their interrelationships are shown in Figure I.

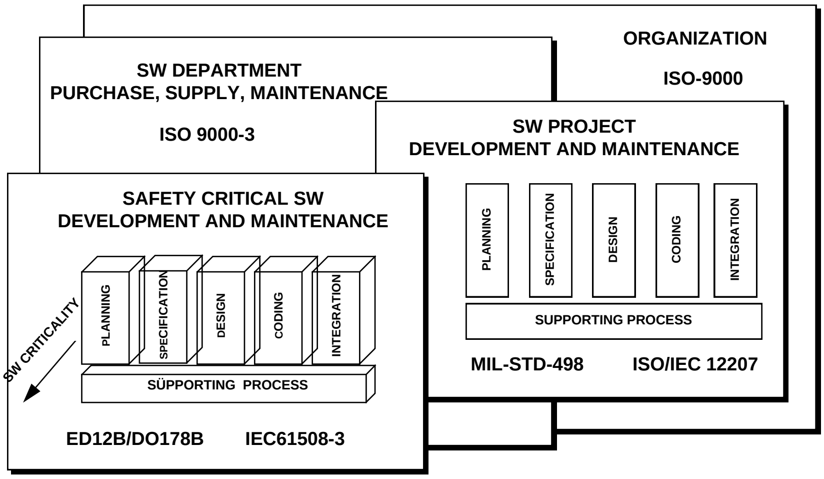


Figure I. Scope and Interrelationships of Standards

The ISO/IEC 12207 Standard is currently considered as reflecting the best practices for all processes and activities of a Software lifecycle.

The IEC 61508-3 and the ED12B/DO178B cover the lifecycle of safety critical software. The IEC 61508-3 is part of an emerging generic standard (IEC 61508) addressing the functional safety of safety-related systems (in particular of the Equipment Under control (EUC), Cf: Annex A §2). This generic standard is expected to be tailored to a specific sector of application.

The EB12B/DO178B Standard defines recommended practices for the development of software in airborne systems and equipment. The Standard is not mandatory, but represents an international consensus in the avionics industry.

As explained in Chapter 1, only five standards have been considered further in the following document.

The MIL-STD-498 has been used in ANS industry. This standard is now superseded by the ISO/IEC 12207.

2.1 ISO/IEC 12207

This international Standard establishes a common framework for software lifecycle processes. The Standard specifies a comprehensive set of processes (described in terms of activities and tasks) covering all aspects of the software lifecycle. This international Standard groups the activities that may be performed during the lifecycle of software into five primary processes, eight supporting processes, and four organisational processes. These lifecycle processes are illustrated in Figure II.

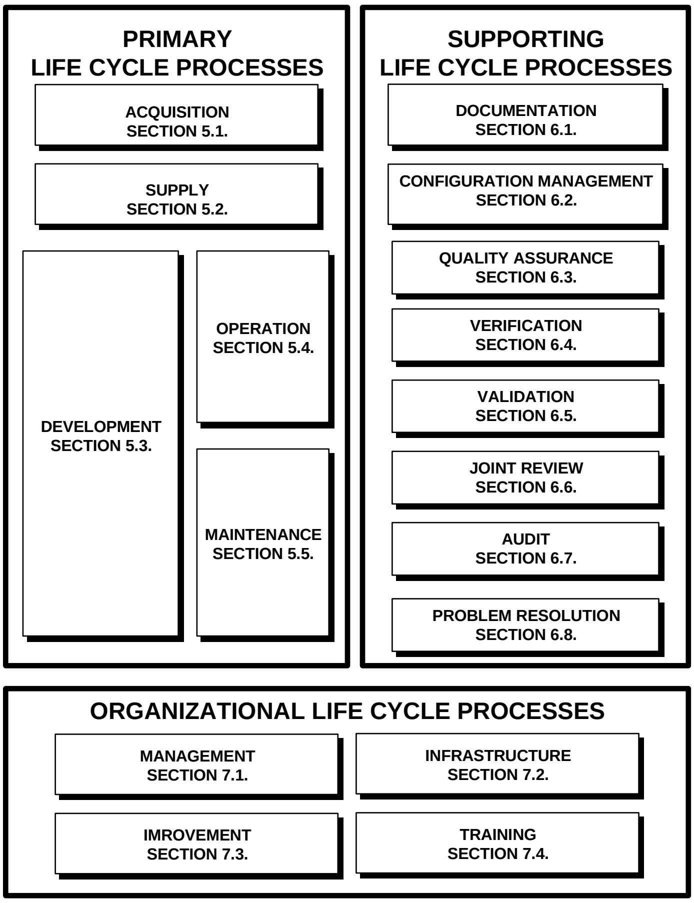


Figure II. Scope of ISO/IEC 12207 Standard

This international Standard is designed to be tailored for an individual organisation, project or application: an organisation, depending on its purpose, can select an appropriate subset to fulfil that purpose. In addition, the framework provides for controlling and improving these processes.

2.2 IEC 61508-3

The international Standard IEC 61508 sets out a generic approach for all safety lifecycle activities for systems comprised of electrical and/or electronic, and/or programmable electronic components (electrical/electronic/programmable electronic systems (E/E/PESs)).

This standard was originally designed based on the following principle:

> A safety-related protection (monitoring) software controls an industrial process (EUC: Equipment Under Control), and can stop it.

As a standard, it:

- is not sector specific
- addresses a few design issues
- is primarily a process standard (mainly dedicated to safety management system, but not to “certify” or “qualify” or get approval for a product).

IEC61508 is a generic standard for all safety lifecycle activities for systems that are used to perform safety functions, which:

- provides a method for the development of the safety requirements specification necessary to achieve the required functional safety for safety-related systems
- adopts a broad range of principles, techniques and measures to achieve functional safety

The standard consists of seven parts:

- Part 1: General requirements;
- Part 2: Requirements for electrical/electronic/programmable electronic systems (E/E/PES);
- Part 3: Software requirements;
- Part 4: Definitions and abbreviations;
- Part 5: Examples of methods for the determination of safety integrity levels;
- Part 6: Guidelines on the application of parts 2 and 3;
- Part 7: Overview of techniques and measures.

Relationships between the seven parts are shown in Figure III.

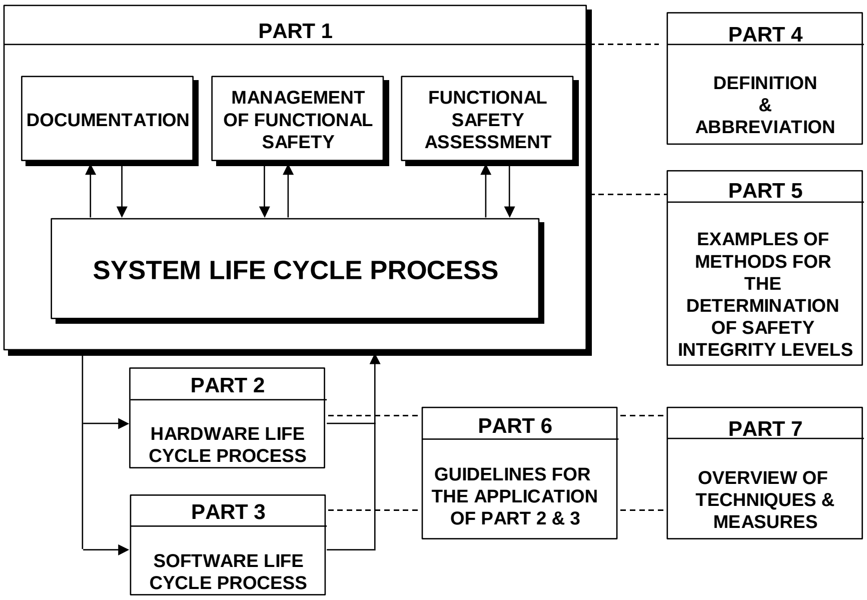

PART 1
DOCUMENTATION
MANAGEMENT OF FUNCTIONAL SAFETY
FUNCTIONAL SAFETY ASSESSMENT
SYSTEM LIFE CYCLE PROCESS
PART 4
DEFINITION & ABBREVIATION
PART 5
EXAMPLES OF METHODS FOR THE DETERMINATION OF SAFETY INTEGRITY LEVELS
PART 2
HARDWARE LIFE CYCLE PROCESS
PART 3
SOFTWARE LIFE CYCLE PROCESS
PART 6
GUIDELINES FOR THE APPLICATION OF PART 2 & 3
PART 7
OVERVIEW OF TECHNIQUES & MEASURES

Figure III. Structure of IEC 61508 Standard

Part III - Annex A of this document describes the general approach adopted in the IEC 61508.

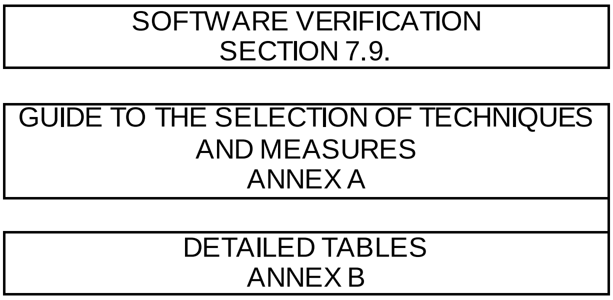
Part 3 of IEC 61508 Standard describes the Software lifecycle activities. The software lifecycle aspects covered by the Standard are shown in Figure IV.

SW SAFETY LIFE CYCLE REQUIREMENTS
GENERAL SECTION 7.1.
SOFTWARE SAFETY REQUIREMENTS SPECIFICATION SECTION 7.2.
SOFTWARE SAFETY VALIDATION PLANNING SECTION 7.3.
SOFTWARE DESIGN AND DEVELOPMENT SECTION 7.4.
PROGRAMMABLE ELECTRONICS INTEGRATION (HARDWARE AND SOFTWARE) SECTION 7.5.
SOFTWARE OPERATION AND MODIFICATION PROCEDURES SECTION 7.6.
SOFTWARE SAFETY VALIDATION SECTION 7.7.
SOFTWARE MODIFICATION SECTION 7.8.

DOCUMENTATION
SECTION 5

SW QUALITY MANAGEMENT SYSTEM
SECTION 6

FUNCTIONAL SAFETY ASSESSMENT
SECTION 8



SOFTWARE VERIFICATION
SECTION 7.9.

GUIDE TO THE SELECTION OF TECHNIQUES AND MEASURES
ANNEX A

DETAILED TABLES
ANNEX B

Figure IV: Scope of IEC 61508-3 Standard.

2.3 ED-12B / DO-178B

The purpose of ED12B/DO178B is to provide aviation airworthiness community with guidance for the production of software for airborne systems and equipment or with a level of confidence in safety that complies with airworthiness requirements.

The document describes the relationship between the system and software lifecycle, and between software development and the system safety assessment processes. However it does not address the system lifecycle, system safety assessment and validation processes.

It is to be noted that no system safety assessment methodology (namely ARP4754 or ED79) was existing when ED12B/DO178B has been written.

Relationships between ED12B/DO178B and other documents developed by the airborne certification community are illustrated in Figure V.

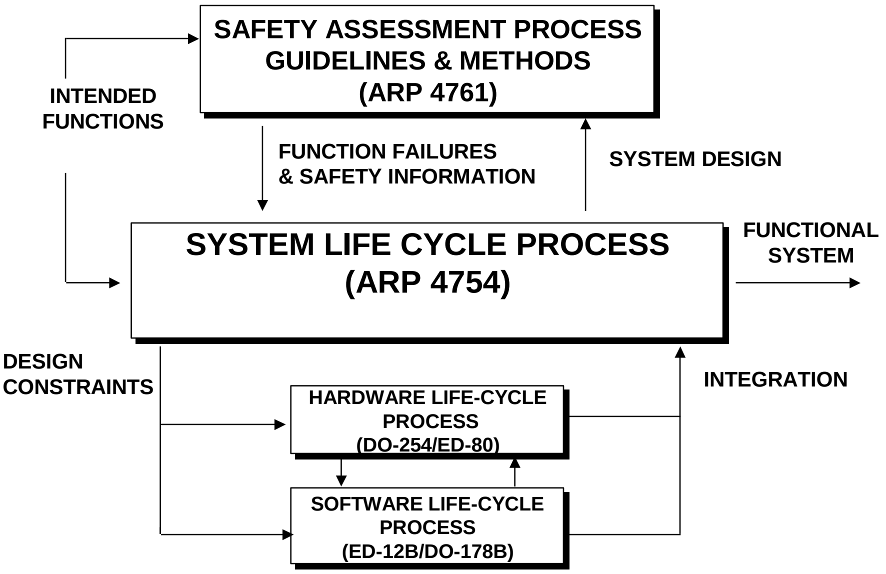


Figure V. Relationships between ED12B/DO178B and ARP documents

The scope of ED12B/DO178B is shown in Figure VI.

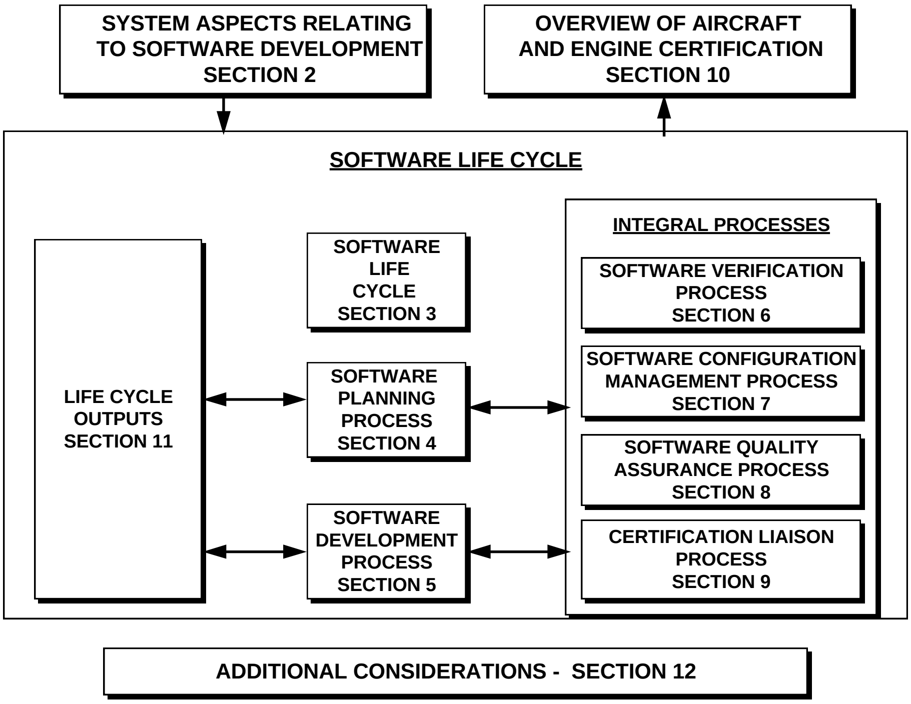

SYSTEM ASPECTS RELATING TO SOFTWARE DEVELOPMENT SECTION 2
OVERVIEW OF AIRCRAFT AND ENGINE CERTIFICATION SECTION 10
SOFTWARE LIFE CYCLE
LIFE CYCLE OUTPUTS SECTION 11
SOFTWARE LIFE CYCLE SECTION 3
SOFTWARE PLANNING PROCESS SECTION 4
SOFTWARE DEVELOPMENT PROCESS SECTION 5
INTEGRAL PROCESSES
SOFTWARE VERIFICATION PROCESS SECTION 6
SOFTWARE CONFIGURATION MANAGEMENT PROCESS SECTION 7
SOFTWARE QUALITY ASSURANCE PROCESS SECTION 8
CERTIFICATION LIAISON PROCESS SECTION 9
ADDITIONAL CONSIDERATIONS - SECTION 12

Figure VI: Scope of ED12B/DO 178B standard.

As the document addresses certification issues, it does not cover operational aspects of software. It does not address contractual relationships between the supplier and purchaser, nor the organisational aspects, competency criteria, and responsibility allocation of the supplier.

The document refers to the concept of software lifecycle but does not prescribe the usage of a specific lifecycle model.

It identifies six processes:

- software planning,
- software development,
- software verification,
- software configuration management,
- software quality assurance and
- software certification.

2.4 ED 109/DO278

This document provides guidelines for the assurance of software contained in non-airborne CNS/ATM systems. ED12B/DO178B, Software Considerations in Airborne Systems and Equipment Certification, defines a set of objectives that are recommended to establish assurance that airborne software has the integrity needed for use in a safety-related application. These objectives have been reviewed, and in some cases, modified for application to non-airborne CNS/ATM systems. This document is intended to be an interpretive guide for the application of ED-12B/DO-178B guidance to non-airborne CNS/ATM systems.

ED109/DO278 applies to software contained in CNS/ATM systems used in ground or space-based applications shown by a system safety assessment process to affect the safety of aircraft occupants or airframe in its operational environment.

The assurance of software resident within the airframe boundaries, including CNS/ATM-related equipment, is addressed by ED12B/DO178B.

A description of the prerequisite safety assessment process is not included in ED109/DO278.

Information on such assessments is available from other industry sources and in related regulatory guidance. Likewise, a complete description of the system life cycle processes, including system validation, as well as CNS/ATM systems approval, is not intended. ED109/DO278 is not intended to be a development standard nor a process document.

2.5 CMMI V1.1

The CMMI is a model developed by the Software Engineering Institute (SEI) of The Carnegie Mellon University. A large number of organizations from industry & US government have been involved in the development of this model.

As stated in the model, the purpose of this model is:

- to provide some guidance for an organisation to improve its processes,
- to serve as a reference to assess process capability/maturity level of the organization, and then to benchmark organizations.

The scope of this model covers the development, acquisition, and maintenance of product or services.

It may be used in various disciplines: System engineering, Software Engineering, Project Management and Supplier Sourcing. The extension to other disciplines (including safety engineering) is possible but requires a specific interpretation of the model to the discipline.

The CMMI is structured in "Process Areas" (PAs) and the maturity is defined in term of levels (from 0 or 1 up to 5).

Level	Project Management PAs	Engineering PAs	Support PAs	Process Management PAs
5 Optimizing			**CAR**: Causal Analysis & Resolution	**OID**: Organizational Innovation & Deployment
4 Quantitatively Managed	**QPM**: Quantitative Project Management			**OPP** : Organizational Process Performance
3 Defined	**IPM**: Integrated Project Management **RSKM**: Risk Management **IT**: Integrated Teaming **ISM**: Integrated Supplier Management	**RD**: Requirements Development **TS**: Technical Solution **PI**: Product Integration **VER**: Verification **VAL**: Validation	**DAR**: Decision Analysis & Resolution **OEI**: Organizational Environment for Integration	**OPF**: Organizational Process Focus **OPD**: Organizational Process Definition **OT**: Organizational Training
2 Managed	**PP**: Project Planning **PMC**: Project Monitoring & Control **SAM**: Supplier Agreement Management	**REQM**: Requirements Management	**MA**: Measurement & Analysis **PPQA**: Process & Product Quality Assurance **CM**: Configuration Management	
1 Initial				

There are two representations of the model: staged or continuous.

The continuous representation is based on an independent levelling of each Process Area, whereas the staged representation is based on "global" levels, each level including both a set of pre-defined PAs and a common level for each of these processes.

For example, using the continuous approach, an organization may be at level 2 for the Project Management PA, and at level 3 for Configuration Management PA, whereas using the staged model, if an organization is at level 3, all the level 3 goals of all the PAs pre-defined as belonging to the Staged Level 3 must be reached.

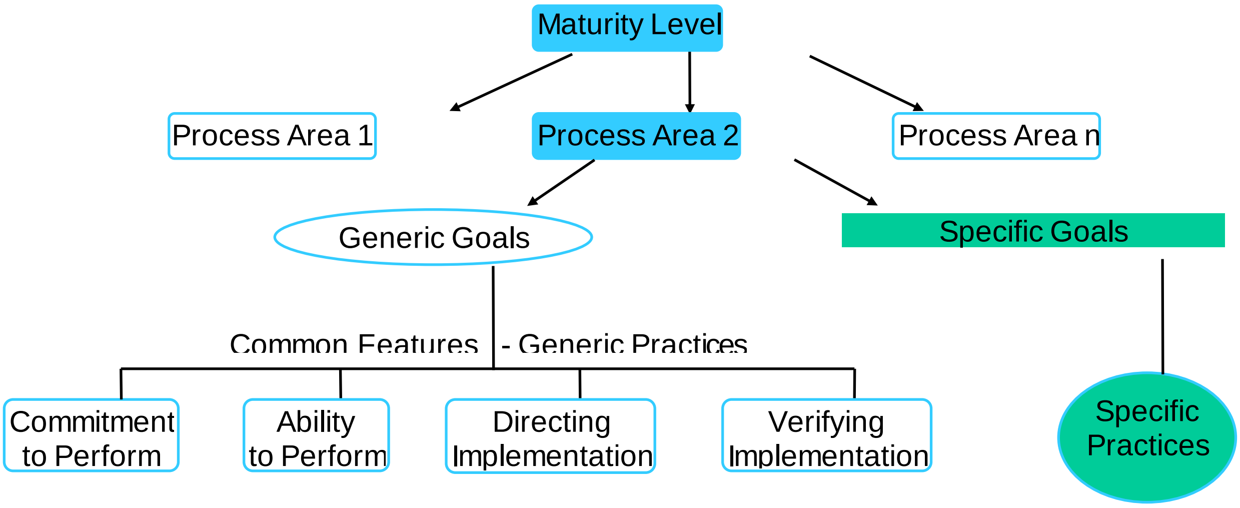
The levels (capability levels) in the continuous representation are the following:

- Incomplete (0),
- Performed (1),
- Managed (2),
- Defined (3),
- Quantitatively Managed (4), and
- Optimizing (5).

The levels (maturity levels) in the staged representation are the following:

- Initial (1),
- Managed (2),
- Defined (3),
- Quantitatively Managed (4), and
- Optimizing (5).

Each Process Area includes a set of "goals". Each goal is supposed to be reached by satisfying a set of requirements called "practices". Goals & practices may be "specific" or "generic". The "specific" goals and practices are dedicated to the Process Areas, whereas the "generic" ones are the same for all the PAs. For example, "Assign responsibility" or "Provide resources" are "generic", i.e. applicable to any process.



This page is intentionally left blank.