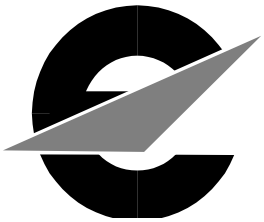
EUROPEAN ORGANISATION FOR THE SAFETY OF AIR NAVIGATION



This document is kept for information purpose only and has been replace in 2009 by ED-153

ANS

SOFTWARE

LIFECYLE

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

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

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	o	Executive Task	o	General Public	þ
Draft	o	Specialist Task	þ	EATMP	o
Proposed Issue	o	Lower Layer Task	o	Restricted	o
Released Issue	þ				

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.

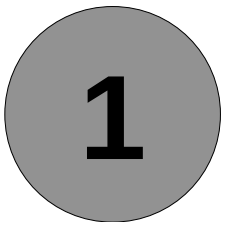
AUTHORITY	NAME AND SIGNATURE	DATE
Chairman of the EATMP Software Task Force	P.MANA	21/12/2005
Chairman of the Safety Assessment Methodology Task Force	P.MANA	21/12/2005
Chairman of the Safety Team	E.MERCKX	21/12/2005
DAP Director	G.PAULSON	21/12/2005

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



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	&	&	ü
Cover - Chapter 2 – SW Standards overview	ü	&	N/A
Part I - Introduction	&	&	ü
Part I - Chapter 1 – Software Safety Assurance System	&	&	&
Part I - Chapter 2 – Primary lifecycle	&	&	ü
Part I - Chapter 3 – Supporting Lifecycle	&	&	ü
Part I - Chapter 4 – Organisational Lifecycle	ü	&	ü
Part I - Chapter 5 – Additional Lifecycle	&	&	ü
Part II -Introduction	ü	&	N/A
Part II – Chapter 1 – ED12B/DO178B	ü*	&*	ü*
Part II - Chapter 2 – IEC61508	ü*	&*	ü*
Part II - Chapter 3 – ISO/IEC 12207	ü*	&*	ü*
Part II - Chapter 4 – ED109/DO278	ü*	&*	ü*
Part II – Chapter 5 – CMMi	ü*	&*	ü*

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

& : Detailed knowledge;

Ü : 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)

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.

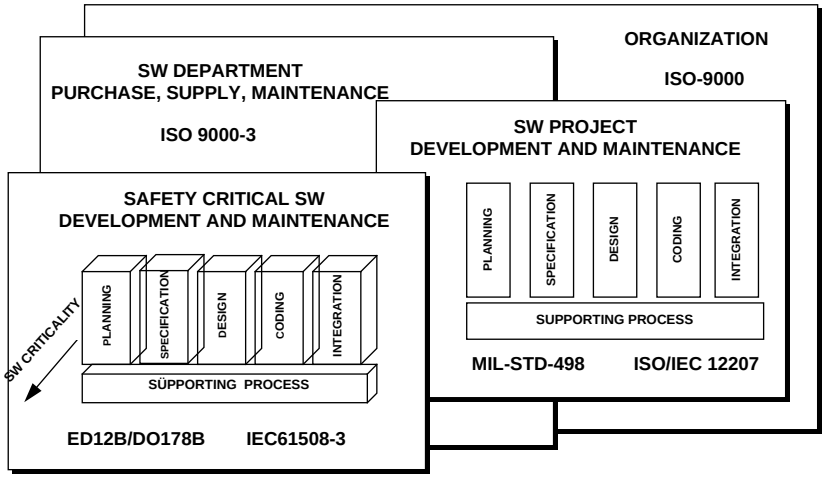


Figure I. Scope and Interrelationships of Standards

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.

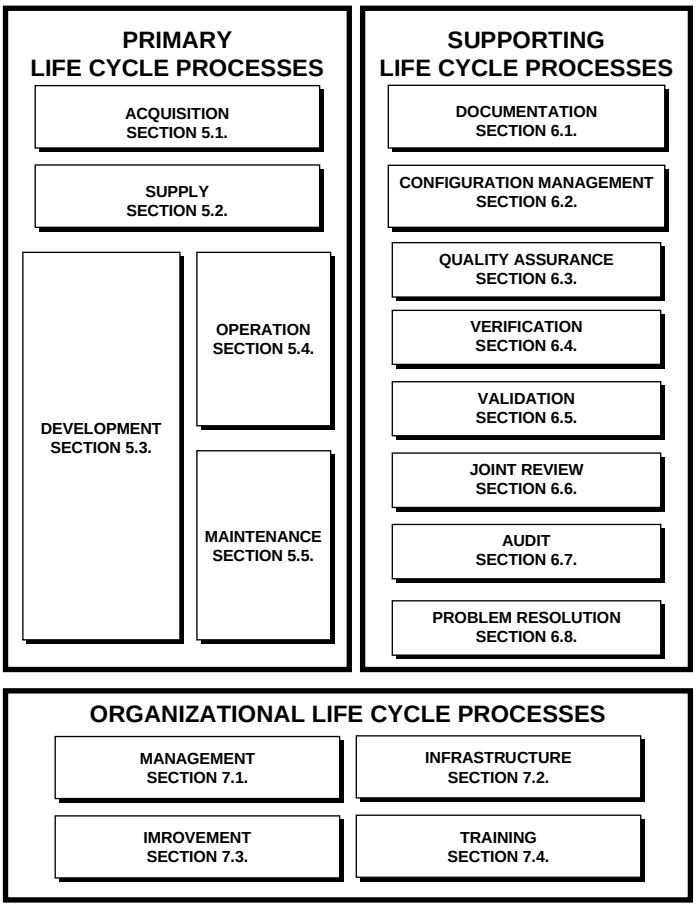


Figure II. Scope of ISO/IEC 12207 Standard

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.

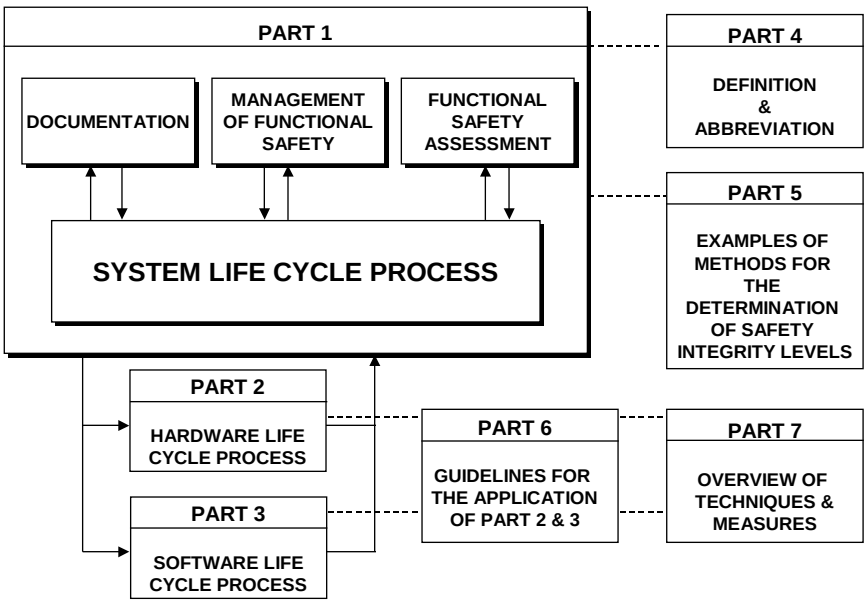


Figure III. Structure of IEC 61508 Standard

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	DOCUMENTATION SECTION 5
GENERAL SECTION 7.1.	
SOFTWARE SAFETY REQUIREMENTS SPECIFICATION SECTION 7.2.	SW QUALITY MANAGEMENT SYSTEM SECTION 6
SOFTWARE SAFETY VALIDATION PLANNING SECTION 7.3.	
SOFTWARE DESIGN AND DEVELOPMENT SECTION 7.4.	FUNCTIONAL SAFETY ASSESSMENT SECTION 8
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.	

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.

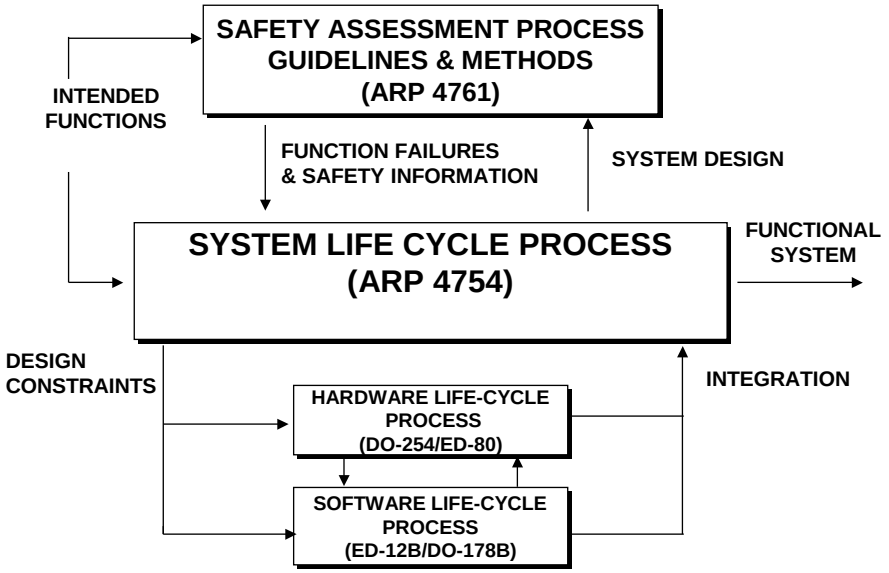


Figure V. Relationships between ED12B/DO178B and ARP documents

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

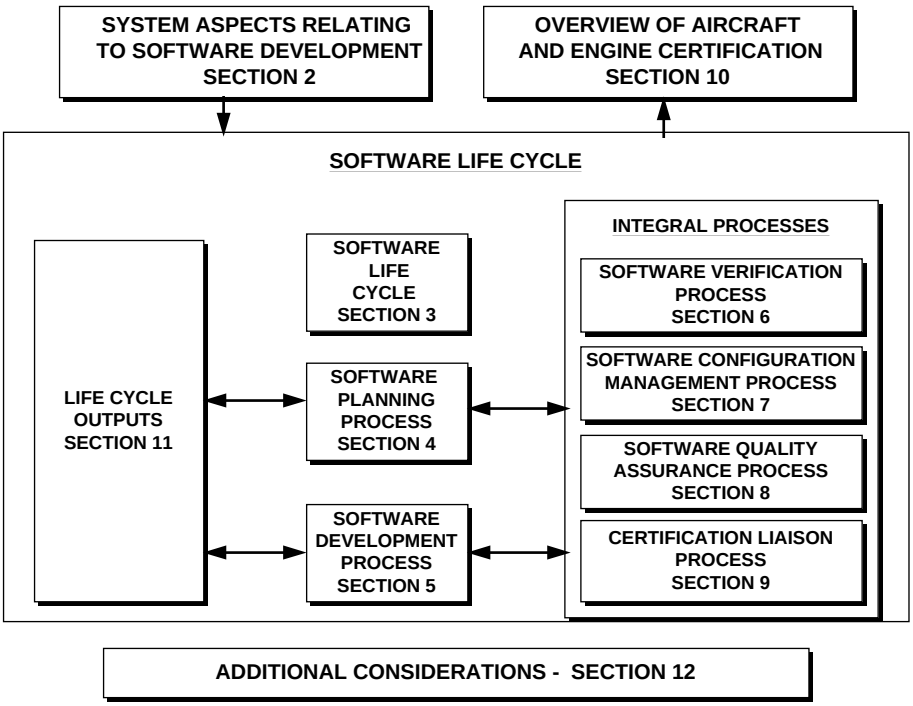


Figure VI: Scope of ED12B/DO 178B standard.

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.

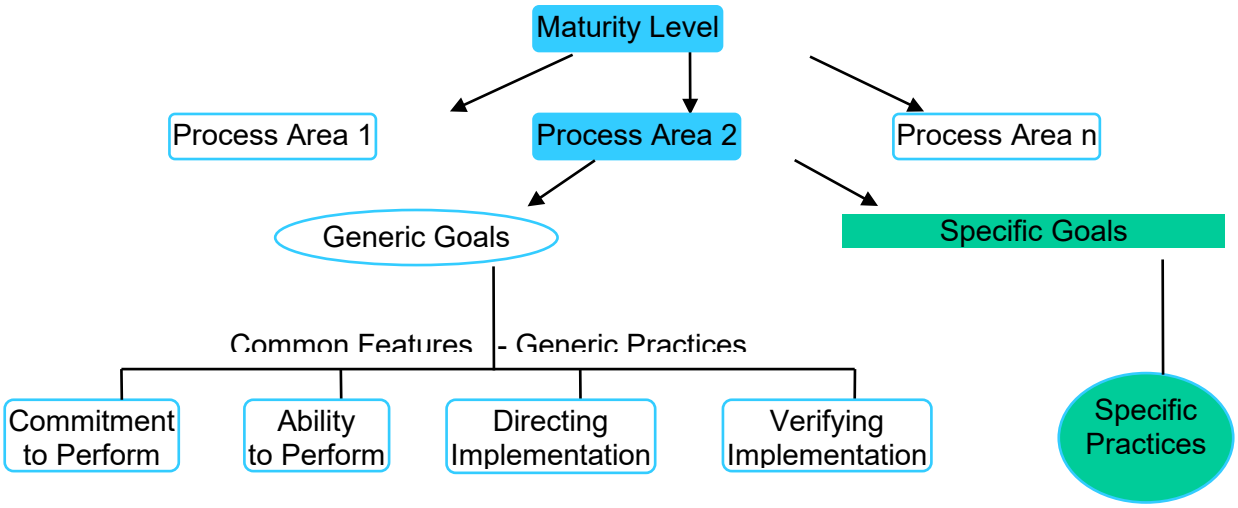
The levels (capability levels) in the continuous representation are the following:

- 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.

PART I

ANS SOFTWARE

LIFECYCLE DEFINITION

This page is intentionally left blank.

TABLE OF CONTENTS

PART I – ANS SOFTWARE LIFECYCLE DEFINITION

CHAPTER 5

INTRODUCTION

1 PURPOSE OF PART I

The main purpose of this part of this document is to define a recommended ANS software lifecycle.

This ANS software lifecycle is reusing IEC/ISO12207 processes structure, because this standard has the widest coverage (from definition till decommissioning) of ANS needs. However, this report does not intend at all to promote any standard, neither to state that any standard fits best ANS needs (even if IEC/ISO 12207 has been used as a processes structure basis).

The purposes of this part of this document are the following:

- To propose a software lifecycle tailored to ANS

- To provide a traceability matrix. For each listed objective a reference is given to the standard paragraph, which covers this objective. This traceability allows having access directly to the exact wording of a standard, for those who want to assess more accurately how a standard covers an objective.

- To provide a compatibility matrix between standards, which will allow identifying commonalities and differences between standards. So, suppliers, ATS providers, regulators and any other organisation or group will be able to evaluate characteristics of a system or equipment integrating software without requiring the use of the standard recommended by its organisation. This compatibility matrix will allow every actors to “speak the same language” when talking about software standards.

- To provide a synthetic overview of objectives and activities coverage by each standard. Tables give at a first glance a general view if objectives are implemented or not using the following symbols:

  - ▪ (means fully covered)
  - **P** (partially covered)
  - blank (not covered

  ED109/DO278 traceability is including specific considerations due to the fact that ED109/DO278 is not a stand-alone document[1] as it is based on ED-12B/ DO-178B.

- To identify area of improvement of existing standards, especially because of ANS particularities.
- To identify objectives which have to be modified for ANS purposes.

The set of ANS software lifecycle processes is divided into:

- A software safety assurance system,
- Five ***primary*** processes,
- Eight ***supporting*** processes,
- Four ***organisational*** processes,
- Additional ANS software lifecycle objectives.

Some process descriptions are printed using *ITALIC* characters because they are copied from ISO/IEC 12207.

Specific interpretation & notation regarding <u>mapping</u> to CMMI model:

The CMMI is designed for any type of development or services, and there is no specific safety “amplification” for safety-constrained development or services. So, rather than pure traceability, the following part of tables related to the CMMI identifies mapping or relationship (full, partial or none). ”Mapping” stands for “same Activity, but not systematically the same point of view nor the same level of detail”, where “traceability” stands for

[1] See ED109 chapter 1.3

"equivalent level of requirement (same coverage, same level of detail)". Refer also to Part II section 5, §1.1 for more details on relationships between ANS Life cycle philosophy & CMMI philosophy.

The detailed used references are the following (where XXX is the acronym of a CMMI Process Area)

- XXX à mapping to the global Process Area XXX
- XXX1 (respectively XXX 1, 2) à mapping to the set of practices related to the goal 1 (respectively to the set of goals 1 and 2) of the Process Area XXX
- XXX 2.1 (respectively XXX 1.1, 2.1, 3.2) à mapping to the Specific Practice 2.1 (respectively to the set of Specific Practices "1.1, 2.1 & 3.2") of the Process Area XXX
- GP 2.4 (respectively GP 2.4, 2.7) à mapping to the Generic Practice "GP2.4" (respectively to the set of Generic Practices "2.4, 2.7") for the set of Process Areas
- XXX GP 2.1 (respectively GP 2.1, 2.7) à mapping to the Generic Practice « 2.1 » (respectively to the set of Generic Practices "2.1, 2.7") of the Process Area XXX

2 DEFINITIONS

Adaptation Data	Data used to customise elements of the Air Traffic Management System for their designated purpose (See note1).
ANS	Air Navigation System
Approval	A means by which an authorised body gives formal recognition that a product, process, service, or operation conforms to applicable requirements. *Note: For example, approval is a generic term to refer to certification, commissioning, qualification, initial operational capability, etc.*
Approval Authority	The relevant body responsible for the approval in accordance with applicable approval requirements.
Configuration data	Data that configures a generic software system to a particular instance of its use (for example, data for flight data processing system for a particular airspace, by setting the positions of airways, reporting points, navigation aids, airports and other elements important to air navigation)

Documentation	Set of documentation items related to a life cycle phase and necessary as inputs to perform other life cycle activities
HMI	Human Machine Interface
Software	Computer programs and corresponding configuration data, including non-developmental software (e.g. proprietary software, Commercial Off The Shelf (COTS) software, re-used software, etc.), but excluding electronic items such as application specific integrated circuits, programmable gate arrays or solid-state logic controllers.
Software Component	A distinct part of a Software. Software component may be further decomposed into other Software Components and Software Units.
Software Failure	The inability of software to perform a required function correctly.
Software Unit	An element specified in the design of a Software Component that is separately testable.
Supplier	A person or organisation seeking approval from the Approval Authority.
System	An Air Navigation System is composed of People, Procedures and Equipment (Software, Hardware and HMI)
Validation	Confirmation by examination and provision of objective evidence that the particular requirements for a specific intended use are fulfilled (usually used for internal validation of the design).
Verification	Confirmation by examination of evidence that a product, process or service fulfils specified requirements.

Note 1: Extended definition of adaptation data

Adaptation data is utilized to customize elements of the CNS/ATM system for its designated purpose at a specific location. These systems are often configured to accommodate site-specific characteristics. These site dependencies are developed into sets of adaptation data. Adaptation data includes:

- Data that configures the software for a given geographical site, and
- Data that configures a workstation to the preferences and/or functions of an operator.

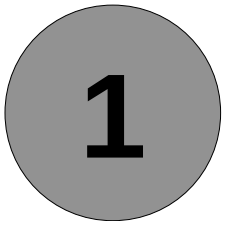
Examples include, but are not limited to:

a. Geographical Data – latitude and longitude of a radar site.
b. Environmental Data – operator selectable data to provide their specific preferences.
c. Airspace Data – sector-specific data.
d. Procedures – operational customization to provide the desired operational role.

Adaptation data may take the form of changes to either database parameters or take the form of pre-programmed options. In some cases, adaptation data involves re-linking the

code to include different libraries. Note that this should not be confused with recompilation in which a completely new version of the code is generated.

Adaptation data should be developed to the same assurance level as the one of the code that processes them.



SOFTWARE SAFETY ASSURANCE SYSTEM

Software Safety Assurance System encompasses the following tasks:

1) Software Safety Assurance System Objectives
2) Software Assurance Level
3) Software Safety Assessment
    1) Software Safety Assessment Initiation
    2) Software Safety Assessment Planning
    3) Software Safety Requirements Specification
    4) Software Safety Assessment Validation, Verification & Process Assurance
    5) Software Safety Assessment Completion

The implementation of the Software Safety Assurance System is the responsibility of the ANSP (Air Navigation Service Provider).

1 SOFTWARE SAFETY ASSURANCE SYSTEM OBJECTIVES

The following table lists the recommended objectives to implement a Software Safety Assurance System.

N°	Obj	Activity Title	Activity	ISO/IEC 12207	ED109	ED-12B/ DO 178B	IEC 61508	CMMI
1	3.0.1	Implementation	A Software Safety Assurance System should be defined, implemented and documented.		P	P	P	
2	3.0.2	Requirements Correctness and Completeness	The software requirements correctly state what is required from the software by the system safety assessment	P (Ref: 5.3.4)	. (Ref: 3.2 – Table A-2 (lines 1,2) Table A-3 (lines 1, 2)	P (Ref: 5.1)	. (Ref: 7.2.2)	. (Ref: RD 1.1, 1.2, 2.1)
3	3.0.3	Requirements Traceability Assurance	All software requirements are traced to the level required by the SW AL	P (Ref: 5.3.4.2; 5.3.5.6; 5.3.6.7; 5.3.7.5)	. (Ref: A3.6, A4.6, A5.6)	. (Ref: 5.5)	P	P (Ref: ReqM 1.4)
4	3.0.4	Unintended Functions	The software implementation should contain no functions, which adversely affect safety or whose effect is not consistent with the safety analysis.		. (Ref: 3.6 Table A-5 line 1)	P (Ref: 6.3.4.a)	P (Ref: 7.4.7.2)	
5	3.0.5	SW AL Allocation	Any ANS software intended for operational use is allocated a Software Assurance Level (SW AL).		. (Ref: Appendix B.4	. (Ref: 2.2.2, 2.2.3)	. (Ref: 7.5.2, 7.6.2)	
6	3.0.6	Requirements Satisfaction Assurance	The ANS software satisfies its software requirements with a level of confidence which is set according to the SW AL allocated during PSSA		. (Ref: 2.1)	. (Ref: 5.1)	. (Ref: 7.2)	
7	3.0.7	Configuration Management Assurance	Assurances should be at all times derived from a known executable version of the software, a known range of configuration data, and a known set of software products and descriptions that have been used in the production of that version.	. (Ref : 6.2)	. (Ref: 3.8 Table A-8)	. (Ref: 7)	. (Ref: 6.2.3)	. (Ref: CM)
8	3.0.8	Assurance Rigour Objective	The assurances and the levelling of assurances should give sufficient confidence that the ANS software can be operated, as a minimum, acceptably		. (Ref: 2.1)	. (Ref: 2.1, 9 & 11.20)	. (Ref: Part 1 – 7.4.2)	

N°	Obj	Activity Title	Activity	ISO/IEC 12207	ED109	ED-12B/ DO 178B	IEC 61508	CMMI
			safely.					
9	3.0.9	Assurance Rigour Criteria (Obj	The variation in rigour of the assurances per software assurance levels should be specified with the following criteria: · required to be achieved with independence, · required to be achieved, · not required.		. (Ref: Chap 3)	. (Ref: Appendix A)	. (Ref: Appendix A)	
10	3.0.10	SW AL Assurance	Assurance should provide confidence that SW AL is achieved.		. (Ref: 3.10 Table A-10 ; 5.1)	. (Ref: 9 & 11.20)	. (Ref: 6.2.2)	
11	3.0.11	SW AL Monitoring	Assurance should be given that once in operation the software meets its SW AL through monitoring. Feedback of ATM software experience should be used to confirm that the Software Safety Assurance System and the assignment of assurance levels is appropriate. For this purpose, the effects resulting from any reported software malfunction or failure from ATM operational experience, should be assessed in respect of their mapping to SWAL definition (See Chapter 2 of this document) . (Reported Software malfunction or failure are output of the ATM occurrence reporting system as part of the ATMSP Safety Management System).		P (Ref: 4.1.6.3)			
12	3.0.12	Software Modifications	Any change to the software should lead first to re-assess the safety impact of such a change on the system and then on the SWAL allocated to this software.		P (Ref: 4.1.4.2)		. (Ref: 7. 8)	
13	3.0.13	COTS	The same level of confidence, through any means chosen and agreed with the Designated Authority, should be provided with the same software assurance level for developmental and non-developmental ATM software (e.g. Commercial Off The Shelf software, etc).		. (Ref: 4.2)			
14	3.0.14	Independence	ATM software components that cannot be shown to be independent of one another should be allocated the software		. (Ref: Chap	. (Ref: Chap	. (Ref: Chap	

N°	Obj	Activity Title	Activity	ISO/IEC 12207	ED109	ED-12B/ DO 178B	IEC 61508	CMMI
			assurance level of the most critical of the dependent components.					
15	3.0.15	All on-line aspects of SW operational changes	The Software Safety Assurance System should deal specifically with software related aspects, including all on-line software operational changes (such as cutover/hot swapping).					

Note: IEC12207, ED12B/DO178B, ED109/DO278 and IEC61508 consider a system as being hardware and software. The Safety Assessment Methodology (SAM), which this document is part of , defines a system as composed of people, procedure and equipment (software, hardware and Human Machine interface (HMI)). Consequently, the people and procedure aspects of a system are not taken into account by these 4 standards.

2 SOFTWARE ASSURANCE LEVEL

See “Recommendations for ANS SW” V1.0 Chapter 2 or SAM-PSSA Chapter 3 Guidance Material A V2.0 (§2.4.2).

3 SOFTWARE SAFETY ASSESSMENT

The FHA is conducted at a functional level, so the software architecture and design are not known at that stage. Therefore FHA does not address hardware and software safety requirements and assurance level.

However, for a system including safety-related software there is a need to analyse the software (function and/or architecture and design) in order to gain assurance that the set of hazards identified during the FHA is correct and complete.

To achieve this certain sub-processes and tasks may be applicable for re-assessing the FHA output at software level. Examples of such are:

- Identification of software failures which confirms the results of the original FHA.
- Identification of software failures (due to e.g. software faults or interface errors that cannot be found at the functional or operational level) which could result in the occurrence of new hazards not identified at the FHA level.

The PSSA intends to identify a system architecture that will meet the safety objectives and apportions these safety objectives into safety requirements to the system elements (people, procedure and equipment (hardware, Software, HMI).

Safety requirements for software are mainly stated as Software Assurance Level.

Anyhow, system safety assessment process remains iterative, consequently software safety assessment, which is part of the SSA (System Safety Assessment. The third step of the Safety Assessment Methodology), has to confirm, verify and complete (if necessary) the assumptions and outcome of the previous steps.

3.1 SOFWARE SAFETY ASSESSMENT INITIATION

FHA (Functional Hazard Assessment) assumptions and output should be confirmed as far as software can impact them.

PSSA (Preliminary System Safety Assessment) assumptions and output should be confirmed as far as software can impact them.

N°	Obj	Activity Title	Activity	ISO/IEC 12207	ED109	ED-12B/ DO 178B	IEC 61508	CMMI
1	3.1.1	System Description	The system description should be suitable to the safety objectives and requirements by performing the following activities: a) The Software purpose should be defined. b) Operational scenarios should be defined (especially HMI: Operator Handbook should define the mode of operation and the human-machine interface). c) The Software/System functions and their relationships should be defined. d) Software boundaries should be defined (operational, time, ..) e) Software external interfaces should be described		. (Ref: 2.2)	. (Ref: 2.1)	. (Ref: I-7.2.1)	P [Ref: a) RD 1.1 b) RD 3.1, TS 1.2 c) RD 3.2 e) RD 2.3, TS 2.3]
2	3.1.2	Operational Environment	Develop a level of understanding of the Software and its environment (physical, operational, control functions, legislative etc) sufficient to enable the other safety lifecycle tasks to be satisfactorily carried out.		P (Ref: 2.2)	P (Ref: 2.1.1)	. (Ref: I-7.2.1)	P (Ref: RD 1.1)
3	3.1.3	Regulatory Framework	Safety regulatory objectives and requirements should be defined.		. (Ref: 3.10 Table A-10 line 2 - 5.1)	. (Ref: 2.1.1, 9, 10)	. (Ref: I-7.2.2.4)	
4	3.1.4	Applicable Standards	Safety standards applicable to the Software should be defined.		.	.	.	
5	3.1.5	System FHA & PSSA Output	The result of the system FHA (Functional Hazard Assessment) or PSSA (Preliminary System Safety Assessment) should be made available.		P (Ref: 2.2)	P (Ref: 2.1.1)	P (Ref: I-7)	

N°	Obj	Activity Title	Activity	ISO/IEC 12207	ED109	ED-12B/ DO 178B	IEC 61508	CMMI
			Results of similar system safety assessment should be used as a reference.					

3.2. SOFTWARE SAFETY ASSESSMENT PLANNING

N°	Obj	Activity Title	Activity	ISO/IEC 12207	ED109	ED-12B/ DO 178B	IEC 61508	CMMI
1	3.2.1	Software Safety Assessment Approach	The overall approach for the Software Safety Assessment across Software Lifecycle should be defined and documented.		. (Ref: §5.1)	. (Ref: 11.1)	. (Ref: 8)	
2	3.2.2	Software Safety Assessment Plan	A plan describing the software safety assessment steps should be produced (e.g. approach, relations between safety assessment and software lifecycle, deliverables (content and s-date), relations with software/system major milestones, project risk management due to safety issues, responsibilities, persons, organisations, risk classification scheme, safety objectives definition approach, hazard identification methods, safety assurance activities, schedule, resource)		. (Ref: 5.1 - 3.10 Table A-10)	P (Ref: 11)	P (Ref: I-7.8)	
3	3.2.3	Software Safety Assessment Plan Review	The Software Safety Assessment plan should be reviewed and commented for suitability and approval.		. (Ref: 5.1 - 3.10 Table A-10)	. (Ref: 9, 10)		
4	3.2.4	Software Safety Assessment Plan Dissemination	The Software Safety Assessment plan should be made available to the interested parties.		P (Ref: 5.1)	. (Ref: 9, 10)		

3.3. SOFTWARE REQUIREMENTS SPECIFICATION

N°	Obj	Activity Title	Activity	ISO/IEC 12207	ED109	ED-12B/ DO 178B	IEC 61508	CMMI
1	3.3.1	Failures Identification	Failures should be identified by considering various ways Software can fail and by considering the sequence of events that lead to the occurrence of the failure. The list of single or multiple failures should be drawn. The combination of failures should be identified.			P (Ref: 2.2)	. (Ref: I-7.4)	
2	3.3.2	Failure Effects	The effects of failure occurrence should be evaluated. The hazards associated with failure occurrences should be identified.			P (Ref: 2.2.1)	. (Ref: I-7.4)	
3	3.3.3	Assessment of Risk	The purpose of this Activity Title is to classify hazards according to the severity of their consequences.			P (Ref: 2.2.1)	. (Ref: I-7.5)	
4	3.3.4	Software Requirements Setting	a) For each function and combination of functions to which software participates, · 1- Refine the functional breakdown. · 2- Evaluate system architecture(s) · 3- Identify risk mitigation means. · 4- Apportion Safety Objectives in to Safety Requirements. · 5- Balance Safety Requirements. b) Software Requirements should be compliant with the System Safety Objectives. (System Safety Objectives specify the maximum acceptable frequency of occurrence of a hazard).			P (Ref: 2.2.1)	. (Ref: I-7.6)	P [Ref: a.1) RD 2.1, 2.2 a.2) TS 2.1, Ver 1.1, 2.2, 2.3]
5	3.3.5	SW Allocation	A SW AL should be allocated to the software			P[2] (Ref: 2.2.3)	P[2] (Ref: I-7.6.1)	

Note: Column ED-12B/ DO178B- These tasks are identified as partially met by ED12B/DO178B because section 2 of this document compensates the lack of system safety standard namely ARP4754/4761, which was elaborated after ED12B/DO178B.

[2] P (Partially) allocation process is not directly applicable for ATM

3.4 SOFTWARE SAFETY ASSESSMENT VALIDATION, VERIFICATION AND PROCESS ASSURANCE

N°	Obj	Activity Title	Activity	ISO/IEC 12207	ED109	ED-12B/ DO 178B	IEC 61508	CMMI
1	3.4.1	Software Safety Assessment Validation	a) Ensure that Software Requirements are complete and correct. b) Traceability, review and ***tracking*** of software safety requirements should be performed.	P (Ref: 6.4; 6.5)	. (Ref: 3.3 Table A-3 lines 1, 2; 3.4 Table A-4 lines 1, 2, 6)			P [Ref: a) RD 3.3, 3.4, 3.5 Ver 2.1, 2.2, 2.3 b) -]
2	3.4.2	Software Safety Assessment Verification	Software Requirements should be consistent with the outcomes of the hazard effects and hazards description and classification.		. (Ref: 2.1)	. (Ref: 2.2.2)		
3	3.4.3	Software Safety Assessment Process Assurance	Every step of the software Safety Assessment performance should be checked.	P (Ref: 6.4)	. (Ref: 3.9 Table A-9)	. (Ref: 8)		

3.5 SOFTWARE SAFETY ASSESSMENT COMPLETION

N°	Obj	Activity Title	Activity	ISO/IEC 12207	ED109	ED-12B/ DO 178B	IEC 61508	CMMI
1	3.5.1	Document Software Safety Assessment Process Results	The Software Safety Assessment process results should be documented.	P (Ref: 6.1)	P (Ref: 5)	P (Ref: 11, Annex A)	P (Ref: I-7.2.2.6,I-7.3.2.5, I-7.4.2.11)	P
2	3.5.2	Software Safety Assessment Documentation Configuration Management	Software Safety Assessment documentation should be put under configuration management.	P (Ref: 6.2)	P (Ref: 3.8 - 4.1.7)	P (Ref: 7.3, Annex A)	P (Ref: I-7.4.2.12)	P (Ref: CM 1.1)
3	3.5.3	Software Safety Assessment Documentation Dissemination	Software Safety Assessment documentation should be disseminated to interested parties.		P (Ref: 5.1 - 3.10 Table A-10)	P (Ref: 9, 10)		P (Ref: GP2.7)

PRIMARY LIFECYCLE PROCESSES

Primary lifecycle processes consist of:

1) Acquisition process;

2) Supply process;

3) Development process;

4) Operation process;

5) Maintenance process.

The objectives and tasks in a primary process are the responsibility of the organisation initiating and performing that process. Depending on the lifecycle phase, different organisations may be responsible for performing a process. Each organisation ensures that the process is in existence and functional.

1 ACQUISITION PROCESS

The Acquisition Process contains the objectives *and tasks of the acquirer. The process begins with the definition of the need to acquire a system, software product or software service. The process continues with the preparation and issue of a request for proposal, selection of a supplier, and management of the acquisition process through to the acceptance of the system, software product or software service.*

The individual organisation having the need may be called the owner. The owner may contract any or all of the acquisition activities to an agent who will in turn conduct these activities according to the Acquisition Process. The acquirer in this sub-clause may be the owner or the agent.

Note: Acquisition process does not relate to business aspects of acquisition, but only to safety and quality aspects of it.

N°	Obj	Activity Title	Activity	ISO/IEC 12207	ED109	ED-12B/ DO 178B	IEC 61508	CMMI
1	4.1.1	Initiation	a) The acquirer begins the acquisition process by describing a concept or a need to acquire, develop, or enhance a system, software product or software service. b) The acquirer will define and analyse the system requirements. c) The system requirements should include business, organisational and user as well as safety, security, and other criticality requirements along with related design, testing, and compliance standards and procedures. d) The acquirer should prepare, document and execute an acquisition plan.	. (Ref: 5.1.1)				. [Ref: all) SAM 2.1; a) TS 2.4; b,c)RD1.2,2.1, ReqM1.4; d) SAM 1.1, GP 2.2, 3.1; ISM GP 2.2, 3.1]
2	4.1.2	Functional Hazard Assessment	The acquirer should determine how safe does the system needs to be.				P (Ref: I-7.2, I-7.3, I-7.4, I-7.5)	
3	4.1.3	Preliminary System Safety Assessment	The acquirer should determine (during the System Design phase) whether the proposed architecture is expected to achieve the Safety		P (Ref: 2)	P (Ref: 2)	P (Ref: I-7.6)	

N°	Obj	Activity Title	Activity	ISO/IEC 12207	ED109	ED-12B/ DO 178B	IEC 61508	CMMI
			Objectives defined by the FHA. (Note1)					
4	4.1.4	Request For Tender	The acquirer should determine which processes, activities, and tasks of this International Standard are appropriate for the project and should tailor them accordingly.	. (Ref: 5.1.2)				P (Ref: SAM 1.2)
5	4.1.5	Contract preparation and update	The acquirer should establish a procedure for supplier selection including proposal evaluation criteria and requirements compliance weighting.	. (Ref: 5.1.3)				. (Ref: SAM 1.2)
6	4.1.6	Supplier monitoring	The acquirer will monitor the supplier's activities.	. (Ref: 5.1.4)				. (Ref: SAM 2.2)
7	4.1.7	Acceptance and completion	The acquirer should prepare for acceptance based on the defined acceptance strategy and criteria. The preparation of test cases, test data, test procedures, and test environment should be included. The extent of supplier involvement should be defined. The acquirer will conduct acceptance review and acceptance testing of the deliverable software product or service and will accept it from the supplier when all acceptance conditions are satisfied.	. (Ref: 5.1.5)				. (Ref: SAM 2.3)

Note 1: To simplify and as the purpose of this document is to describe the objectives related to the software lifecycle, it has been considered that the acquirer performs the PSSA (Preliminary System Safety Assessment). Even if in a real project this step may be performed in relation with the system supplier, however it remains the acquirer responsibility to validate and accept it. As this document focuses on the software-related objectives, the main purpose of the PSSA is to allocate an Assurance Level to the software, which has to remain under the Acquirer ultimate responsibility (at least by validating it, when not allocating it).

Note 2: This document intends to address the software aspects of SSA (System Safety Assessment: the third step of the Safety Assessment Methodology).

2 SUPPLY PROCESS

The Supply Process contains the objectives *and tasks of the supplier. The process may be initiated either by a decision to prepare a proposal to answer an acquirer's request for proposal or by signing and entering into a contract with the acquirer to provide the system, software product or software service. The process continues with the determination of procedures and resources needed to manage and assure the project, including development of project plans and execution of the plans through delivery of the system, software product or software service to the acquirer.*

N°	Obj	Activity Title	Activity	ISO/IEC 12207	ED109	ED-12B/ DO 178B	IEC 61508	CMMI
1	4.2.1	Initiation	The supplier conducts a review of requirements included in the request for proposal taking into account organisational policies and other regulations.	. (Ref: 5.2.1)				. (Ref: RD 1.1, 1.2, 3.3, 3.4)
2	4.2.2	Preparation of response	The supplier should define and prepare a proposal in response to the request for proposal, including its recommended tailoring of this International Standard.	. (Ref: 5.2.2)				. (Ref: ReqM 1.1)
3	4.2.3	Contract	The supplier should negotiate and enter into a contract with the acquirer organisation to provide the software product or service.	. (Ref: 5.2.3				. (Ref: ReqM 1.2)
4	4.2.4	Planning	The supplier should define or select a software lifecycle model appropriate to the scope, magnitude, and complexity of the project. The processes, activities, and tasks of this International Standard should be selected and mapped onto the lifecycle model. The supplier should develop and document project management plan(s). NOTE 1	. (Ref: 5.2.4)	P (Ref: 3.1)	P (Ref: 4)	P (Ref: I-6)	. (Ref: PP 1.3, 2.7)
5	4.2.5	Execution & control	The supplier should implement and execute the project management plan(s). The supplier should monitor and control the progress and the quality of the software products or services of the project throughout the contracted lifecycle.	. (Ref: 5.2.5)	. (Ref: 3.9, Table A-9)	P (Ref: 4.6)	P (Ref: I-6.2.2)	. (Ref: PMC 1 SAM 2.2)
6	4.2.6	Review & evaluation	a) The supplier should co-ordinate contract review activities, interfaces, and communication with the acquirer's organisation. b) The supplier should perform quality assurance activities.	. (Ref: 5.2.5)			P (Ref: I-6.2)	. [Ref: a) PMC 1.5, 1.6, 1.7 b) PPQA]

N°	Obj	Activity Title	Activity	ISO/IEC 12207	ED109	ED-12B/ DO 178B	IEC 61508	CMMI
7	4.2.6	Software Acceptance Definition	The developer should support the acquirer's acceptance review and testing of the software product. Acceptance review and testing should consider the results of the Joint Reviews, Audits, Software Qualification Testing, and System Qualification Testing (if performed). The results of the acceptance review and testing should be documented.	. (Ref: 5.3.13.1)				. (Ref: PI 3.4 Ver Val)
8	4.2.6	SW Acceptance Support	The developer should support the acquirer's acceptance review and testing of the software product.	. (Ref: 5.3.13)				
9	4.2.7	Software Product Delivery	The developer should complete and deliver the software product as specified in the contract.	. (Ref: 5.3.13.2)				. (Ref: PI 3.4)
10	4.2.7	Delivery & completion	The supplier should deliver and provide assistance to the acquirer in support of the delivered software product or service as specified in the contract.	. (Ref: 5.2.6)				. (Ref: SAM 2.4)
11	4.2.7	Support to Acquirer	The developer should provide initial and continuing training and support to the acquirer as specified in the contract.	. (Ref: 5.3.13.3)				. (Ref: SAM 2.4)

NOTE 1: Since ANS systems may operate continuously and may have been in operation for many years, the software lifecycle plans for these systems should include processes for software changes, technology upgrades, etc., specifically with respect to safety issues.

3 DEVELOPMENT PROCESS

The Development Process contains the objectives *and tasks of the developer. The process contains the* objectives *for requirements analysis, design, coding, integration, testing, and installation and acceptance related to software products. It may contain system-related* objectives *if stipulated in the contract. The developer performs or supports the activities in this process in accordance with the contract.*

Note: System related objectives are part of FHA & PSSA steps of the System Safety Assessment. However as these processes are interacting in an iterative way, system requirements, architecture, integration, … are to be reassessed to be confirmed and validated when software activities are performed. That is why these system objectives are listed in the software related one (See Part I - Chapter 1 §3).

N°	Obj	Activity Title	Activity	ISO/IEC 12207	ED109	ED-12B/ DO 178B	IEC 61508	CMMI
1	4.3.3	Process Implementation	a) The developer should define or select a software lifecycle model appropriate to the scope, magnitude, and complexity of the project. b) The developer should select, tailor, and use those standards, methods, tools, and computer programming languages. c) The developer should develop plans for conducting the activities of the development process.	. (Ref: 5.3.1)	. (Ref: 3.1 Table A-1 lines 1 to 7 for COTS; 4.1.9 Table A-10 lines 1, 2, 3)	. (Ref: 3, 4, 11.2)	P (Ref: 7.1.2.1, 7.4.1.3, Part I-7)	. [Ref: a) PP 1.3 b) GP 2.2, 3.1, PP2.4 c) PP 2]
2	4.3.3	Software Development Plan	This plan is used to determine the proposed software lifecycle commensurate with the rigour required for the level of software being developed.		. (Ref: 3.1 Table A-1, Lines 1, 5, 7; 4.1.4; 4.1.9 line 3)	. (Ref: 11.1, 11.2)		. (Ref: PP 1.1, 1.3)
3	4.3.1	System Requirements Analysis	The system requirements specification should describe: functions and capabilities of the system; business, organisational and user requirements; safety, security, human-factors engineering (ergonomics), interface, operations, and maintenance requirements; design constraints and qualification requirements.	. (Ref: 5.3.2)	P (Ref: 2.2)	. (Ref: 2.1)	. (Ref: Part I-7.6, Part II-7.2 II-7.9)	. (Ref: RD 2.1, 2.2, 2.3, 3.2)

N°	Obj	Activity Title	Activity	ISO/IEC 12207	ED109	ED-12B/ DO 178B	IEC 61508	CMMI
4	4.3.2	System Architectural Design	It should be ensured that all the system requirements are allocated among hardware, software, and manual-operations.	. (Ref: 5.3.3)	P (Ref: 2.1)	P (Ref: 2.3)	P (Ref: Part II-7.4)	. (Ref: RD 2.2, ReqM 1.4)
5	4.3.4	SW Requirements Analysis	The developer should establish and document software requirements, including the quality characteristics specifications.	. (Ref: 5.3.4)	. (Ref: 3.2 Table A-2 line 1)	. (Ref: 5.1, 11.6, 11.9)	. (Ref: 7.2)	. (Ref: RD 2.1, 2.3)
6	4.3.5	SW Architectural Design	The developer should transform the requirements for the software item into an architecture that describes its top-level structure and identifies the software components.	. (Ref: 5.3.5)	. (Ref: 3.2 Table A-2 line 3)	. (Ref: 5.2, 11.7, 11.10)	. (Ref: 7.4.3)	. (Ref: TS 2.1, 2.2)
7	4.3.6	SW Detailed Design	The developer should develop a detailed design for each software component of the software item.	. (Ref: 5.3.6)	. (Ref: 3.2 Table A-2 lines 1, 2)	. (Ref: 5.2, 11.7, 11.10)	. (Ref: 7.4.5)	. (Ref: TS 3.1)
8	4.3.6	SW Coding	The developer should produce code requirements.	. (Ref: 5.3.7)	. (Ref: 3.5 Table A-5 lines 1, 2; 3.6 Table A-6 lines 1, 2, 3, 4)	. (Ref: 5.3, 11.8, 11.11)	. (Ref: 7.4.6, 7.4.7)	. (Ref: TS 3.1, Ver)
9	4.3.7	SW Integration	The developer should integrate the software units and software components into the software item.	. (Ref: 5.3.8)	. (Ref: 3.1) Table A-1 line 1)	. (Ref: 5.4)	. (Ref: 7.4.8)	. (Ref: PI 1.1, 1.3)
10	4.3.3	System Integration	The software configuration items should be integrated, with hardware configuration items, manual operations, and other systems as necessary, into the system. The aggregates should be tested, as they are developed, against their requirements. The integration and the test results should be documented. For each qualification requirement of the system**,** a set of tests, test cases (inputs, outputs, test criteria) and test procedures for conducting System Qualification Testing should be developed and documented.	. (Ref: 5.3.10)		P (Ref: 5.4)	. (Ref: 7.5, Part II-7.5)	. (Ref: PI 1.3, 3.2, 3.3 Ver)

N°	Obj	Activity Title	Activity	ISO/IEC 12207	ED109	ED-12B/ DO 178B	IEC 61508	CMMI
11	4.3.8	SW Installation	The developer should develop a plan to install the software product in the target environment as designated in the contract. The resources and information necessary to install the software product should be determined and be available.	. (Ref: 5.3.12)			. (Ref: Part I-7.9.1.1, I-7.9.2.1, I-7.9.2.3, I-7.13.1.1 I-7.13.2.1, I-7.13.2.2)	. (Ref: PP 2, PI 1)

3.1 PROCESS IMPLEMENTATION

N°	Obj	Activity Title	Activity	ISO/IEC 12207	ED109	ED-12B/ DO 178B	IEC 61508	CMMI
1	4.3.3, 4.3.16	Lifecycle Definition	If not stipulated in the contract, the developer should define or select a software lifecycle model appropriate to the scope, magnitude, and complexity of the project. The activities and tasks of the Development Process should be selected and mapped onto the lifecycle model.	. (Ref: 5.3.1.1)	. (Ref 3.1 Table A-1 line 3)	. (Ref: 3)	. (Ref: 7.1.2.1, 7.1.2.3 =>7.1.2.5)	. (Ref: PP 1.3, 2.1)
2	4.3.3, 4.3.15	Outputs Documentation	The developer should document the outputs in accordance with the Documentation Process.	. (Ref: 5.3.1.2.a)	. (Ref 3.1 Table A-1 line 1)	. (Ref: 4.1, 4.3)	. (Ref: 7.1.2.7)	. (Ref: GP 2.2, 3.1)
3	4.3.3, 5.2	Outputs Configuration Management	The developer should place the outputs under the Configuration Management Process and perform change control in accordance with it.	. (Ref: 5.3.1.2.b)	. (Ref 3.8 Table A-8 line 1 to 6; For COTS: 4.1.7 Table 4-3 lines 1 to 4)	. (Ref: 4.3)	. (Ref: 7.1.2.8)	. (Ref: CM, GP 2.6)
4	4.3.3, 5.8, 5.2.2	Software Products Problems	The developer should document and resolve problems and non-conformances found in the software products and tasks in accordance with the Problem Resolution Process.	. (Ref: 5.3.1.2.c)	. (Ref 3.8 Table A-8 line 3)			. (Ref: PMC 2.1, 2.2, 2.3 CM 2.1, 2.2)
5	4.3.3, 5.X	Support Process Compliance	The developer should perform the Supporting processes as specified in the contract.	. (Ref: 5.3.1.2.d)	. (Ref 3.9 Table A-9 line 1)		. (Ref: 7.1.1)	. (Ref: ReqM1.1 PP2.3, PMC1.4, PPQA, PMC 1.6, 1.7, 2 Ver, Val GP2.9)
6	4.3.9, 4.3.10, 4.3.11, 4.3.12, 4.3.14	Environment Definition	The developer should select, tailor, and use those standards, methods, tools, and computer programming languages (if not stipulated in the contract) that are documented, appropriate, and established by the organisation for performing the activities of the Development Process and supporting processes.	. (Ref: 5.3.1.3)	. (Ref 3.1. Table A-1 line 3)	. (Ref: 4.4, 4.5)	. (Ref: 7.1.2.6, Annex A&B, 7.4.4.2)	. (Ref: PP 2.4 IPM 1.1 GP 2.2, 2.3, 3.1)

N°	Obj	Activity Title	Activity	ISO/IEC 12207	ED109	ED-12B/ DO 178B	IEC 61508	CMMI
7	4.3.3, 4.3.16	Development Plan	The developer should develop plans for conducting the activities of the development process. The plans should include specific standards, methods, tools, actions, and responsibility associated with the development and qualification of all requirements including safety. If necessary, separate plans may be developed. These plans should be documented and executed.	. (Ref: 5.3.1.4)	. (Ref 3.1 Table A-1 line 1)	. (Ref: 2.2, 4.1, 4.2 ,11.2)	P (Ref: 7.1.2.2)	. (Ref: PP 2, PMC 1.1 IPM 1.1, 1.3, 1.4 GP 2.2, 2.3, 3.1)
8	4.3.9, 4.3.10, 4.3.11	Development Standards	Software development standards (for each phase) consistent with the system safety objectives are defined, under change control and reviewed.		. (Ref 3.1 Table A-1 line 5)	. (Ref 4.1, 4.2)	P (Ref: 7.4.4)	P (Ref: GP2.2)
9	4.3.3	Non-Deliverable Items	Non-deliverable items may be employed in the development or maintenance of the software product. However, it should be ensured that the operation and maintenance of the deliverable software product after its delivery to the acquirer are independent of such items, otherwise those items should be considered as deliverable.	. (Ref: 5.3.1.5)	. (Ref 3.1 Table A-1 line 4)			

3.1.1 SOFTWARE DEVELOPMENT PLAN

N°	Obj	Activity Title	Activity	ISO/IEC 12207	ED109	ED-12B/ DO 178B	IEC 61508	CMMI
1	4.3.3	System Overview	The developer should provide an overview of the system (functions and their allocation to the hardware and software, the architecture, processor(s) used, hardware/software interfaces, and safety features)		P (Ref: 2 ; 4.1.3; 5.1)	. (Ref: 11.1)	. (Ref: I-7.2.1)	P (PP 2.7)
2	4.3.3	Software Overview	The developer should describe the software functions with emphasis on the proposed safety and partitioning concepts.		. (Ref 5.1)	. (Ref: 11.1)		
3	4.3.3, 4.3.16	Software Lifecycle	The developer should describe the software lifecycle processes to be used to form the specific software lifecycle(s) to be used on the project, including the transition criteria for the software development processes.		. (Ref: 3.1 Table A-1 lines 2, 3)	. (Ref: 11.1, 11.2)	. (Ref: 7.1.2.1)	. (Ref: PP 1.3, IPM 1.3)
4	4.3.3, 5.X	Software Lifecycle Data	The developer should specify the software lifecycle data that will be produced and controlled by the software lifecycle processes.		. (Ref: 5)	. (Ref: 11.1)	. (Ref: 7.1.2.7, Table 1)	. (Ref: GP 2.2, 3.1)
5	4.3.3, 4.3.16	Schedule	The developer should describe the means to provide the relevant visibility of the activities of the software lifecycle processes so reviews can be planned.		. (Ref: 3.1.2 ; 4.1.4.2 ; 5.1)	. (Ref: 11.1)		. (Ref: PP 2.1 PMC GP 2.2, 3.1)
6	4.3.9, 4.3.10	Standards	The developer should identify the SW Requirements Standards, SW Design Standards, SW Code Standards, SW testing standards, SW integration standards and System integration standards for the project. Also, references to the standards for previously developed software, including COTS software, if those standards are different.		. (Ref: 3.1 Table A-1 line 5; For COTS: 4.1.4.2)	. (Ref: 11.2)	P (Ref: 7.4.4)	. (Ref: GP 2.2, 3.1 PP 2.4)
7	4.3.3, 4.3.12, 4.3.17, 4.3.18,	Software Development Environment	The developer should state the chosen software development environment in terms of hardware and software, including:		. (Ref: 3.1. Table A-1 line 3)	. (Ref: 11.2)	P (Ref: 7.4.4)	. (Ref: PP 2.4 GP 2.2, 2.3, 3.1)

N°	Obj	Activity Title	Activity	ISO/IEC 12207	ED109	ED-12B/ DO 178B	IEC 61508	CMMI
	7.1.X		(1) The chosen requirements development method(s) and tools to be used. (2) The chosen design method(s) and tools to be used. (3) The programming language(s), coding tools, compilers, linkage editors and loaders to be used. (4) The hardware platforms for the tools to be used					
8	4.3.3, 4.3.12, 4.3.17, 4.3.18, 4.3.20, 7.2.X	Additional considerations	The developer should describe specific features, for example, complexity level, alternative methods of compliance, tool qualification, previously developed software, COTS software, HMI, deactivated code and product service history.		. (Ref: 4)	. (Ref: 11.1)		
9	4.3.3, 4.3.7	Software Integration Plan	The developer should develop an integration plan to integrate the software units and software components into the software item. The plan should include test requirements, procedures, data responsibilities, and schedule. The plan should be documented.	. (Ref: 5.3.8.1)	. (Ref: §3.1 Table A-1 lines 1, 2, 3, 4)	. (Ref: 5.4.2)	. (Ref: 7.4.7.1)	. (Ref: PI 1.1, 1.3 Ver 1.3)

3.2 SYSTEM REQUIREMENTS ANALYSIS

ED12B/DO178B does not address system-related issues (supposed to be covered by ARP 4754). ED109/DO278 neither.

N°	Obj	Activity Title	Activity	ISO/IEC 12207	ED109	ED-12B/ DO 178B	IEC 61508	CMMI
1	4.3.1	System Requirements Analysis	The specific intended use of the system to be developed should be analysed to specify system requirements. The system requirements specification should describe: functions and capabilities of the system; business, organisational and user requirements; safety, security, human-factors engineering (ergonomics), interface, operations, and maintenance requirements; design constraints and qualification requirements. The system requirements specification should be documented.	. (Ref: 5.3.2.1)	P (Ref: 2)	P (Ref: 2.1.1, 2.2)	. (Ref: Part I-7.6, Part II-7.2.3)	. (Ref: RD 1.1, 2, 3.1, 3.2)
2	4.3.1, 4.3.15	System Requirements Definition Criteria	The system requirements should be specified & documented considering the criteria listed below: a) Traceability to acquisition needs; b) Consistency with acquisition needs; c) Testability; d) Feasibility of system architectural design; e) Feasibility of operation and maintenance.	. (Ref: 5.3.2.2)	P (Ref: 2)	P (Ref: 2.1.1)	P (Ref: Part II-7.2.2)	. [Ref: a) ReqM 1.4, b) ReqM1.5 c), d), e) RD 3.1, 3.2, 3.3, 3.4, 3.5]

3.3 SYSTEM ARCHITECTURAL DESIGN

ED12B/DO178B does not address system-related issues (supposed to be covered by ARP 4754).

N°	Obj	Activity Title	Activity	ISO/IEC 12207	ED109	ED-12B/ DO 178B	IEC 61508	CMMI
1	4.3.2	System Architecture Definition	A top-level architecture of the system should be established. The architecture should identify items of hardware, software, and manual-operations. It should be ensured that all the system requirements are allocated among the items. Hardware configuration items, software configuration items, and manual operations should be subsequently identified from these items. The system architecture and the system requirements allocated to the items should be documented.	. (Ref: 5.3.3.1)	P (Ref: 2.2)	P (Ref: 2.3)	P (Ref: Part II-7.4.2)	. (Ref: TS 2.1, 2.2 RD 2.2)
2	4.3.2, 4.3.15	System Architecture Definition Criteria	The system architecture and the requirements for the items should be defined & documented considering the criteria listed below. a) Traceability to the system requirements; b) Consistency with the system requirements; c) Appropriateness of design standards and methods used; d) Feasibility of the software items fulfilling their allocated requirements; e) Feasibility of operation and maintenance.	. (Ref: 5.3.3.2)			P (Ref: Part II-7.4)	. [Ref: ReqM 1.4, TS 2.1, 2.2]

3.4 SOFTWARE REQUIREMENTS ANALYSIS

N°	Obj	Activity Title	Activity	ISO/IEC 12207	ED109	ED-12B/ DO 178B	IEC 61508	CMMI
1	4.3.4	Software Requirements Definition	The developer should establish and document software requirements, including the quality characteristics specifications, described below. a) Functional and capability specifications, including performance, physical characteristics, and environmental conditions under which the software item is to perform; b) Interfaces external to the software item; c) Qualification requirements; d) Safety specifications, including those related to methods of operation and maintenance, environmental influences, and personnel injury; e) Human-factors engineering (ergonomics) specifications, including those related to manual operations, human-equipment interactions, constraints on personnel, and areas needing concentrated human attention, that are sensitive to human errors and training; f) Data definition and database requirements; g) Installation and acceptance requirements of the delivered software product at the operation and maintenance site(s); h) User documentation; i) User operation and execution requirements; j) User maintenance requirements.	. (Ref: 5.3.4.1)	. (Ref: 3.2 Tables A2.1, A2.2)	. (Ref: 5.1, 11.9)	. (Ref: 7.2.2.3, 7.2.2.4, 7.2.2.7=> 7.2.2.11)	. (Ref: RD 2.1, 2.3)
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)

N°	Obj	Activity Title	Activity	ISO/IEC 12207	ED109	ED-12B/ DO 178B	IEC 61508	CMMI
3	4.3.4, 4.3.15, 4.3.20	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)
5	4.3.7	Software Integration Definition update	The developer should update the Software integration definition (including the plan & procedures) in accordance with the outcome of this phase	. (Ref: 5.3.5.5)			. (Ref: 7.4.3.2.f)	. (Ref: PI 1, PI GP 2.2, 2.3, 3.1)

3.5 SOFTWARE ARCHITECTURAL DESIGN

N°	Obj	Activity Title	Activity	ISO/IEC 12207	ED109	ED-12B/ DO 178B	IEC 61508	CMMI
1	4.3.5, 4.3.13	Top-Level Software Architecture Definition	The developer should transform the requirements for the software item into an architecture that describes its top-level structure and identifies the software components. It should be ensured that all the requirements for the software item are allocated to its software components and further refined to facilitate detailed design. The architecture of the software item should be documented.	. (Ref: 5.3.5.1)	. (Ref: 3.2. Table A-2 line 3)	. (Ref: 5.2.2, 11.10)	. (Ref: 7.4.1.1, 7.4.1.2, 7.4.3.1, 7.4.3.3)	. (Ref: TS 2.1, 2.2 RD 2.2
2	4.3.5	Interfaces Design	The developer should develop and document a top-level design for the interfaces external to the software item and between the software components of the software item.	. (Ref: 5.3.5.2)	. (Ref: 3.2 Table A-2 line3)	. (Ref: 11.10)	P (Ref: 7.4.2.2.b)	. (Ref: TS 2.3)
3	4.3.5, 4.3.13	Assurance Level Related Design	The design should be commensurate with the Assurance Level.		. (Ref: 3.2 Table A-2.line 3)	. (Ref: 11.10)	. (Ref: 7.4.2)	. (Ref: TS 1.1, 1.3, 2.1, 2.2)
4	4.3.9, 4.3.10, 4.3.17, 4.3.18	Software Architectural Design Standards	Definition of the methods, rules and tools to be used to develop software architectural design.		. (Ref: 3.2 Table A-2 line 3)	. (Ref: 11.7)	. (Ref: 7.4.3.2)	. (Ref: TS GP 2.2, 2.3, 3.1)
5	4.3.5	Database Top-Level design	The developer should develop and document a top-level design for the database.	. (Ref: 5.3.5.3)				. (Ref: TS 2.1, 2.2)
6	4.3.7	Software Integration Definition update	The developer should update the Software integration definition (including the plan & procedures) in accordance with the outcome of this phase	. (Ref: 5.3.5.5)			. (Ref: 7.4.3.2.f)	. (Ref: PI 1, PI GP 2.2, 2.3, 3.1)
7	4.3.5, 4.3.13, 4.3.14, 4.3.15, 4.3.20	Software Architecture Definition Criteria	The developer should design & document the architecture of the software item and the interface and database designs considering the criteria listed below. a) Traceability to the requirements of	. (Ref: 5.3.5.6)	. (Ref: 3.2 Table A-2 line 3)	. (Ref: 5.2, 5.5, 11.7)	P (Ref: 7.4.2.2 =>7.4.2.11, 7.4.3.2)	. (Ref: TS 2.1, 2.2 ReqM 1.4 PI 2.1)

N°	Obj	Activity Title	Activity	ISO/IEC 12207	ED109	ED-12B/ DO 178B	IEC 61508	CMMI
			the software item; b) External consistency with the requirements of the software item; c) Internal consistency between the software components; d) Appropriateness of design methods and standards used; e) Feasibility of detailed design; f) Feasibility of operation and maintenance.					

3.6 SOFTWARE DETAILED DESIGN

N°	Obj	Activity Title	Activity	ISO/IEC 12207	ED109	ED-12B/ DO 178B	IEC 61508	CMMI
1	4.3.6	Software Detailed Design Definition	The developer should develop a detailed design for each software component of the software item. The software components should be refined into lower levels containing software units that can be coded, compiled, and tested. It should be ensured that all the software requirements are allocated from the software components to software units. The detailed design should be documented.	. (Ref: 5.3.6.1)	. (Ref: 3.2 Table A-2 lines 4, 5)	. (Ref: 11.10)	. (Ref: 7.4.1.4, 7.4.5.1, 7.4.5.4)	. (Ref: TS 2.1, 2.2, 3.1)
2	4.3.6	Interfaces Design	The developer should develop and document a detailed design for the interfaces external to the software item, between the software components, and between the software units. The detailed design of the interfaces should permit coding without the need for further information. The detailed design of the interfaces should permit coding without the need for further information.	. (Ref: 5.3.6.2)	P (Ref: 3.2 Table A-2 lines 4, .5)	. (Ref: 5.2.2, 11.10)		. (Ref: TS 2.3)
3	4.3.9, 4.3.10	Software Detailed Design Standards	Definition of the methods, rules and tools to be used to develop software detailed design.		. (Ref: 3.1 Tables A2.4 A2.5)	. (Ref: 11.7, 11.10)		. (Ref: TS GP 2.2, 2.3, 3.1)
4	4.3.5, 4.3.13, 4.3.14, 4.3.15, 4.3.20	Software Detailed Design Definition Criteria	The developer should design & document the software detailed design considering the criteria listed below. a) Traceability to the requirements of the software item; b) External consistency with architectural design; c) Internal consistency between software components and software units; d) Appropriateness of design methods and standards used; e) Feasibility of testing; f) Feasibility of operation and maintenance.	. (Ref: 5.3.6.7)	. (Ref:3.2 Table A-2 lines 4, 5)	. (Ref: 5.2.2, 5.5, 11.7)	P (Ref: 7.4.5.2, 7.4.5.3)	. (Ref: TS 2.1, 2.2 ReqM 1.4 PI 2.1)

N°	Obj	Activity Title	Activity	ISO/IEC 12207	ED109	ED-12B/ DO 178B	IEC 61508	CMMI
5	4.3.5	Database Detailed Design	The developer should develop and document a detailed design for the database.	. (Ref: 5.3.6.3)				. (Ref: TS 2.1, 2.2, 3.1)
6	4.3.7	Software Integration Definition Update	The developer should update the Software integration definition (including the plan & procedures) in accordance with the outcome of this phase	. (Ref: 5.3.6.6)			. (Ref:7.4.5.5)	. (Ref: PI 1, PI GP 2.2, 2.3, 3.1)

3.7 SOFTWARE CODING

N°	Obj	Activity Title	Activity	ISO/IEC 12207	ED109	ED-12B/ DO 178B	IEC 61508	CMMI
1	4.3.9, 4.3.10	Coding Standards	Definition of programming languages, methods, rules and tools to be used to code software.		. (Ref: 3.2 Table A-2 line 6	. (Ref: 11.8, 11.11)	. (Ref: 7.4.4.6)	. (Ref: TS GP 2.2, 2.3, 3.1)
2	4.3.6, 4.3.15, 4.3.9, 4.3.10, 4.3.19, 4.3.20	Software Units Code definition Criteria	The developer should develop software code considering the criteria listed below. a) Traceability to the requirements and design of the software item; b) External consistency with the requirements and design of the software item; c) Internal consistency between unit requirements; d) Appropriateness of coding methods and standards used; e) Feasibility of software integration and testing; f) Feasibility of operation and maintenance.	. (Ref: 5.3.7.5)	P (Ref: 3.7 Table A-7)	. (Ref: 5.3, 5.5, 11.8, 11.11)	P (Ref: 7.4.6.1, 7.4.7.1, 7.4.7.2)	. (Ref: TS 3.1 ReqM 1.4)
3	4.3.6	Development & Documentation	The developer should develop and document each software unit and database	. (Ref: 5.3.7.1)	P (Ref: 3.5 Table A-5 - 3.6 Table A-6)	. (Ref: 5.3)	. (Ref: 7.4.6, 7.4.7)	. (Ref: TS 3.1)

3.8 SOFTWARE INTEGRATION

N°	Obj	Activity Title	Activity	ISO/IEC 12207	ED109	ED-12B/ DO 178B	IEC 61508	CMMI
1	4.3.7, 4.3.20	Software Integration Definition Criteria	The developer should update the integration plan considering the criteria listed below: a) Traceability of components to the system requirements).; b) External consistency of components with the system requirements; c) Internal consistency; d) Appropriateness of methods used; e) Feasibility of operation and maintenance;	P (Ref: 5.3.8.5)			P (Ref: 7.4.8.2, 7.4.8.5)	. (Ref: TS 2.1, 3.1, 3.2 PI 1, PI GP 2.2, 3.1 PP3.1 ReqM 1.4)
2	4.3.9, 4.3.10	Software Integration Standards	Definition of the methods, rules and tools to be used to integrate software components. Definition of methods to handle patch and deactivated code.		. (Ref: 3.6 Table A-6 line 2)	P (Ref: 5.4.3, 6.4.3.b)		. (Ref: PI 1.2 PI GP 2.2, 2.3, 3.1)
3	4.3.7	Software Integration Definition Update	The developer should update the schedule for Software Integration in accordance with the results of former verification	. (Ref: 5.3.7.4)				. (Ref: PI 1.1)
4	4.3.7	Software Integration	The developer should integrate the software units and software components as the aggregates are developed in accordance with the integration plan. It should be ensured that each aggregate interfaces other software items and that the software item is integrated at the conclusion of the integration activity. The integration should be documented.	. (Ref: 5.3.8.2)			. (Ref: 7.4.8.3, 7.4.8.4)	. (Ref: PI 3.2, 3.3)

3.9 SYSTEM INTEGRATION

N°	Obj	Activity Title	Activity	ISO/IEC 12207	ED109	ED-12B/ DO 178B	IEC 61508	CMMI
1	4.3.3	System Integration Definition	The software configuration items should be integrated, with hardware configuration items, manual operations, and other systems as necessary, into the system. The system integration should be defined & documented considering the criteria listed below: a) Appropriateness of methods and standards used; b) Conformance to expected results; c) Feasibility of system integration; d) Feasibility of operation and maintenance; e) criteria on which system integration completion will be judged.	. (Ref: 5.3.10.1, 5.3.10.3)	P (Ref: 3.2 Table A-2 line 7)	. (Ref: 5.4.1, 5.4.2)	P (Ref: 7.5.2.1, 7.5.2.2, 7.5.2.3, 7.5.2.4, 7.5.2.5, 7.5.2.7, 7.5.2.8)	. (Ref: PI 1.2, 1.3, 3.2, 3.3)
2	4.3.13, 4.3.14	Software Compatibility with target Hardware	Integration procedures should describe how to merge SW with HW, how to ensure SW compatibility with HW, integration environment.		P (Ref: 3.2 Table A-2 line 7)	. (Ref: 5.4)	. (Ref: 7.5.2)	. (Ref: PI 1.3)
3	4.3.9, 4.3.10	System Integration Standard	Definition of the methods, rules and tools to be used to integrate a system: HW/SW & system components			P (Ref: 6.4.3.a)		. (Ref: PI 1.2 PI GP 2.2, 2.3, 3.1)

3.11 SOFTWARE INSTALLATION

N°	Obj	Activity Title	Activity	ISO/IEC 12207	ED109	ED-12B/ DO 178B	IEC 61508	CMMI
1	4.3.8	Software Installation Plan	The developer should develop a plan to install the software product in the target environment as designated in the contract. The resources and information (schedule, procedures, sequence, responsibilities) necessary to install the software product should be determined and be available. As specified in the contract, the developer should assist the acquirer with the set-up activities. Where the installed software product is replacing an existing system, the developer should support any parallel running activities that are required by contract. The installation plan should be documented.	P (Ref: 5.3.12.1)			. (Ref: Part I-7.9.1.1, I-7.9.2.1, I-7.9.2.3)	. (Ref: PI 1, PI GP 2.2,2.3, 3.1
2	4.3.8	Software Installation Performance	a) The developer should install the software product in accordance with the installation plan. b) It should be ensured that the software code and databases initialise, execute, and terminate as specified in the contract. c) The installation events and results should be documented.	. (Ref: 5.3.12.2)			. (Ref: Part I-7.13.1.1, I-7.13.2.1, I-7.13.2.2)	. (Ref: PI 3.4)

4 OPERATION PROCESS

The Operation Process contains the objectives *and tasks of the operator. The process covers the operation of the software product and operational support to users. Because operation of software product is integrated into the operation of the system, the* objectives *and tasks of this process refer to the system.*

ED12B/DO178B, ED109/DO278 and CMMI do not cover operation.

N°	Obj	Activity Title	Activity	ISO/IEC 12207	ED109	ED-12B/ DO 178B	IEC 61508	CMMI
1	4.4.1	Process Implementation	a) The operator should develop a plan and set operational standards for performing the activities and tasks of this process. b) The operator should establish procedures for providing feedback.	. (Ref: 5.4.1)			P (Ref: Part I-7.15)	
2	4.4.2	Intended Operational Environment	The system should be operated in its intended environment according to the user documentation.	. (Ref: 5.4.3)			P (Ref: Part I-7.15)	
3	4.4.3	User support	The operator should provide assistance and consultation to the users as requested.	. (Ref: 5.4.4)				
4	4.4.4	Software Operation	Procedures to operate the software should be defined, documented and executed.	. (Ref: 5.4.3)			P (Ref: Part I-7.15)	
5	4.4.5	Performance Monitoring	Some means commensurate with the SWAL stringency should exist to monitor the Software performance, especially the SWAL allocated to this software, but also to provide assurance that the SWAL allocation process and criteria are correct and complete.					

5 MAINTENANCE PROCESS

The Maintenance Process contains the objectives *and tasks of the maintainer. This process is activated when the software product undergoes modifications to code and associated documentation due to a problem or the need for improvement or adaptation. The objective is to modify existing software product while preserving its integrity. This process includes the migration and decommissioning of the software product. The process ends with the decommissioning of the software product.*

ED12B/DO178B and ED109/DO278 do not cover maintenance.

N°	Obj	Activity Title	Activity	ISO/IEC 12207	ED109	ED-12B/ DO 178B	IEC 61508	CMMI
1	4.5.1	Process Implementation	The maintainer should develop, document, and execute plans and procedures for conducting the activities and tasks of the Maintenance Process.	. (Ref: 5.5.1)			P (Ref: Part I-6.2.1.j , I-7.7,I-7.15)	. (Ref: PP 2)
2		SWAL allocation confirmation	First the impact on safety of the problem or modification as provided by the "Problem Resolution Process" should be confirmed throughout the maintenance process.					
3		SWAL satisfaction	The maintainer should ensure that any maintenance activity does not impair the confidence that (new or old confirmed) SWAL is satisfied.	. (Ref: 5.5.4)			P (Ref: 7.8, Part I-7.16)	. (Ref: CM 1.3, 3.2)
4	4.5.4	Software Migration	A migration plan should be developed, documented, and executed. If a system or software product (including data) is migrated from an old to a new operational environment, it should be ensured that any software product or data produced or modified during migration are in accordance with migration requirement.	. (Ref: 5.5.5)				P (PI 3.4)
5	4.5.5	SW Decommissioning	A decommissioning plan to remove active support by the operation and maintenance organisations should be developed and documented. An impact analysis should be performed.	P (Ref: 5.5.6)			P (Ref: Part I-7.17)	

SUPPORTING LIFECYCLE PROCESSES

This clause defines the following supporting lifecycle processes:

1) Documentation process;

2) Configuration management process;

3) Quality assurance process;

4) Verification process;

5) Validation process;

6) Joint review process;

7) Audit process;

8) Problem resolution process.

The objectives and tasks in a supporting process are the responsibility of the organisation performing that process. Depending on the lifecycle phase, different organisations may be responsible for performing a process. Each organisation ensures that the process is in existence and functional.

1 DOCUMENTATION PROCESS

The Documentation Process is a process for recording information produced by a lifecycle process or activity. The process contains the set of objectives*, which plan, design, develop, produce, edit, distribute, and maintain those documents needed by all concerned such as managers, engineers, and users of the system or software product.*

ED109/DO278 and ED12B/DO178B do not prescribe or recommend delivering documents as such. Instead, some “Software Lifecycle Data” has to be produced as evidence.

N°	Obj	Activity Title	Activity	ISO/IEC 12207	ED109	ED-12B/ DO 178B	IEC 61508	CMMI
1	5.1.1	Process Implementation	A plan, identifying the documents to be produced during the lifecycle of the software product, should be developed, documented, and implemented. Document should be identified to allow searching versions (old and latest).	. (Ref: 6.1.1)	. (Ref: 3.1 Table A-1 lines 1, 2, 3, 4)	P (Ref: 4.3, 11)	. (Ref: Part I-5.1, I-5.2.7, I-5.2.9=> I-5.2.11)	. (Ref: GP 2.2, 3.1 PP2.3, 2.7 CM 1.1)
2	5.1.2	Design & Development	Each identified document should be designed in accordance with applicable documentation standards for format, content description, page numbering, figure/table placement, proprietary/security marking, packaging, and other presentation items.	. (Ref: 6.1.2)		P (Ref: 11)	. (Ref: Part I-5.2.8, I-Annex A)	. (Ref: PP 2.3 PMC 1.4)
3	5.1.3	Production	The documents should be produced and provided in accordance with the plan. Production and distribution of documents may use paper, electronic, or other media. Master materials should be stored in accordance with requirements for record retention, security, maintenance, and backup.	. (Ref: 6.1.3)		P (Ref: 4.3, 11)	P (Ref: Part I-5.2.11)	. (Ref: PMC 1.4)

N°	Obj	Activity Title	Activity	ISO/IEC 12207	ED109	ED-12B/ DO 178B	IEC 61508	CMMI
4	5.1.3, 4.3.4	Documentation (SW Requirement)	The developer should develop and update the SW Requirement.	. (Ref: 5.3.4.1)	. (Ref: 3.2 Tables A2.1, A2.2)	. (Ref: 5.1, 11.9)	. (Ref: 7.2.2.3, 7.2.2.4, 7.2.2.7=> 7.2.2.11)	. (Ref: RD 2.1, 2.3)
5	5.1.3, 4.3.5	Documentation (SW architectural design)	The developer should develop and update SW architectural design documentation.	. (Ref: 5.3.5.4)	. (Ref: 3.2, Table A.2 line 3)			. (Ref: TS 2.1, 2.2)
6	5.1.3, 4.3.6	Documentation (SW detailed design)	The developer should develop and update SW detailed design documentation as necessary.	. (Ref: 5.3.6.4)	P (Ref: For COTS 4.1.2)			. (Ref: TS 2.1, 2.2)
7	5.1.3, 4.3.6	documentation (SW coding)	The developer should develop and update the SW coding documentation as necessary.	. (Ref: 5.3.7.3)				. (Ref: TS 3.1, 2.2)
8	5.1.3, 4.3.7	Documentation (SW integration)	The developer should develop and update the SW integration documentation as necessary.	. (Ref: 5.3.8.3)				. (Ref: PI1.1, 1.3)
10	5.1.3, 4.3.3, 5.2.2	Baseline Update	Upon successful completion of the acceptance/approval/ audits, if conducted, the developer should: - Update and prepare the deliverable software product for System Integration, System Testing, Software Installation, or Software Acceptance Support as applicable. - Establish a baseline for the design and code of the software item.	. (Ref: 5.3.9.5, 5.3.11.4)				. (Ref: PI 3.4 CM 1.3)
11	5.1.4	Maintenance	The tasks, that are required to be performed when documentation is to be modified, should be performed.	. (Ref: 6.1.4)	. (Ref: 3.8 Table A-8 lines 3, 4)	P (Ref: Annex A)		. (Ref: PMC 1.4)

2 CONFIGURATION MANAGEMENT PROCESS

The Configuration Management Process is a process of applying administrative and technical procedures throughout the software lifecycle to: identify, define, and baseline software items in a system; control modifications and releases of the items; record and report the status of the items and modification requests; ensure the completeness, consistency, and correctness of the items; and control storage, handling, and delivery of the items.

Rationale: Configuration Management ensures that assurances, for the safety of the software in the system context, are at all times derived from:

- a known executable version of the software,
- a known range of configuration data, and
- a known set of software products and descriptions that have been used in the production of that version.

Note: at the equipment level, configuration management should trace software and hardware versions to ensure that compatibility is achieved.

N°	Obj	Activity Title	Activity	ISO/IEC 12207	ED109	ED-12B/ DO 178B	IEC 61508	CMMI
1	5.2.1	Process Implementation	A configuration management plan should be developed, documented & implemented The plan should describe: a)- the configuration management activities; b)- procedures and schedule for performing these activities; c)- the organisation(s) responsible for performing these activities; and their relationship with other organisations, such as software development or maintenance; d)- Software lifecycle environment control management (tools used to develop or verify SW)	P (Ref: 6.2.1)	. (Ref: 3.1 Table A-1 lines 1, 2, 3)	. (Ref: 7.1, 11.4)	P (Ref: Part I-6.2.1)	. (Ref: CM 1.2, CM GP 2.2, 2.4, 3.1)

N°	Obj	Activity Title	Activity	ISO/IEC 12207	ED109	ED-12B/ DO 178B	IEC 61508	CMMI
			e)- Definition of SW lifecycle data (all output) control management (identify for each output which kind of Configuration Management to set-up).					
2	5.2.2	Configuration Identification	A scheme should be established for identification of software items and their versions to be controlled for the project. For each software item and its versions, the following should be identified: the documentation that establishes the baseline; the version references; and other identification details. The items to be configuration-identified should be drawn with its associated configuration management level.	P (Ref: 6.2.2)	. (Ref: 3.8 Table A-8 line 1)	. (Ref: 7.2.1, 7.2.2)	P (Ref: 6.2.3.c)	. (Ref: CM 1.1, 1.3)
3	5.2.2, 5.2.3	Baseline & Configuration Item Traceability	A baseline or configuration item should be traceable to the baseline or configuration item from which it was derived.		. (Ref: 3.8 Table A-8 line 2)	. (Ref: 7.2.2.e, 7.2.2.f)		. (Ref: CM 1.3)
4	5.2.3, 5.8.4	Configuration Control	The following should be performed: identification and recording of change requests, problem reports; analysis and evaluation of the changes; approval or disapproval of the request; and implementation, verification, and release of the modified software item. An audit trail should exist, whereby each modification, the reason for the modification, and authorisation of the modification can be traced. Control and audit of all accesses to the controlled software items that handle safety or security critical functions should be performed.	. (Ref: 6.2.3)	. (Ref: 3.8 Table A-8 line 3)	. (Ref: 7.2.3=>7.2.5)	. (Ref: 6.2.3.d, 6.2.3.e)	. (Ref: CM 2, 3)
5	5.2.3	Software Lifecycle Environment Control	The objective of software lifecycle environment control is to ensure that the tools used to produce the software are identified, controlled, and retrievable.		. (Ref: 3.8 Table A-8 line 6)	. (Ref: 7.2.9)	P (Ref: 6.2.3.c)	. (Ref: CM GP 2.6)
6	5.2.4	Configuration Status Accounting	Management records and status reports that show the status and history of controlled software items including baseline should be prepared. Status	. (Ref: 6.2.4)	. (Ref: 3.8 Table A-8 line 3)	. (Ref: 7.2.6)	P (Ref: 6.2.3.e)	. (Ref: CM 3.1)

N°	Obj	Activity Title	Activity	ISO/IEC 12207	ED109	ED-12B/ DO 178B	IEC 61508	CMMI
			reports should include the number of changes for a project, latest software item versions, release identifiers, the number of releases, and comparisons of releases.					
7	5.2.5	Configuration Evaluation	The following should be determined and ensured: the functional completeness of the software items against their requirements and the physical completeness of the software items (whether their design and code reflect an up-to-date technical description).	. (Ref: 6.2.5)	. (Ref: 3.8 Table A-8 line 3)	P (Ref: 7.2.4)	. (Ref: 6.2.3.d)	. (Ref: CM 3.2)
8	5.2.6	Release Management & Delivery	The release and delivery of software products and documentation should be formally controlled. Master copies of code and documentation should be maintained for the life of the software product. The code and documentation that contain safety or security critical functions should be handled, stored, packaged, and delivered in accordance with the policies of the organisations involved. The replication process should be verified.	P (Ref: 6.2.6)	. (Ref: 3.8 Table A-8 line 4)	. (Ref: 7.2.7)	P (Ref: 6.2.3.f)	P (Ref: CM 2 CM 1.2)
9	5.2.6	Software Load Control	To ensure that the executable object code is loaded into the system with appropriate safeguards.		. (Ref: 3.8 Table A-8 line 5)	. (Ref: 7.2.8)		
10	5.2.2, 5.2.3, 5.2.6	Software Patch Management	Requirements to manage patch: use limitations & justification, configuration management, regression analysis.			. (Ref: 5.4.3)		

3 QUALITY ASSURANCE PROCESS

The Quality Assurance Process is a process for providing adequate assurance that the software products and processes in the project lifecycle conform to their specified requirements and adhere to their established plans. To be unbiased, quality assurance needs to have organisational freedom and authority from persons directly responsible for developing the software product or executing the process in the project. Quality assurance may be internal or external depending on whether evidence of product or process quality is demonstrated to the management of the supplier or the acquirer. Quality assurance may make use of the results of other supporting processes, such as Verification, Validation, Joint Reviews, Audits, and Problem Resolution.

N°	Obj	Activity Title	Activity	ISO/IEC 12207	ED109	ED-12B/ DO 178B	IEC 61508	CMMI
1	5.3.1	Process implementation	A quality assurance process tailored to the project should be established. The objectives of the quality assurance process should be to assure that the software products and the processes employed for providing those software products comply with their established requirements and adhere to their established plans. A plan for conducting the quality assurance process activities and tasks should be developed, documented, implemented, and maintained (including configuration management of evidences records) for the life of the contract.	. (Ref: 6.3.1)	. (Ref: 3.1 Table A-1 lines 1, 2, 3, 4)	. (Ref: 8.1, 8.2, 11.5)	P (Ref: 7.1.2.2, Part I-6.2.5, I-8)	. (Ref: PPQA GP 2.2, 3.1)
2	5.3.2	Product assurance	It should be assured that all the plans required by the contract are documented, comply with the contract, are mutually consistent, and are being executed as required. It should be assured that software products and related	P (Ref: 6.3.2)	. (Ref: 3.1 Table A-1 lines 6, 7; 3.9 Table A-9 line 3)	. (Ref: 8.3)		. (Ref: GP 2.9 PPQA 2)

N°	Obj	Activity Title	Activity	ISO/IEC 12207	ED109	ED-12B/ DO 178B	IEC 61508	CMMI
			documentation comply with the contract and adhere to the plans. A Software Conformity review should be performed.					
3	5.3.3	Process assurance	It should be assured that those software lifecycle processes (supply, development, operation, maintenance, and supporting processes including quality assurance) employed for the project comply with the contract and adhere to the plans. It should be assured that the internal software engineering practices, development environment, test environment, and libraries comply with the contract.	. (Ref: 6.3.3)	. (Ref: 3.9 Table A-9 line 1)	. (Ref: 8.2)		. (Ref: GP 2.9 PPQA 1)

4 VERIFICATION PROCESS

The Verification Process is a process for determining whether the software products of an activity fulfil the requirements or conditions imposed on them in the previous activities. For cost and performance effectiveness, verification should be integrated, as early as possible, with the process (such as supply, development, operation, or maintenance) that employs it. This process may include analysis, review and test.

This process may be executed with varying degrees of independence. The degree of independence may range from the same person or different person in the same organisation to a person in a different organisation with varying degrees of separation.

In the case where the process is executed by an organisation independent of the supplier, developer, operator, or maintainer, it is called Independent Verification Process (Confirmation by examination of evidence that a product, process or service, fulfils specified requirements).

N°	Obj	Activity Title	Activity	ISO/IEC 12207	ED109	ED-12B/ DO 178B	IEC 61508	CMMI
1	5.4.X	Process implementation	a) Based upon the scope, magnitude complexity, criticality analysis, target lifecycle activities and software products requiring verification should be determined. b) Verification activities and tasks, including associated methods, techniques, and tools for performing the tasks, should be selected for the target lifecycle activities and software products. c) Based upon the verification tasks as determined, a verification plan should be developed, documented and implemented.	. (Ref: 6.4.1)	. (Ref: 3.1 Table A-1 lines 1, 2, 3, 4)	. (Ref: 6.1, 11.3)	. (Ref: 7.4.1.5, 7.9.2, Part I-7.4, I-7.6,I-7.18)	. (Ref: Ver 1 Ver GP 2.2, 3.1)
2	5.4.X	Verification	According to some criteria to be defined in the verification plan, the following activities are subject to verification: Contract, Process, Requirements, Design, Source Code, Executable Code, Data, Verification Process Outputs, Integration, Documentation.	P (Ref: 6.4.2)	P (Ref: 3.1 Table A-1 lines 1, 2, 3, 4)	P (Ref: AnnexA-3 =>Annex A-7, 6.2, 6.3, 6.4)	P (Ref: 7.9.2)	. (Ref: Ver)

4.1 PROCESS IMPLEMENTATION

N°	Obj	Activity Title	Activity	ISO/IEC 12207	ED109	ED-12B/ DO 178B	IEC 61508	CMMI
1	5.4.2, 5.4.7	Criticality Evaluation criteria	A determination should be made if the project warrants a verification effort and the degree of organisational independence of that effort needed. The project requirements should be analysed for criticality. Criticality may be gauged in terms of: a) The potential of an undetected error in a system or software requirement for causing death or personal injury, mission failure, or financial or catastrophic equipment loss or damage; b) The maturity of and risks associated with the software technology to be used; c) Availability of funds and resources.	. (Ref: 6.4.1.1)	. (Ref: 2.1)	. (Ref: 2.2, 2.3.3, Annex A)	. (Ref: Part I-7.4, I-7.6)	. (Ref: Ver 1.3)
2	5.4.X	Verification Process Implementation	If the project warrants a verification effort, a verification process should be established to verify the software product.	. (Ref: 6.4.1.2)	. (Ref: 3.4, 3.5, 3.7)	. (Ref: 6)	. (Ref: 7.9.2.1=>7.9.2.7,Part I-7.18.1)	. (Ref: Ver GP 2.2, 3.1))
3	5.4.7	Verification Organisation Independence	If the project warrants an independent verification effort, a qualified organisation responsible for conducting the verification should be selected. This organisation should be assured of the independence and authority to perform the verification activities.	. (Ref: 6.4.1.3)	P (Ref: 3.1.1)	P (Ref: Annex A, 11.3.b)	. (Ref: 7.9, Part I-7.18.2.3)	P (Ref: Ver GP 2.3, 2.4, 2.7)
4	5.4.2	Verification Environment Definition	Based upon the scope, magnitude complexity, criticality analysis, target lifecycle activities and software products requiring verification should be	. (Ref: 6.4.1.4)	. (Ref: 3.1 Table A-1 line 4)	. (Ref: Annex A, 11.3.c/d)	. (Ref: 7.9.2.2)	. (Ref: Ver 1.2, GP 2.2, 2.3, 3.1)

N°	Obj	Activity Title	Activity	ISO/IEC 12207	ED109	ED-12B/ DO 178B	IEC 61508	CMMI
			determined. Verification activities and tasks, including associated methods, techniques, and tools for performing the tasks, should be selected for the target lifecycle activities and software products.					
5	5.4.2	Transition Criteria	All essential information from a phase of the software lifecycle needed for the correct execution of the next phase should be available and verified.		. (Ref: 3.1 Table A-1 line 2)	. (Ref: 11.3.e)	. (Ref: 7.9.2.6)	. (Ref: PP 1.3, PMC 1 IPM 1.3, 1.4)
6	5.4.2	Verification Plan	Based upon the verification tasks as determined, a verification plan should be developed and documented. The plan should address the lifecycle activities and software products subject to verification, the required verification tasks for each lifecycle activity and software product, and related resources, responsibilities, and schedule. The plan should address procedures for forwarding verification reports to the acquirer and other involved organisations.	. (Ref: 6.4.1.5)	. (Ref: 3.1 Table A-1 lines 1, 2, 3, 4)	. (Ref: 11.3)	. (Ref: 7.9.2.1, Part I-7.18.2.1)	. (Ref: Ver GP 2.2, 3.1)
7	5.4.2	Verification Results	The verification plan should be implemented. Problems and non-conformances detected by the verification effort should be entered into the Problem Resolution Process. All problems and non-conformances should be resolved. Results of the verification activities should be made available to the acquirer and other involved organisations.	. (Ref: 6.4.1.6)	. (Ref: 3.9 Table 9 Line 1)	. (Ref: 6.2.e)	. (Ref: 7.9.2)	. (Ref: Ver 2, 3, Ver GP 2.7, 2.8 CM 2.1 PMC 2)

4.2 VERIFICATION

N°	Obj	Activity Title	Activity	ISO/IEC 12207	ED109	ED-12B/ DO 178B	IEC 61508	CMMI
1	5.4.2	Contract Verification	The contract should be verified considering the criteria listed below: a) The supplier has the capability to satisfy the requirements. b) The requirements are consistent and cover user needs. c) Adequate procedures for handling changes to requirements and escalating problems are stipulated. d) Procedures and their extent for interface and co-operation among the parties are stipulated, including ownership, warranty, copyright and confidentiality. e) Acceptance criteria and procedures are stipulated in accordance with requirements.	. (Ref: 6.4.2.1)	. (Ref: 3.10 Table A-10 lines 1, 2, 3)			. (Ref: all) PPQA 1, Ver 2 a) SAM 1.2, b) ReqM 1.1, RD 3.3, 3.4, 3.5 c) ReqM 1.3 PMC GP 2.2, 2.4, 3.1, 2.7 d) IPM 2 e) Ver 1.3)
2	5.4.2	Process Verification	The process should be verified considering the criteria listed below: a) Project planning requirements are adequate and timely. b) Processes selected for the project are adequate, implemented, being executed as planned, and compliant with the contract. c) The standards, procedures, and environments for the project's processes are adequate. d) The project is staffed and personnel trained as required by the contract.	. (Ref: 6.4.2.2)	. (Ref: 3.3 Table A-3, 3.4 Table A-4, 3.7 Table A-7)			. (Ref: all) Ver GP 2.2, 3.1 a) ReqM 1.5, PP 3.1, 3.2 b) PP 3.1, IPM 1.1, 1.3, PMC1.1, Ver GP 2.8, PPQA 1, Ver GP 2.9 c) PP 3.1 d) PP 2.4, 2.5 Ver GP 2.3, 2.5)
3	5.4.1	System Requirements Verification	The requirements should be verified considering the criteria listed below: a) The system requirements are consistent, feasible, and testable. b) The system requirements have been appropriately allocated to hardware items, software items, and manual operations according to design criteria. c) The system requirements are consistent, feasible and testable.	. (Ref: 6.4.2.3)				. (Ref: all) Ver 2 a, c) RD 3.3, b) Ver 2 d) Ver 1.3, 2, 3 e) ReqM 1.5, RD 3.3 f) RD 3.3, 3.5 g) None
4	5.4.5	Architectural Design Verification	The design should be verified considering the criteria listed below: a) The design is correct and consistent with	. (Ref: 6.4.2.4)	. (Ref: 3.4 Table A-4	. (Ref: Annex A-4, 6.3.3)	. (Ref: 7.9.2.9)	. (Ref: all) Ver 1.3, 2 a) TS 1.1, 2.1

N°	Obj	Activity Title	Activity	ISO/IEC 12207	ED109	ED-12B/ DO 178B	IEC 61508	CMMI
			and traceable to requirements. c) Selected design can be derived from requirements. f) Design conforms to Design standards		lines 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13 Over-compliant in line 13) (partitioning)			c) TS1.1,1.3,2.1 ,ReqM1.4 f) TS 2.1)
5	5.4.5	Detailed design Verification	The design should be verified considering the criteria listed below: b) The design implements proper sequence of events, inputs, outputs, interfaces, logic flow, allocation of timing and sizing budgets, and error definition, isolation, and recovery. d) The design implements safety, and other critical requirements correctly as shown by suitably rigorous methods. e)No conflict exist between software design and the HW/SW features of the target computer (initialisation, asynchronous operation, interruptions) f) Design conforms to Design standards	. (Ref: 6.4.2.4)	. (Ref: 3.4 Table A-4 lines 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13 Over-compliant in line 13) (partitioning)	. (Ref: Annex A-4, 6.3.3)	. (Ref: 7.9.2.9)	. [Ref: Ver 1.3, 2 TS 2.1
6	5.4.6	Source Code Verification	The source code should be verified considering the criteria listed below: a) The code is traceable to design and requirements, testable, correct, and compliant with requirements and coding standards. b) The code implements proper event sequence, consistent interfaces, correct data and control flow, completeness, appropriate allocation timing and sizing budgets, and error definition, isolation, recovery, stack usage, exception handling, interrupt conflict, …. c) Selected code can be derived from design or requirements. d) The code implements safety, security, and other critical requirements correctly as shown by suitably rigorous methods. e) Source Code conforms to Code standards f) Traceability to requirements	. (Ref: 6.4.2.5)	. (Ref: 3.5 Table A-5 lines 1, 2, 3, 4, 5, 6)	. (Ref: Annex A-5)	. (Ref: 7.9.2.12)	. (Ref: Ver 1.3, 2 TS 3.1 ReqM 1.4)
7	5.4.8	Executable Code Verification	The Executable Code should be verified (traceability to requirements).		. (Ref: 3.6 Table A-6 lines 1, 2, 3, 4, 5 Over-compliant)	. (Ref: Annex A-6)		. (Ref: TS 3.1 ReqM 1.4 Ver 3)

N°	Obj	Activity Title	Activity	ISO/IEC 12207	ED109	ED-12B/ DO 178B	IEC 61508	CMMI
8	5.4.6	Software Units Test Definition	The developer should define and document test requirements and schedule for testing software units. The test requirements should include stressing the software unit at the limits of its requirements.	. (Ref: 5.3.6.5)	. (Ref: 3.6.3)		. (Ref: 7.4.5.4)	. (Ref: TS 3.1, Ver 1.3)
9	5.4.5, 5.4.6	Module Testing Standards	Definition of methods, rules and tools to be used to test software modules (unit testing).		P (Ref: 3.6 Table A-6 lines 3, 4)	P (Ref: 6.4.3.c)		. (Ref: TS GP 2.2, 2.3, 3.1, Ver 1.2)
10	5.4.12	Development & Documentation	The developer should develop and document tests procedures and data for testing each software unit and database	. (Ref: 5.3.7.1)	P (Ref: 3.5 Table A-5; 3.6 Table A-6)	. (Ref: 5.3)	. (Ref: 7.4.6, 7.4.7)	. (Ref: TS 3.1, Ver 1.3)
11	5.4.5, 5.4.6	Software Units Tests definition Criteria	The developer should develop software test considering the criteria listed below. a) Traceability to the requirements and design of the software item; b) External consistency with the requirements and design of the software item; c) Internal consistency between unit requirements; d) Test coverage of units;	. (Ref: 5.3.7.5)	P (Ref: 3.7 Table A-7)	. (Ref: 5.3, 5.5, 11.8, 11.11)	P (Ref: 7.4.6.1, 7.4.7.1, 7.4.7.2)	. (Ref: TS 3.1 ReqM 1.4 Ver 1.3, 2)
12	5.4.5, 5.4.6	Software Units Testing	The developer should test each software unit and database ensuring that it satisfies its requirements. The test results should be documented.	. (Ref: 5.3.7.2)	. (Ref: 3.6 Table A-6 lines 3, 4, 5)		. (Ref: 7.4.7.1, 7.4.7.3)	. (Ref: TS 3.1, Ver 3)
13	5.4.4	Integration Verification	The integration should be verified considering the criteria listed below: a) The software components and units of each software item have been completely and correctly integrated into the software item. b) The hardware items, software items, and manual operations of the system have been completely and correctly integrated into the system. c) The integration tasks have been performed in accordance with an integration plan. d) Linking and loading data and memory map	. (Ref: 6.4.2.6)	P (Ref: 3.5 Table A-5 line 7)	. (Ref: 6.3.5)	. (Ref: 7.9.2.10, 7.9.2.11)	P (Ref: all) Ver 2, 3 a,b) PI 3.1 c) PI GP 2.9 d, e,f) Ver 1.3 g) PI 3.2)

N°	Obj	Activity Title	Activity	ISO/IEC 12207	ED109	ED-12B/ DO 178B	IEC 61508	CMMI
			e) Incorrect HW addresses f) Memory overlaps g) Missing SW components.					
14	5.4.3	Software Requirement	The developer should conduct Software requirement verification. It should be ensured that the implementation of each software requirement is tested for compliance. The verification results should be documented. The requirements should be verified considering the criteria listed below: a) The software requirements related to safety, security, and criticality are correct as shown by suitably rigorous methods. b)No conflict exist between software requirements and the HW/SW features of the target computer (system response time, Input/output HW) c) Requirements conform to requirements standards d)Algorithms are accurate and correct.	. (Ref: 6.4.2.3, 5.3.9)	. (Ref: 3.3 Table A-3 lines 1, 2, 3, 4, 5, 6, 7)	P (Ref: Annex A-3, 6.1.a, 6.3.1, 6.3.2) (Because only Software requirements not System requirements)	. (Ref: 7.9.2.8)	. (Ref: Ver 3)
15	5.4.12	Software Verification Evaluation	The developer should evaluate the design, code, tests, test results, and user documentation considering the criteria listed below. The results of the evaluations should be documented. a) Test coverage of the requirements of the software item; b) Conformance to expected results; c) Feasibility of system integration and testing, if conducted d) Feasibility of operation and maintenance.	. (Ref: 5.3.9.3)			. (Ref: 7.7.2.4, 7.7.2.6)	. (Ref: TS 2.1, 3.1 Ver 1.3)
17	5.4.1	System Verification Evaluation Criteria	The system verification should be defined & documented considering the criteria listed below. a) Test coverage of system requirements; b) Conformance to expected results; c) Feasibility of operation and maintenance.	. (Ref: 5.3.11.2)	P (Ref: 3.7 Table A-7 lines 2, 3)	P (Ref: 2.7)	. (Ref: Part I-7.8, I-7.14, Part II-7.7.2.3, II-7.7.2.5=> II-7.7.2.7)	. (Ref: Ver 1.3, Val 1.3 Ver 3 Val 2)
18	5.4.12	System Verification Evaluation	It should be ensured that the implementation of each system requirement is tested for compliance and that the system is ready for delivery. The qualification testing results should be documented.	. (Ref: 5.3.11)			. (Ref: Part I-7.8, I-7.14, Part II-7.7)	. (Ref: Ver 3, Val 2 ReqM 1.4 Ver GP 2.9)

N°	Obj	Activity Title	Activity	ISO/IEC 12207	ED109	ED-12B/ DO 178B	IEC 61508	CMMI
19	5.4.3	Operational Testing	For each release of the software product, the operator should perform operational testing, and, on satisfying the specified criteria, release the software product for operational use.	. (Ref: 5.4.2)			P (Ref: Part I-7.15)	. (Ref: Val 2 PI 3.4)
20	5.4.3, 5.4.9	Adaptation data verification	Adaptation data should be verified [3]		. (Ref: 3.2 Table A-2 line 8)			. (Ref Ver, Val)
21	5.4.9	Data Verification	Data structures, application data modifiable parameters, plant interfaces and all communications interfaces should be verified.				(Ref: 7.9.2.13)	(Ref: Ver 1.3, 2 TS 3.1 PI 2.2)
22	5.4.12	Documentation Verification	The documentation should be verified considering the criteria listed below: a) The documentation is adequate, complete, and consistent. b) Documentation preparation is timely. c) Configuration management of documents follows specified procedures.	. (Ref: 6.4.2.7)			P (Ref: Part I-5.2)	. (Ref: a) Ver 2 b) PI GP 2.8, 2.9 c) CM 3, GP2.6)
23	5.4.12	Verification Process Outputs Verification	Test cases, test procedures and test results should be verified.		. (Ref: 3.7 Table A-7 lines 1, 2, 3, 4, 5, 6, 7, 8 Over-compliant)	. (Ref: Annex A-7, 6.3.6, 6.4.4)		. (Ref: Ver 1.3, 2)
24	5.4.2	Verification of retrieval & release process	The software retrieval and release process should be verified.					

[3] see adaptation data definition of ED109

5 VALIDATION PROCESS

The Validation Process is a process for determining whether the requirements and the final, as-built system or software product fulfils its specific intended use. Validation may be conducted in earlier stages. This process may be conducted as a part of Software Acceptance Support.

This process may be executed with varying degrees of independence. The degree of independence may range from the same person or different person in the same organisation to a person in a different organisation with varying degrees of separation. In the case where the process is executed by an organisation independent of the supplier, developer, operator, or maintainer, it is called Independent Validation Process (Confirmation by examination and provision of objective evidence that the particular requirements for a specific intended use are fulfilled (usually used for internal validation of the design)).

ED109/DO278 and ED12B/DO178B do not cover this activity, because these standards are Approval/Certification oriented. Consequently, Validation objectives, as described here after, can be considered as covered by Approval/Certification.

N°	Obj	Activity Title	Activity	ISO/IEC 12207	ED109	ED-12B/ DO 178B	IEC 61508	CMMI
1	5.5.1	Process implementation	A validation process should be established to validate the system or software product. Validation tasks, including associated methods, techniques, and tools for performing the tasks**,** should be selected. A validation plan should be developed, documented and implemented.	**.** (Ref: 6.5.1)			**.** (Ref: 7.3, 7.7 Part I-7.8, I-7.14, I-8)	**.** (Ref: Val 1.2, Val GP 2.2, 3.1)
2	5.5.2, 5.5.3, 5.5.4, 5.5.5, 5.5.6, 5.5.7	Validation	Prepare selected test requirements, test cases, and test specifications for analysing test results. Ensure that these test requirements, test cases, and test specifications reflect the particular requirements for the specific intended use. Test and validate the software product as appropriate in selected areas of the target environment.	**.** (Ref: 6.5.2)			**.** (Ref: 7.3.2, 7.7.2)	**.** (Ref: a,b) Val 1.1, 1.3 c) Val 2)

6 JOINT REVIEW PROCESS

The Joint Review Process is a process for evaluating the status and products of an activity of a project as appropriate. Joint reviews are at both project management and technical levels and are held throughout the life of the contract. This process may be employed by any two parties, where one party (reviewing party) reviews another party (reviewed party).

ED109/DO278, ED 12B/DO 178B and IEC 61508 do not define a specific process for Joint Review objectives. However, reviews are part of their Verification process.

N°	Obj	Activity Title	Activity	ISO/IEC 12207	ED109	ED-12B/ DO 178B	IEC 61508	CMMI
1	5.6.1	Process implementation	Periodic reviews should be held at predetermined milestones as specified in the project plan(s). The review results should be documented and distributed.	. (Ref: 6.6.1)	P (Ref: 3.9 Table A-9 Line 3 partial)	P (Ref: 6, 8.3)	P (Ref: Part I-6.2.1.b, I-7.18.1)	. (Ref: PP 2.1, 2.6 PMC 1.6, 1.7)
2	5.6.2	Project management reviews	Project status should be evaluated relative to the applicable project plans, schedules, standards, transition criteria and guidelines.	. (Ref: 6.6.2)	P (Ref: 3.9 Table A-9 Line 1 partial)	P (Ref: 4.6, 8.2.b/c)	P (Ref: 7.3.2.4, Part I-6.2.3)	. (Ref: PMC 1)
3	5.6.3	Technical reviews	Technical reviews should be held to evaluate the software products or services under consideration.	. (Ref: 6.6.3)	P (Ref: 3.9 Table A-9 Line 3 partial)	. (Ref: 6, 8.3)	. (Ref: 7.2.2.4, 7.4.1.2, 7.4.6.2, 7.4.4.5, Part I-5.2.11)	. (Ref: PMC 1.6 , 1.7)
4	5.6.3	Software Requirements Joint Review	The developer should conduct joint review(s) in accordance with Joint Review Process. Upon successful completion of the review(s) a baseline for the requirements of the software item should be established	. (Ref: 5.3.4.3)				. (Ref: PMC 1.6, 1.7 CM 1.3)
5	5.6.3	Software Architecture Joint Review	The developer should conduct joint review(s) in accordance with Joint Review Process.	. (Ref: 5.3.5.7)				. (Ref: PMC 1.6, 1.7)

N°	Obj	Activity Title	Activity	ISO/IEC 12207	ED109	ED-12B/ DO 178B	IEC 61508	CMMI
6	5.6.3	Software Detailed Design Joint Review	The developer should conduct joint review(s) in accordance with Joint Review Process.	. (Ref: 5.3.6.8)				. (Ref: PMC 1.6, 1.7)
7	5.6.3	Code Joint Review	The developer should conduct joint review(s) in accordance with Joint Review Process.					. (Ref: PMC 1.6, 1.7)
8	5.6.3	Software Integration Joint Review	The developer should conduct joint review(s) in accordance with Joint Review Process.	. (Ref: 5.3.8.6)				. (Ref: PMC 1.6, 1.7)

7 AUDIT PROCESS

The Audit Process is a process for determining compliance with the requirements, plans, and contract as appropriate. This process may be employed by any two parties, where one party (auditing party) audits the software products or activities of another party (audited party).

ED109/DO278 and ED 12B/DO 178B do not define a specific process for Audit. However, audits are part of Software Quality Assurance process.

N°	Obj	Activity Title	Activity	ISO/IEC 12207	ED109	ED-12B/ DO 178B	IEC 61508	CMMI
1	5.7.1	Process implementation	a) Audits should be held at predetermined milestones as specified in the project plan(s). b) After completing an audit, the audit results should be documented and provided to the audited party.	. (Ref: 6.8.1)		P (Ref: 8.2.d, 11.19)	P (Ref: Part I-6.2.1.k, 7.8.2.2, Part I-7.7.2.1.c, I-7.15.2.3)	. (Ref: .GP 2.7, 2.9)
2	5.7.2, 5.7.3, 5.7.4, 5.7.5	Software Audit	Audits should be conducted to ensure that: a) As-coded software products (such as a software item) reflect the design documentation. b) The acceptance review and testing requirements prescribed by the documentation are adequate for the acceptance of the software products. c) Test data comply with the specification. d) Software products were successfully tested and meet their specifications. e) Test reports are correct and discrepancies between actual and expected results have been resolved. f) User documentation complies with standards as specified. g) Activities have been conducted according to applicable requirements, plans, and contract. h) The costs and schedules adhere to the established plans.	. (Ref: 6.8.2)		P (Ref: 8.2.d)	P (Ref: 6.2.3.e)	. (Ref: CM 3.2 a) TS 3.1, Ver 2 b) Ver 1.1, Ver 2 c) Ver 1.3, Ver 2 d) Ver 2, 3, PMC 2, CM 2.1, 3.2 e) Ver 3, PMC 2, CM 2.1, 3.2 f) TS 3.2, Ver 2 g) GP 2.9 h) PMC 1.1)

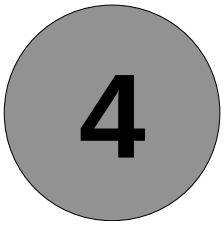
8 PROBLEM RESOLUTION PROCESS

The Problem Resolution Process is a process for analysing and resolving the problems (including non-conformances), whatever their nature or source, that are discovered during the execution of development, operation, maintenance, or other processes. The objective is to provide a timely, responsible, and documented means to ensure that all discovered problems are analysed and resolved and trends are recognised.

N°	Obj	Activity Title	Activity	ISO/IEC 12207	ED109	ED-12B/ DO 178B	IEC 61508	CMMI
1	5.8.1	Process implementation	A problem resolution process should be established for handling all problems (including non-conformances) detected in the software products and activities.	. (Ref: 6.9.1)	. (Ref: 3.8 Table A-8 line 3)	. (Ref: 7, 11.17)	. (Ref: 7.8, Part I-6.2.1, I-7.16)	. (Ref: PMC GP 2.2,3.1)
2	5.8.2	Problem resolution	When problems (including non-conformances) have been detected in a software product or an activity, a problem report should be prepared to describe each problem detected. The problem report should be used as part of a closed-loop process: from detection of the problem, through investigation, analysis and resolution of the problem and its cause, and onto trend detection across problems.	. (Ref: 6.9.2)	. (Ref: 3.8 Table A-8 line 3)	. (Ref: 7.2, 11.17)	. (Ref: 7.8.2)	. (Ref: PMC 2 CAR)
3	5.8.3	Problem & Modification Analysis	a) Problem report or modification request should be analysed for its impact on the organisation, the existing system, and the interfacing systems. b) In particular safety impact should be analysed in order to assess the consequences and severity of such a	. (Ref: 5.5.2 ; 6.9.1)			P (Ref: 7.8, Part I-7.8, I-7.15)	. (Ref: ReqM 1.3 CM 2)

N°	Obj	Activity Title	Activity	ISO/IEC 12207	ED109	ED-12B/ DO 178B	IEC 61508	CMMI
			change/modification to ensure that its anomalous behaviour does not lead to consequences, which are not compatible with the initial SW AL.					
4	5.2.3, 5.8.4	Problem Report Configuration Management	Problem report should be put under configuration management.	. (Ref: 6.2.3)	. (Ref: 3.8 Table A-8 line 3)	. (Ref: 7.2.3=>7.2.5)	. (Ref: 6.2.3.d, 6.2.3.e)	. (Ref: CM 2, 3)

These process objectives are part of Software Configuration Management for ED109/DO278 and ED12B/DO178B.



ORGANISATIONAL LIFECYCLE PROCESSES

This clause defines the following organisational lifecycle processes:

1) Management process;

2) Infrastructure process;

3) Improvement process;

4) Training process.

The objectives and tasks in an organisational process are the responsibility of the organisation using that process. Depending on the lifecycle phase, different organisations may be responsible for performing a process. Each organisation ensures that the process is in existence and functional.

1 MANAGEMENT PROCESS

The Management Process contains the generic objectives *and tasks, which may be employed by any party that has to manage its respective processes). The manager is responsible for product management, project management, and task management of the applicable process(es), such as the acquisition, supply, development, operation, maintenance, or supporting process.*

ED109/DO278, ED-12B/DO-178 B and IEC-61508 do not provide **generic** requirements for management. That is why all requirements concerning management are referenced in the related process, for example planning objectives of the supplier are referenced in Supplier Process – Planning (cf : 3.4).

N°	Obj	Activity Title	Activity	ISO/IEC 12207	ED109	ED-12B/ DO 178B	IEC 61508	CMMI
1	6.1.1	Initiation & scope definition	The management process should be initiated by establishing the requirements of the process to be undertaken. The manager should establish the feasibility of the process by checking that the resources (personnel, materials, technology, and environment) required to execute and manage the process are available, adequate, and appropriate and that the time-scales to completion are achievable.	. (Ref: 7.1.1)				. (Ref: a) GP 2.2, 3.1 PP2.4 b) PP 3.2)
2	6.1.2	Planning	The manager should prepare the plans for execution of the process. The plans associated with the execution of the process should contain descriptions of the associated activities and tasks and identification of the software products that will be provided. These plans should include, but are not limited to, the following: - Schedules for the timely completion of tasks; - Estimation of effort; - Adequate resources needed to execute the tasks; - Allocation of tasks; - Assignment of responsibilities;	. (Ref: 7.1.2)				. (Ref: all) PP1, 2 GP 2.2, 3.1 b) PP 2.1 c) PP 1.4 d) PP 2.4, 2.5 e) GP 2.3, 2.4 f) GP 2.4 g) PP 2.2 RskM 2.2 h) M&A 1 PMC GP 2.2, 3.1 Ver GP 2.2, 3.1

N°	Obj	Activity Title	Activity	ISO/IEC 12207	ED109	ED-12B/ DO 178B	IEC 61508	CMMI
			- Quantification of risks associated with the tasks or the process itself; - Quality control measures to be employed throughout the process; - Costs associated with the process execution; - Provision of environment and infrastructure.					i) PP 1.4 j) PP 2.4 GP 2.3)
3	6.1.3	Execution & control	The manager should initiate the implementation of the plan to satisfy the objectives and criteria set, exercising control over the process. The manager should monitor the execution of the process, providing both internal reporting of the process progress and external reporting to the acquirer as defined in the contract. The manager should investigate, analyse, and resolve the problems discovered during the execution of the process.	. (Ref: 7.1.3)				. (Ref: a) PMC GP 2.8 b) PMC 1 c) PMC 2)
4	6.1.4	Review & evaluation	a) The manager should ensure that the software products and plans are evaluated for satisfaction of requirements. b) The manager should assess the evaluation results of the software products, activities, and tasks completed during the execution of the process regarding the achievement of the objectives and completion of the plans.	. (Ref: 7.1.4)				. (Ref: a) ReqM 1.5 b) PMC)
5	6.1.5	Closure	When all software products, activities, and tasks are completed, the manager should determine whether the process is complete taking into account the criteria as specified in the contract or as part of organisation's procedure. The manager should check the results and records of the software products, activities, and tasks employed for completeness. These results and records should be archived in a suitable environment as specified in the contract.	. (Ref: 7.1.5)				. (Ref: a) IPM 1.3 b) PMC 1.1, 1.6, 1.7, GP 2.8 c) PMC 1.4)

2 INFRASTRUCTURE PROCESS

The Infrastructure Process is a process to establish and maintain the infrastructure needed for any other process. The infrastructure may include hardware, software, tools, techniques, standards, and facilities for development, operation, or maintenance.

N°	Obj	Activity Title	Activity	ISO/IEC 12207	ED109	ED-12B/ DO 178B	IEC 61508	CMMI
1	6.2.1	Process implementation	The infrastructure should be defined and documented to meet the requirements of the process (e.g. development or verification) employing the infrastructure, considering the applicable procedures, standards, tools, and techniques. The establishment of the infrastructure should be planned and documented.	. (Ref: 7.2.1)	P (Ref: 3.1 Table A-1 line 3 partial)	. (Ref: 4.4, 11.2)	P (Ref: 6.2.3.c, 7.4.4)	. (Ref: a) PP 2.4 GP 2.3 b) PP 2)
2	6.2.2	Establishment of the infrastructure	The configuration of the infrastructure should be planned and documented. Functionality, performance, safety, security, availability, space requirements, equipment, costs, and time constraints should be considered.	. (Ref: 7.2.2)	P (Ref: 3.8 Table A-8 line 6 partial)	. (Ref: 4.4, 7.2.9, 11.15)	P (Ref: 8.3)	. (Ref: CM 1.1, CM GP 2.2, 3.1)
3	6.2.3	Maintenance of the infrastructure	The infrastructure should be maintained, monitored, and modified as necessary to ensure that it continues to satisfy the requirements of the process (e.g. development or verification)) employing the infrastructure. As part of maintaining the infrastructure, the extent to which the infrastructure is under configuration management should be defined.	. (Ref: 7.2.3)	. (Ref: 3.8 Table A-8 line 6	. (Ref: 7.2.9)	P (Ref: 6.2.3.c)	. (Ref: PMC 1.1 CM)

3 IMPROVEMENT PROCESS

The Improvement Process is a process for establishing, assessing, measuring, controlling, and improving a software lifecycle process.

N°	Obj	Activity Title	Activity	ISO/IEC 12207	ED109	ED-12B/ DO 178B	IEC 61508	CMMI
1	6.3.1	Process implementation	The organisation should establish a suite of organisational processes for all software lifecycle processes as they apply to its business activities. The processes and their application to specific cases should be documented in organisation's publications. As appropriate, a process control mechanism should be established to develop, monitor, control, and improve the process(es).	. (Ref: 7.3.1)				. (Ref: OPD 1.3, 2.1, OPD GP 2.6 OPF)
2	6.3.2	Process assessment	A process assessment procedure should be developed, documented, and applied. Assessment records should be kept and maintained. The organisation should plan and carry out reviews of the processes at appropriate intervals to ensure their continuing suitability and effectiveness in the light of assessment results.	. (Ref: 7.3.2)				. (Ref: OPF 1.2)
3	6.3.3	Process improvement	The organisation should effect such improvements to its processes as it determines to be necessary as a result of process assessment and review. Process documentation should be updated to reflect improvement in the organisational processes.	. (Ref: 7.3.3)				. (Ref: OPF 1.3, 2.1, 2.2)

ED109/DO278, ED 12B/DO 178B and IEC 61508 do not cover this process.

4 TRAINING PROCESS

The Training Process is a process for providing and maintaining trained personnel. The acquisition, supply, development, operation, or maintenance of software products is largely dependent upon knowledgeable and skilled personnel. For example: developer personnel should have essential training in software management and software engineering. It ***is,*** *therefore, imperative that personnel training be planned and implemented early so that trained personnel are available as the software product is acquired, supplied, developed, operated, or maintained.*

N°	Obj	Activity Title	Activity	ISO/IEC 12207	ED109	ED-12B/ DO 178B	IEC 61508	CMMI
1	6.4.1	Process implementation	A review of the project requirements should be conducted to establish and make timely provision for acquiring or developing the resources and skills required by the management and technical staff. The types and levels of training and categories of personnel needing training should be determined. A training plan, addressing implementation schedules, resource requirements, and training needs, should be developed and documented.	. (Ref: 7.4.1)			. (Ref: Part I-6.2.1, I-Annex B)	. (Ref: PP 2.5 PMC 1.1 GP 2.5 OT 1.1, 1.3)
2	6.4.2	Training material development	Training manuals, including presentation materials used in providing training, should be developed.	. (Ref: 7.4.2)				. (Ref: OT 1.4)
3	6.4.3	Training plan implementation	The training plan should be implemented to provide training to personnel. Training records should be maintained.	. (Ref: 7.4.3)			. (Ref: Part I-6.2.2, I-Annex B)	. (Ref: PP 2.5 OT 2.1, 2.2)

ED109/DO278 and ED 12B/DO 178B do not cover this process.

This page is intentionally left blank

ADDITIONAL ANS SOFTWARE LIFECYCLE OBJECTIVES

These additional software lifecycle objectives are the following:

1) Software Development Environment

2) Commercial Off The Shelf (COTS) Considerations

1 SOFTWARE DEVELOPMENT ENVIRONMENT

This paragraph is some kind of complement to the Infrastructure Process (which is generic) for the Software Lifecycle environment concerning the Development Process.

N°	Obj	Activity Title	Activity	ISO/IEC 12207	ED109	ED-12B/ DO 178B	IEC 61508	CMMI
1	7.1.1	Definition	A suitable set of development tools should be selected for the required Assurance Level.			. (Ref: 4.4.1)	P (Ref: 7.4.4)	. (Ref: GP 2.3)
2	7.1.2	Programming Languages	Suitable programming languages should be selected for the required Assurance Level.				P (Ref: 7.4.4.3)	. (Ref: TS 3.1, GP 2.3)
3	7.1.3	Compiler Considerations	Compilers features (optimisations, limitations, ..) should be defined.			. (Ref: 4.4.2)		. (Ref: TS 3.1, GP 2.3)
4	7.1.1	Software Development Tool Qualification	The limitations for such a qualification should be defined.		P (Ref: 5.2)	. (Ref: 12.2, 12.2.1)	P (Ref: 7.7.2.7)	. (Ref: Ver 1.2)

2 COMMERCIAL OFF THE SHELF (COTS) CONSIDERATIONS

Due to the extent of COTS usage in ANS software, a special attention has to be paid to COTS.

This section will be reviewed once ED109 is published, as the intention is to base the ANS software lifecycle on ED109 as far as COTS is concerned. Anyhow as ED109 is not yet published when this document "ANS Software lifecycle" is released, some words have been extracted from ED109 to give guidance.

Text in "Times New Roman" font is extracted from ED109.

2.1 COTS DEFINITION

First a definition of COTS as used in this document is necessary:

COTS software encompasses a wide range of software, including purchased software, Non-Developmental Items (NDI), and software previously developed without consideration of ED-109. The term "Previously Developed Software" is also used for such software. This software may or may not have been approved through other "approval processes." **Partial data or no data may be available as evidence of objectives of ANS developmental process.** For the rest of this section, all such software is referred to as COTS for the sake of brevity. This terminology was selected because of the usual use of the term "COTS" within the "ground" ANS community.

Examples of COTS are operating systems, real-time kernels, graphical user interfaces, communication and telecommunication protocols, language run-time libraries, mathematical and low-level bit routines, and string manipulation routines. COTS software can be purchased apart from or in conjunction with COTS hardware, such as workstations, mainframes, communication and network equipment, or hardware items (e.g., memory, storage, I/O devices). There also may be some instances where the use of COTS software is impractical to avoid, e.g., library code associated with certain compilers.

COTS deliverables vary by the contract with the COTS supplier. They may extend from license rights, executable code, user documentation, and training to the full set of COTS lifecycle data, including the source code resulting from the COTS development. COTS information disclosure relates to cost, protection of intellectual properties, and legal questions (e.g., ownership of the software, patents, liability, and documentation responsibility). These aspects are beyond the scope of this guidance material, which addresses only those aspects that are specific to software assurance.

Development processes used by COTS suppliers and procurement processes applied by acquirers may not be equivalent to recommended processes, and may not be fully consistent with the guidance of this document. The use of COTS may mean that

alternate methods are used to gain assurance that the appropriate objectives are satisfied. These methods include, but are not limited to, product service experience, prior assurance, process recognition, reverse engineering, restriction of functionality, formal methods, and audits and inspections. Data may also be combined from more than one method to gain assurance data that the objectives are satisfied.

In cases where sufficient data is not available to satisfy the objectives, this section may be used as guidance with agreement from the appropriate Approval Authority.

2.2 Scope of COTS Section

This section applies only to COTS used for ANS applications and is not intended to alter or substitute any of the objectives applied to ANS software unless justified by a safety assessment process and accepted by the appropriate Approval Authority.

2.3 System Aspects Relating to COTS in ANS

COTS software may need to be integrated into high integrity ANS systems or equipment; however, the higher the risk of the ANS function, the more demanding the assurance requirements are for the system and the software. Alternate methods may be used to augment design assurance data for COTS software components at a desired assurance level. When COTS are used in an ANS system, additional considerations such as planning, acquisition, and verification should be addressed.

Risk mitigation techniques may be considered to reduce the ANS system's reliance on the COTS. The goal of these mitigation techniques is to accommodate the assigned SWAL by reducing the effect of anomalous behaviour of COTS on the ANS system function. Risk mitigation techniques may be achieved through a combination of people, procedure, equipment, or architecture. For example, architectural means may involve partitioning, redundancy, safety monitoring, COTS safe subsets by the use of encapsulation or wrappers, and data integrity checking.

2.4 COTS Planning Process

The purpose of the COTS planning process is to co-ordinate lifecycle processes specific to COTS and to define the methods and tools necessary for the incorporation of COTS in ANS systems. The verification of the COTS planning process is to assure that all issues regarding the use of COTS have been addressed. The ANS software planning process should accommodate COTS software if its use is anticipated. The COTS planning process includes planning for all aspects of COTS, including acquisition, verification, configuration management, and software quality assurance.

As part of the approval process, early submittal of the results of the COTS assessment and selection processes to the appropriate Approval Authority is recommended.

2.4.1 COTS Planning Process Objectives

The objectives of the COTS planning process are:

a. Activities for acquisition and integral processes, including additional considerations, integration, and maintenance, are defined.

b. Transition criteria for these processes and transition criteria with respect to ANS life cycle processes are defined.

c. Plans for COTS processes, including COTS transition criteria, are consistent with the ANS software plans.

2.4.2 COTS Planning Process Activities

The activities associated with the COTS planning process are:

a. COTS planning activities should evaluate the level of applicability of the COTS product to ANS requirements. The following considerations should be included in the evaluation to determine the level of effort involved in the use of COTS:

(1) Product availability.
(2) Requirements (mapping of ANS requirements to COTS capabilities; reference § 3.5 of this chapter).
(3) Availability of life cycle data.
(4) Level of integration and extent of additional efforts, such as, glue code, architecture mitigation techniques, etc. to allow incorporation of the COTS into the ANS system.
(5) Availability of applicable product service history or service experience.
(6) Supplier qualifications, such as, the use of standards, service history and length of service, technical support, etc.
(7) Configuration control, including visibility of COTS supplier's product version control.
(8) Modification considerations. Modified COTS has additional considerations of warranty, authority to modify, continued technical support, etc., unless such modifications are allowed by the COTS supplier. The modifications themselves should be considered a new development. Change impact analysis should be performed to determine the extent of the necessary re-verification.
(9) Maintenance issues (e.g., patches, retirement, obsolescence, and change impact analysis).
(10) Evidence of SQA activities.
(11) Verifiability of the COTS software (includes limitations, need for special test facilities, etc.).

(12) Level of compliance with SWAL objectives.
(13) Information on COTS in-service problems and resolution of those problems.

b. Relationships between the COTS planning process, the COTS acquisition process, and the COTS integral processes should be defined. Additionally, relationships between COTS processes and appropriate ANS life cycle processes should be defined. Every input to a process need not be complete before that process can be initiated, if the transition criteria established for the process are satisfied.

c. Reviews should be conducted to ensure:

(1) The COTS planning process and the ANS planning process are consistent.
(2) COTS transition criteria are compatible with the ANS transition criteria.
(3) Transition criteria are verified to assure that the outputs of each process are sufficient to begin the next process.

Note: COTS usage may necessitate considerations of glue code, architectural mitigation techniques, derived requirements, and COTS-specific integration. Any supplemental software due to COTS integration in ANS systems should be considered ANS developmental software for which all of the objectives in this document apply.

2.5 COTS Acquisition Process

The focus of this section is on the assurance aspects of acquiring COTS. There are additional acquisition considerations not described in this document. The COTS acquisition process is comprised of requirements definition, assessment, and selection.

a. Requirements Definition: The ANS software requirements definition process identifies software requirements that COTS may satisfy. COTS may contain more capabilities than the requirements needed by the ANS system. A definition of these capabilities may be available from the supplier or derived from the COTS user's manuals, technical materials, product data, etc. In the model depicted in Figure 3.5-1, the ANS requirements satisfied by COTS are the intersection of COTS capabilities and ANS requirements.

Due to the use of COTS, there may be derived requirements (e.g., platform dependent requirements, interrupt handling, interface handling, resource requirements, usage constraints, error handling, partitioning) that should be added to the ANS software requirements.

All ANS requirements satisfied by the COTS software and the resulting derived requirements should be provided to the safety assessment process.

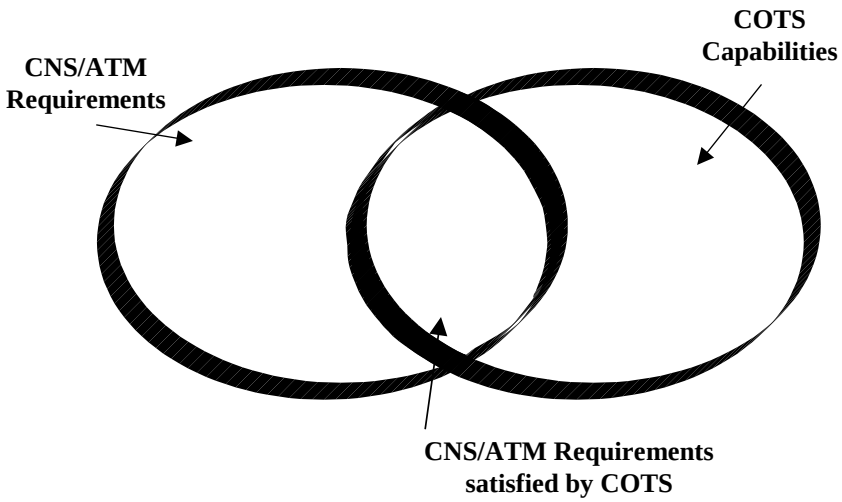


FIGURE 3.5-1 – Requirements Intersection

b. Assessment: Candidate COTS products should be assessed for their ability to implement the ANS requirements, for the effect of their respective derived requirements, and for their support of the needed assurance level. During the COTS assessment process, more than one COTS candidate product may be examined to determine the extent of intersection of COTS capabilities with the ANS requirements as depicted in Figure 3.5-1. Availability and relevance of COTS life cycle data to support the appropriate assurance level should also be assessed. Additionally, the impact of any unneeded COTS capabilities should be assessed.

d. Selection: The selection is an iterative process based on results from the assessment process and comparison of COTS suppliers (e.g., COTS supplier's experience in ANS, the ability of the COTS supplier to support COTS software version control and maintenance over the expected lifetime of the ANS system, COTS supplier's commitment to keep the ANS applicants informed of detected errors, COTS supplier's willingness to address the issue of Escrow). Analyses may be conducted to compare advantages of using COTS versus developing the software.

2.5.1 COTS Acquisition Process Objectives

The objectives of the COTS acquisition process are:

a. The degree to which of the ANS software requirements are satisfied by the COTS capabilities is determined.

b. The adequacy of life cycle data available for assurance purposes is determined.

c. The derived requirements are identified. Derived requirements consist of:

(1) Requirements imposed on the ANS system due to the usage of COTS.

(2) Requirements to prevent the unneeded capabilities of the COTS from adversely affecting the ANS system.

d. The compatibility of COTS with target hardware and other ANS software is assured.

2.5.2 COTS Acquisition Process Activities

The activities of the COTS acquisition process are:

a. The COTS capabilities should be examined, and an analysis should be conducted against the ANS requirements. The purpose of this analysis is to determine the ANS requirements satisfied by COTS and to aid in the comparison of candidate COTS products.

b. Available COTS software lifecycle data should be assessed. A gap analysis should be performed against the objectives of this document for the proposed software assurance level. This analysis aids in comparison of candidate COTS products. This analysis is used to identify the objectives that are partially or fully satisfied, and those that need to be addressed through alternate methods.

c. Analysis should be conducted to identify derived requirements. This analysis should include all COTS software capabilities, both necessary and unnecessary. Derived requirements may be classified as follows:

(1) Requirements to prevent adverse effects of any unneeded functions of any COTS software. This may result in requirements for isolation, partitioning, wrapper code, coding directives, customization, etc.

(2) Requirements that the selected COTS may impose on the ANS system including those for preventing adverse effects of needed COTS functions (e.g. input formatting, call order, initialization, data conversion, resources, range checking). This may result in requirements for interface code, coding directives, architecture considerations, resource sizing, glue-code, etc.

d. All ANS requirements satisfied by COTS software, the resulting derived requirements, and any pertinent supplier-provided data should be provided to the safety assessment process.

e. The selected COTS should be shown to be compatible with the target computer(s) and interfacing systems.

2.6 COTS Verification Process

The COTS verification process is an extension of the verification process discussed in this document (Part I – Chapter 3, §4). In particular, the COTS acquisition process frequently identifies verification objectives that cannot be satisfied using traditional means. For those verification objectives where compliance cannot be demonstrated by the available COTS data (e.g., design or requirements), additional activities, including alternate methods such as reverse engineering, may be used after acceptance by the Approval Authority.

2.6.1 COTS Verification Process Objectives

There are no additional verification objectives imposed upon the ANS system because of use of COTS.

2.6.2 COTS Verification Process Activities

Typical verification activities for COTS software achieved include:

a. Software reviews and analyses of ANS requirements satisfied by COTS,
b. Requirements-Based Testing (RBT) of ANS requirements satisfied by COTS,
c. Verification of development of any supplemental software due to COTS (e.g., glue code, partitioning, wrappers), and
d. Verification of integration of COTS into the ANS system.

2.6.3 Alternative Methods for COTS

The use of alternate methods should satisfy both of the following conditions:

a. The safety assessment process supports the justification.
b. Acceptance is granted by the appropriate Approval Authority.

Activities used for specific alternate methods or for combination of alternate methods are considered on a case-by-case basis. An example of activities associated with the usage of service experience for assurance credit is provided below in Section 3.6.4.

2.6.4 Use of Service Experience for Assurance Credit of COTS Software

Use of service experience data for assurance credit is predicated upon two factors: sufficiency and relevance. Sufficient service experience data may be available through the typical practice of running new ANS systems in parallel with operational

systems in the operational environment, long duration of simulation of new ANS systems, and multiple shadow operations executing in parallel at many locations. Relevant service experience data may be available for ANS systems from reuse of COTS software from in-service ANS Systems, or ANS system verification and pre-operational activities. For COTS software with no precedence in ANS applications, many processes may be used to collect service experience; examples include the validation process, the operator training process, the system qualification testing, the system operational evaluation, and field demonstrations.

The following applies for accumulation of service experience:

a. The use, conditions of use, and results of COTS service experience should be defined, assessed by the safety assessment process, and submitted to the appropriate Approval Authority.

b. The COTS operating environment during service experience time should be assessed to show relevance to the intended use in ANS. If the COTS operating environment of the existing and intended applications differ, additional verification should be performed to ensure that the COTS application and the ANS applications will operate as intended in the target environment. It should be assured that COTS capabilities to be used are exercised in all operational modes. Analysis should also be performed to assure that relevant permutations of input data are executed.

c. Any changes made to COTS during service experience time should be analysed. An analysis should be conducted to determine whether the changes made to COTS alter the applicability of the service experience data for the period preceding the changes.

d. All in-service problems should be evaluated for their potential adverse effect on ANS system operation. Any problem during service experience time, where COTS implication is established and whose resulting effect on ANS operations is not consistent with the safety assessment, should be recorded. Any such problem should be considered a failure. A failure invalidates the use of related service experience data for the period of service experience time preceding the correction of that problem.

e. COTS capabilities which are not necessary to meet ANS requirements should be shown to provide no adverse effect on ANS operations.

f. Service experience time should be the accumulated in-service hours. The number of copies in service should be taken into account to calculate service experience time, provided each copy and associated operating environment are shown to be relevant, and that a single copy accounts for a certain pre-negotiated percentage of the total.

Note: The text here after is added as a note in ED109/DO278, which make it informative and not normative. However, putting this text as informative was the result of a consensus with airworthiness experts. EATMP Software Task Force Members has decided to put it as normative.

Available COTS data may not be able to demonstrate satisfaction of all of the verification objectives described in this document. For example, high-level requirements testing for both robustness and normal operation may be demonstrated for COTS but the same tests for low-level requirements may not be accomplished. The use of service experience may be proposed to demonstrate satisfaction of these verification objectives for COTS. The amount of service experience to be used is selected based on engineering judgement and experience with the operation of ANS systems. The results of software reliability models cannot be used to justify service experience time. A possible approach for different assurance levels is provided below:

(1) Cannot be applied for SWAL1.
(2) A minimum of one year (8,760 hours) of service experience with no failure for SWAL2.
(3) A minimum of six months (4,380 hours) of service experience with no failure for SWAL3.
(4) SWAL4 objectives are typically satisfied without a need for alternate methods.

2.7 COTS Configuration Management Process

This section describes the configuration management process for a system using COTS. The configuration management system of the COTS supplier is not under the control of ANS configuration management system. The ANS configuration management system should include control of the COTS versions.

2.7.1 COTS Configuration Management Process Objectives

The objectives of the COTS configuration management process are:

a. The COTS specific configuration and data items (for example, software, documentation, adaptation data) are uniquely identified in the ANS software configuration management system.

b. The ANS problem reporting includes the management of problems found in COTS.

c. The ANS change control process ensures that the incorporation of COTS releases is controlled.

d. COTS-specific configuration and data items are included in the ANS archive, retrieval, and release.

2.7.2 COTS Configuration Management Process Activities

The activities associated with configuration management of COTS are:

a. An identification method should be established to ensure that the COTS configuration and data items are uniquely identified.

   Note: The identification method may be based on identification from the COTS supplier and any additional data such as release or delivery date.

b. The ANS problem reporting should include management of problems found in COTS, and a bi-directional problem reporting mechanism with the COTS supplier should be established.

c. The ANS change control process for the incorporation of updated COTS versions should be established.

   An impact analysis of changes to the COTS baseline should be performed prior to incorporation of new releases of COTS.

   Note: The list of changes (problem fixes and new, changed, or deleted functions) implemented in each new release may be available from the COTS supplier.

d. The ANS archival, retrieval, and release should include COTS-specific configuration and data items.

   Note: Consideration may be given to technology obsolescence issues for accessing archived data and escrow issues.

2.8 COTS Quality Assurance

The ANS quality assurance process should also assess the COTS processes and data outputs to obtain assurance that the objectives associated with COTS are satisfied.

Note: It is recommended that the COTS supplier quality assurance is co-ordinated with the ANS quality assurance process where feasible.

2.9 COTS Specific Objectives

The following objectives should be satisfied in addition to the objectives contained in this document for non-COTS software.

N°	Obj	Activity Title	Activity	ISO/IEC 12207	ED109	ED-12B/ DO 178B	IEC 61508	CMMI
1	7.2.1	COTS planning	Acquisition and integral process plans are defined.		. (Ref: 4.1.9 Table 4-1 line 1)			. (Ref: SAM GP 2.2, 3.1)
2	7.2.2	COTS planning	COTS plans are consistent with ANS software plans.		. (Ref: 4.1.9 Table 4-1 line 3)			. (Ref: SAM 2.1 TS 2.4 IPM 1.3)
3	7.2.3	COTS planning	Transition criteria are defined.		. (Ref: 4.1.9 Table 4-1 line 2)			. (Ref: SAM 2.1, 2.4)
4	7.2.4	COTS Acquisition	Adequacy of lifecycle data is determined.		. (Ref: 4.1.9 Table 4-2 line 2)			. (Ref: SAM 2.1, TS 2.4)
5	7.2.5	COTS Acquisition	ANS requirements satisfied by the COTS software is determined.		. (Ref: 4.1.9 Table 4-2 line 1)			. (Ref: SAM 2.1, TS 2.4)
6	7.2.6	COTS Acquisition	Compatibility of COTS with target hardware and other ANS software is assured.		. (Ref: 4.1.9 Table 4-2 line 4)			. (Ref: SAM 2.1, TS 2.4)
7	7.2.7	COTS Acquisition	Derived* requirements are defined		. (Ref: 4.1.9 Table 4-2 line 3)			. (Ref: TS 2.4, RD 2.2, 3.4)
8	7.2.8	COTS Configuration Management	COTS configuration and data items are archived.		. (Ref: 4.1.9 Table 4-3 line 1)			. (Ref: CM 1.2)

N°	Obj	Activity Title	Activity	ISO/IEC 12207	ED109	ED-12B/ DO 178B	IEC 61508	CMMI
9	7.2.9	COTS Configuration Management	COTS configuration and data items are identified.		. (Ref: 4.1.9 Table 4-3 line 4)			. (Ref: CM 1.1, SAM GP 2.6)
10	7.2.10	COTS Configuration Management	COTS problem reporting is established.		. (Ref: 4.1.9 Table 4-3 line 2)			. (Ref: SAM 2.3 CM 2.1)
11	7.2.11	COTS Configuration Management	Incorporation of COTS release is controlled.		. (Ref: 4.1.9 Table 4-3 line 3)			. (Ref: CM 2.2)

*: COTS Derived requirements are defined in this Chapter 5 in §3.5.2.c

End of Part I

PART II

SOFTWARE STANDARDS COVERAGE

This page is intentionaly left blank.

TABLE OF CONTENTS

PART II – SOFTWARE STANDARDS COVERAGE

Chapter 1 ED12B/DO178B Coverage

Chapter 2 IEC61508 Coverage

INTRODUCTION

This Part provides coverage, traceability matrices between:

- On one hand: the five selected standards:
  - ED12B/DO178B
  - IEC 61508,
  - ISO/IEC 12207
  - ED109/DO 278
  - CMMI.
- And on the other hand: the recommended ANS software lifecycle detailed in Part I - Chapter 2 of this document.

The purpose of these coverage matrices is to provide some means to assess, which activities are commonly recommended by a standard and the recommended ANS software lifecycle. These matrices intend to help any organisation to identify discrepancies, omissions, gaps between their own practices (based on either one of these standards) and the recommended ANS software lifecycle.

These matrices also help to identify the problems when applying one of these standards (either generic or not dedicated, customised, tailored to ANS) to ANS.

Please note that the column “Coverage” has to be understood as coverage versus Part I of this document. For example, 2-3.1 in the coverage column means that the standard requirement is covered by Part I Chapter 2 section 3.1.

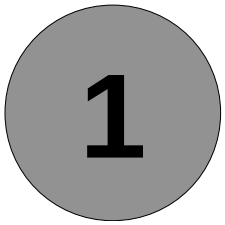
So, when a “N” is noted in this column, it means that this task is not covered by Part I of this document. So it means that this task is not recommended as part of an ANS software lifecycle.

This chapter identifies:

- A status of standards coverage (versus Part I);
- Major omissions of each standard.

Warning: when a standard objective/activity/task/requirement/evidence is noted “covered” by the recommended ANS software life cycle, this does not mean that this objective/activity/task/requirement/evidence is applicable whatever the Assurance Level. This gradation against Assurance Level will be defined in the EATMP “Recommendations for ANS Software” document.

This page is intentionally left blank.



ED12B/DO178B COVERAGE

1. ED12B/DO 178B OBJECTIVES COVERAGE MATRIX

This matrix intends to identify the applicability of ED12B/DO178B "**airborne**" (precisely for airworthiness) objectives to ANS (Part I of this document, which recommends a set of ANS software lifecycle processes).

ED12B/DO178B objectives are listed in Annex A of ED12B/DO178B document.

Some assumptions shall be stated to understand how ED12B/DO178B objectives applicability to **ANS** has been assessed.

ASSUMPTIONS:

· **COTS** (Commercial Off The Shelf): the following approach has been elaborated:

- COTS can be considered **for ANS** as a black box for level D. It means that the COTS features are identified and verified/approved in the frame of the whole application.

- For level C, COTS'vendors shall co-operate also to provide evidences. However the level of evidences remains to be further defined, because the evidences required by ED12B/DO178B are not applicable to certain major COTS used within ANS software applications (ex: Unix for Workstation, X-Windows, ..)

  This assumption (COTS'vendor co-operation) shall be more considered as a goal. However, sometimes this may not be achieved up to the level requested by ED12B/DO178B.

The main difficulties when applying ED12B/DO178B to COTS originate from (non-exhaustive list):

- The need to collect evidences throughout the COTS development process, (so information provided by the COTS vendor)
- The access to source code (not always to be sold),
- Low-level requirements, (difficulty to define the level of refinement of these low-level requirements for COTS which leads to a balance between traceability and compliance)
- Robustness ...

RECOMMENDATION:

When the COTS'vendor co-operation does not allow to comply straight forward with ED12B/DO178B *objectives*, then either new means of compliance have to be proposed to comply with these objectives or some alternative objectives have to be fulfilled to provide an equivalent level of safety.

Some alternative means of compliance for COTS can be the following (non-exhaustive list proposed by D012 Frequently Asked Question to EUROCAE/RTCA WG52/SC190):

- Process Recognition: Process Recognition is the acceptance of the evidence that a development process was applied to a PDS product.
- Prior product qualification: Prior product qualification may occur where the COTS is already certified or qualified to an accepted government or industry standard. Examples of product qualification, which may be used, include avionics, security certifications, medical device certifications, military application qualifications, and nuclear industry certifications.
- Reverse Engineering: Reverse engineering is the process of generating higher level artifacts from existing artifacts such as source code from object code, or executable code.
- Restriction of functionality: The concept "restriction of functionality" involves restricting the use of a COTS to a subset of its functionality, by methods such as, but not limited to, run-time checks, design practices (e.g., company design and code standards), or build-time restrictions. The restricted set of functional requirements may then become the basis for use of the COTS.
- Product service history: the utilization of previous field experience of the product

- Formal Methods: descriptive notations and analytical methods used to construct, develop and reason about mathematical models of system behavior

- Audits and inspections: a mean by which one can determine a process has been adequately performed.

Some other alternative means of compliance or even alternative objectives in practice in ANS systems consist in:

- Long-Run/Trial testing: testing a component/software during a long period (to be defined) under a predefined load,

- Testing at the System level: for example intensive testing under a predefined level of load,

- Ghosting phase: for example, 1 year of operational use of the system as a back-up, …

- For levels A & B, COTS are considered as developed software. It means that the COTS'supplier shall co-operate to provide evidences. However some further investigations shall propose some new alternative means of compliance.

Finally, ED109/DO 278 (How to apply ED12B to ground CNS/ATM software will propose an approach to address COTS when using ED12B.

· **HMI** (Human Machine Interface):

- Problems can be reached due to the large number of objects to be displayed, to the use of tools generating code, to the use of non-deterministic COTS. So low-level requirements implementation, traceability to source code can be difficult to achieve, as well as robustness.

- Attention must be paid to the HMI related requirements (especially as far as verifiability is concerned).

1.1 SOFTWARE PLANNING PROCESS OBJECTIVES

ED 12B §	Topic	Coverage	Rationale
A-1.1	SW development and integral processes activities are defined.	2-3/2-3.1	OK for ANS
A-1.2	Transition criteria, interrelationships and sequencing among processes are defined.	2-3.1.1	OK, but Problem for COTS (level C)
A-1.3	SW lifecycle environment is defined.	2-3.1.1 5-1.1	OK, but Problem for COTS (level C)
A-1.4	Additional considerations are addressed.	N	OBJECTIVE TO BE REDESIGNED
A-1.5	SW development standards are defined.	2-3.1.1	OK, but Problem for COTS. Other alternative means of compliance required.
A-1.6	SW plans comply with this document.	2-3.1	OK, but Problem for COTS (level C)
A-1.7	SW plans are co-ordinated.	2-3.1	OK, but Problem for COTS

1.2 SOFTWARE DEVELOPMENT PROCESS OBJECTIVES

ED 12B §	Topic	Coverage	Rationale
A-2.1	High-level requirements are developed.	2-3.4	To be completed for ANS.
A-2.2	Derived high-level requirements are defined.	2-3.4	OK, but Problem for COTS (level C)
A-2.3	SW architecture is developed.	2-3.5 2-3.6	OK, but Maybe not applicable for level D,
A-2.4	Low-level requirements are developed.	N	Maybe not applicable for level D, Problem for COTS (level C)
A-2.5	Derived low-level requirements are defined.	N	Maybe not applicable for level D, Problem for COTS (level C)
A-2.6	Source code is developed.	2-3.7	This objective implies to buy COTS source. Maybe not applicable for level D, Problem for COTS (level C)
A-2.7	Executable Object Code is produced and integrated in the target computer.	2-3.8 2-3.10	OK

Note: For objectives A-2.3 through A-2.6, the data items produced as a result of these objectives are not requested to be verified by other objectives; therefore, their existence should not be a compliance objective for level D.

1.3 VERIFICATION OF OUPUTS OF SOFTWARE REQUIREMENTS PROCESS

ED 12B §	Topic	Coverage	Rationale
A-3.1	SW high-level requirements comply with system requirements.	3-4.2	OK
A-3.2	High-level requirements are accurate and consistent.	3-4.2	OK
A-3.3	High-level requirements are compatible with target computer.	3-4.2	OK
A-3.4	High-Level requirements are verifiable.	3-4.2	OK, but Problem for level C (HMI)
A-3.5	High-level requirements conform to standards.	3-4.2	OK
A-3.6	High-level requirements are traceable to system requirements.	3-4.2	OK
A-3.7	Algorithms are accurate.	3-4.2	OK (To Be Clarified)

1.4 VERIFICATION OF OUTPUTS OF SOFTWARE DESIGN PROCESS

The concept of high-level requirements and low-level requirements has not been kept in the recommended ANS software lifecycle. Only software requirements (with many levels of them) have been defined mainly due to the difficulty to fulfil objectives stating that low-level requirements must be traceable down to source or executable code when using COTS with limited co-operation of the COTS'supplier and because of code generating tools (HMI).

So in the following table "OK" means that this objective is requested to be met in the recommended ANS software life cycle but with a prior customisation and mainly for new developed software and for ED12B Assurance Levels A & B.

ED 12B §	Topic	Coverage	Rationale
A-4.1	Low-level requirements comply with high-level requirements.	N (3-4.2)	Problem of low-level requirements for level C (COTS, HMI)
A-4.2	Low-level requirements are accurate and consistent.	N (3-4.2)	Problem of low-level requirements for level C (COTS, HMI)
A-4.3	Low-level requirements are compatible with target computer.	3-4.2	Problem of low-level requirements for level C (COTS, HMI)
A-4.4	Low-level requirements are verifiable.	3-4.2	Problem of low-level requirements for level C (COTS, HMI)
A-4.5	Low-level requirements conform to standards.	3-4.2	Problem of low-level requirements for level C (COTS, HMI)
A-4.6	Low-level requirements are traceable to high-level requirements.	3-4.2	OK, but problem of low-level requirements for COTS, HMI (level C)
A-4.7	Algorithms are accurate.	3-4.2	OK (To Be Clarified)
A-4.8	SW architecture is compatible with high-level requirements.	3-4.2	OK
A-4.9	SW architecture is consistent.	3-4.2	OK, but Problem for level C (COTS)
A-4.10	SW architecture is compatible with target computer.	3-4.2	OK
A-4.11	SW architecture is verifiable.	3-4.2	OK
A-4.12	SW architecture conforms to standard.	3-4.2	OK, but Problem for level C (COTS)

ED 12B §	Topic	Coverage	Rationale
A-4.13	SW partitioning integrity is confirmed.	3-4.2	OK, but Problem for level C & D (COTS)

1.5 VERIFICATION OF OUTPUTS OF SOFTWARE CODING & INTEGRATION PROCESS

ED 12B §	Topic	Coverage	Rationale
A-5.1	Source code complies with low-level requirements.	3-4.2	Problem of low-level requirements for level C (COTS, HMI)
A-5.2	Source code complies with SW architecture.	3-4.2	OK, but problem of low-level requirements for level C (COTS, HMI)
A-5.3	Source code is verifiable.	3-4.2	OK, problem to access source code for COTS, HMI (level C)
A-5.4	Source code conforms to standards.	3-4.2	OK, but Problem for level C (COTS, HMI)
A-5.5	Source code is traceable to low-level requirements.	(3-4.2)	Problem of low-level requirements for level C (COTS, HMI)
A-5.6	Source code is accurate and consistent.	3-4.2	OK, but problem of low-level requirements for level C (COTS, HMI)
A-5.7	Output of software Integration Process is complete and correct.	3-4.2	OK, but problem of low-level requirements for level C (COTS, HMI)

1.6 TESTING OF OUTPUTS OF INTEGRATION PROCESS

ED 12B §	Topic	Coverage	Rationale
A-6.1	Executable Object Code complies with high-level requirements.	3-4.2	OK
A-6.2	Executable Object Code is robust with high-level requirements.	3-4.2	OK, but problem of robustness for level C & D (COTS, HMI)
A-6.3	Executable Object Code complies with low-level requirements.	3-4.2	Problem of low-level requirements for level C (COTS, HMI)
A-6.4	Executable Object Code is robust with low-level requirements.	3-4.2	Problem of low-level requirements for level C (COTS, HMI)
A-6.5	Executable Object Code is compatible with target computer.	3-4.2	OK

1.7 VERIFICATION OF VERIFICATION PROCESS RESULTS

ED 12B §	Topic	Coverage	Rationale
A-7.1	Test procedures are correct.	3-4.2	OK
A-7.2	Test results are correct and discrepancies explained.	3-4.2	OK
A-7.3	Test coverage of high–level requirements is achieved.	3-4.2	OK, but problem of robustness for level C & D (COTS, HMI)
A-7.4	Test coverage of low–level requirements is achieved.	3-4.2	Problem of low-level requirements for level C (COTS, HMI)
A-7.5	Test coverage of SW structure (modified condition/decision) is achieved.	N	ONLY FOR LEVEL A
A-7.6	Test coverage of SW structure (decision coverage) is achieved.	3-4.2	OK , but problem for COTS, HMI
A-7.7	Test coverage of SW structure (statement coverage) is achieved.	3-4.2	OK, but problem for level C (COTS, HMI)
A-7.8	Test coverage of SW structure (data coupling and control coupling) is achieved.	3-4.2	OK, but problem for level C (COTS, HMI)

1.8 SOFTWARE CONFIGURATION MANAGEMENT PROCESS

ED 12B §	Topic	Coverage	Rationale
A-8.1	Configuration items are identified.	3-2	OK
A-8.2	Baselines and traceability are established.	3-2	OK
A-8.3	Problem reporting, change control, change review and configuration status accounting are established.	3-2	OK, attention must be paid to baseline and configuration item traceability.
A-8.4	Archive, retrieval and release are established.	3-2	OK
A-8.5	Software load control is established.	3-2	OK To Be Checked
A-8.6	Software life cycle environment control is established.	3-2 4-2	OK, With clarifications to categories meaning.

Note: The incorporation of two levels of data control (CC1 & CC2) is designed to allow the developer flexibility. Individual companies must define the attributes of the control categories (e.g. retention times, signature requirements, etc.) for themselves. An example of how this might work is to control CC2 data until the approval for the current development is obtained. Upon approval, CC2 data may be archived for historical records or destroyed. Recognise that opting for a single control category (CC1) drives cost and illogical requirements such as problem reports written for errors discovered within other problem reports.

1.9 SOFTWARE QUALITY ASSURANCE PROCESS

ED 12B §	Topic	Coverage	Rationale
A-9.1	Assurance is obtained that SW development and integral processes comply with approved SW plans and standards.	3-3 3-6	OK
A-9.2	Assurance is obtained that transition criteria for the SW lifecycle processes are satisfied.	3-3 3-6	OK
A-9.3	SW conformity review is conducted.	3-3 3-6	OK, Limitations for levels C & D

1.10 CERTIFICATION LIAISON PROCESS

As certification is not applicable to ANS the following table will not be referenced to the “recommended” software lifecycle described in the Part I - Chapter 2 of this document. However, some attention has to be paid to these airborne activities which could inspire the approval/acceptance process for ANS.

ED 12B §	Topic	Coverage	Rationale
A-.10.1	Communication and understanding between the applicant and the certification authority is established.	N	TO BE TAILORED TO ANS
A-.10.2	The means of compliance is proposed and agreement with the Plan for SW Aspects of certification is obtained.	N	TO BE TAILORED TO ANS
A-.10.3	Compliance substantiation is provided.	N	NOT APPLICABLE TO ANS

2. ED12B/DO178B STANDARD COVERAGE

2.1 SYSTEMS ASPECTS RELATING TO SOFTWARE DEVELOPMENT

Note that this ED12B/DO178B chapter does not identify any objective or requirement or mean of compliance. This chapter was mainly dedicated to compensate the lack of existing system standard (such as ARP4754, which has been written after ED12B/DO178B).

ED 12B §	§ Purpose	Topic	Details	Coverage	Rationale
2.1.1	Information flow from system to software processes	The System Safety Assessment process defines the safety related requirements to be implemented by the SW	System safety requirements are inputs to the software life cycle e.g.: criticality, software level, safety strategies and design constraints	2-3.2	System requirements analysis
2.1.2	Information flow from software to system processes	The System Safety Assessment Process determines the impact of the SW design and implementation on system safety using information provided by the SW life cycle process	Data includes: fault containment boundaries, software requirements, and error sources detected or eliminated through software architecture.	1-3.3	Software Safety Assessment
2.2	Failure Condition and SW Development Assurance Level	Relationship between SW levels and system failure condition categories need to be established	The severity of a failure condition determines its category : Catastrophic, Hazardous, Major, Minor and No Effect (2.2.1) The software levels are associated to these categories: they are assigned during the System Safety Assessment depending on the potential contribution of software to system failure condition(s). Software levels A, B, C, D, and E corresponds to the above failure condition categories (2.2.2) The standard provides guidance on software level definition. (2.2.3)	1-2 2-3.1 2-3.2	Software Safety Assessment Development plan System safety requirements
2.3.1	System architectural Considerations	Several architectural strategies may limit the impact of errors, or detect errors and provide acceptable system responses to contain errors	Partitioning	1-3.3 2-3.3	Software Safety Assessment
2.3.2	System architectural Considerations	Several architectural strategies may limit the impact of errors, or detect errors and provide acceptable system responses to contain errors	Multiple version dissimilar SW	N	Not applicable to ANS
2.3.3	System architectural Considerations	Several architectural strategies may limit the impact of errors, or detect errors and provide acceptable system responses to contain errors	Safety monitoring	2-3.3 3-4.1 1-3.3	Software Safety Assessment
2.4	Guidance for different software types	Specific recommendations could be made for user-modifiable software		N	
2.4	Guidance for different software types	Specific recommendations could be made option-selectable software		N	
2.4	Guidance for different software types	Specific recommendations could be made for COTS SW		5.3	TO BE CUSTOMISED TO ANS
2.5	System considerations for field-loadable SW	Specific recommendations could be made for field-loadable software		N	
2.6	System requirements considerations for SW Verification			3-4.2	

ED 12B §	§ Purpose	Topic	Details	Coverage	Rationale
2.7	SW considerations in system verification.		System verification is not covered but system verification may provide a significant coverage of the code structure.	2-3.11	

2.2 SOFTWARE LIFE CYCLE

ED 12B §	§ Purpose	Topic	Details	Coverage	Rationale
3.0	General requirements for the software life cycle(s) selection	A life cycle specifies a set of distinct but interacting processes.	For each software element, a life cycle should be selected. The standard does not refer to a preferred type of life cycle.	2-3.1	
3.1	SW life cycle processes identification	All required life cycle processes need to be identified and specified	Planning process	2-2	
3.1	SW life cycle processes identification	All required life cycle processes need to be identified and specified	Development process: Requirements, design, coding, integration	2-3	
3.1	SW life cycle processes identification	All required life cycle processes need to be identified and specified	integral processes: Verification	3-4	
3.1	SW life cycle processes identification	All required life cycle processes need to be identified and specified	Integral processes : Configuration management	3-2	
3.1	SW life cycle processes identification	All required life cycle processes need to be identified and specified	Integral processes : Quality assurance	3-3	
3.1	SW life cycle processes identification	All required life cycle processes need to be identified and specified	Integral processes: Certification liaison	N	NOT APPLICABLE TO ANS
3.2	SW Life cycle Definition	The activities associated with each process, their chronology and the responsibilities for performing them should be specified.	The life cycle activities should be specified. The sequencing of processes depends on the project and processes may be iterative. Example of life cycle of software are given including: previously developed software, partitioned function and prototyping	2-3.1	
3.3	Transition criteria between processes	Transition criteria between processes should be established. These criteria are used to decide to terminate a given process.	Criteria depend on planning and software level. Examples are given. Feedback from other processes and partial inputs are considered. The software plans should define the transition criteria (4.3.b).	2-3.1.1	

2.3 SOFTWARE PLANNING PROCESS

ED 12B §	§ Purpose	Topic	Details	Coverage	Rationale

ED 12B §	§ Purpose	Topic	Details	Coverage	Rationale
4.1	SW planning process objectives	The strategy for SW development should be defined to the extent required by the SW criticality	It is required to define the means of producing software, which will satisfy the system requirements and provide confidence, which is consistent with airworthiness requirements. (table A.1 in annex A)	2-3.1	
4.2	SW planning process activities	Guidance for SW planning process		2-3.1	
4.3	Plans		Plan for Software Aspects Of Certification	N	NOT APPLICABLE TO ANS
4.3	Plans		Software Development Plan	2-3.1.1 3-1	
4.3	Plans		Software Verification Plan	3-4.1 3-1	
4.3	Plans		Software Configuration Management Plan	3-2 3-1	
4.3	Plans		Software Quality Assurance Plan	3-3 3-1	
4.4	Life cycle environment planning	To define the methods, tools procedures, programming languages and hardware used to develop, verify and control the SW products and data.	The standard addresses - the selection of the SW development environment - the selection of programming language and compiler - the selection of a SW test environment - the selection of SW development standards The standard identifies the major objectives of such selection: - to limit the opportunity for introducing errors during the development process - to improve the detection of errors in the verification process - to include safety features, such as fault tolerance. It does not require the use of any specific method or technique.	4-2 3-1	
4.5	Development standards	Rules and constraints for the development process and its consistency should be specified in terms of development standards.	These standards define the methods, rules and tools to be used to develop the high-level requirements (requirements standards; 11.6), the software architecture and low-level requirements (design standards; 11.7) and to code the software (code standards; 11.8). They are in compliance with the safety-related requirements and are a basis for the verification process.	2-3.1	
4.6	Review and assurance of the planning process	To ensure that the plans and standards comply with all requirements and that means are provided to execute them	The review and assurance of the planning process shall ensure that : - methods are compliant with the objectives -life cycle processes can be applied consistently - each process produces evidence that its outputs can be traced to their activity and inputs.	2-2 3-6	

2.3.1 SOFTWARE LIFE CYCLE ENVIRONMENT PLANNING

ED 12B §	§ Purpose	Topic	Details	Coverage	Rationale
4.4.1	SW Development Environment	To establish the development environment.	Qualified tools to minimise the risk to the final software should be chosen. A verification process and standards in agreement with the software level should be developed. An error introduced by one part of the environment should be detected by another part. Specific cases are analysed: tools in combination and optional features of software tools.	5-1.1 (2-3.1 4-2)	
4.4.2	Language And compiler Considerations	To take into account the choice of compiler and language in the software planning and verification activities.	The standard highlights the need to carefully consider the language and compiler which may impair the traceability between the source code and the object code. Planning process should provide means to ensure verification coverage and define the means in the appropriate plan. The Planning process should consider the particular features and changes of the programming language and compiler.	5-1.1 (2-3.1 4-2)	
4.4.3	SW Test environment	Qualified tools, methods, procedures and hardware to test the outputs of the integration process should be chosen.	Emulator and simulator should be qualified as defined in 12.2. In case of differences between the target computer and the emulator or the simulator, the ability to detect potential errors should be considered and detection of those errors should be provided by other software verification process activities.(12.2) The test should be performed in the integrated target computer, since errors are only detected in it. (6.4.1)	5-1.2 (2-3.1 4-2)	

2.4 SOFTWARE DEVELOPMENT PROCESS

ED 12B§	§ Purpose	Topic	Details	Coverage	Rationale
5	General requirements	The software development process includes the following sub-process: -SW requirements Specification - SW design -SW coding - SW integration	The standard identifies the notion of requirements (high level requirements issued from specification, low level requirements issued from design, derived (non directly traceable to specification or design)). The document requires that standards are set and are followed. Traceability shall be ensured at all stages. Post-certification modifications without re-entering the certification process are not allowed. The development activities and the independence of those performing them are graded against software level (see table A.2 in annex 1).	2-3	The difference between high and low level requirements does not exist anymore in the SW standards comparison. **See for independence applicability (definition of independence to be validated)**

ED 12B§	§ Purpose	Topic	Details	Coverage	Rationale
5.1	Software requirements process	From the outputs of the system life cycle, this process develops the software requirements data.	Objectives are: - high-level requirements are developed (functional, performance, interface and safety-related) -derived high-level requirements are indicated to the system safety assessment process. Software requirements data are produced (11.9)	2-3.4 1-3-3	
5.2	Software Design General Requirement		The objectives are: - software architecture and low-level requirements are derived from high-level requirements - derived low-level requirements are provided to the system safety assessment process. There is a guidance for designing for user-modifiable software. Design data shall be produced. (11.10)	2-3.5 2-3.6 1-3.3 2-3.8	The concept of low-level requirements has not been kept in the Chapter 3 (ANS software lifecycle) mainly because of COTS, code generating tools. Patch & deactivated code
5.3	Coding process	From the software architecture and low-level requirements, the source code and the object code shall be developed.	Objective is : Source code is developed that is traceable, verifiable, consistent and correctly implements low level requirements. Outputs of the process are source code (11.11) and object code.	2-3.7	
5.4	Integration process	Executable object code is generated from the source code and loaded into the target hardware. The integration consists of software integration and system integration.	Objective of the integration process is : The executable object code is loaded into the target hardware for hardware/software integration. The integration consists of software integration and hardware/software integration. Considerations for deactivated code and software patches are given. Patches should not be used without re-certification of the software.	2-3.8 2-3.10	
5.5	Traceability	There should be a traceability between system specification, high- and low-level software requirements and source code.	Traceability is systematically required.	2-3	

2.5 SOFTWARE VERIFICATION PROCESS

ED 12B §	§ Purpose	ED-12B/DO 178B	Details	Coverage	Rationale
6.1	SW Verification process objectives	Technical assessment of the results of both the software development and verification processes with the aim of detecting and reporting errors and with some level of independence.	To detect and report errors that may have been introduced during the software development processes. Verification is not simply testing and includes a combination of reviews, analyses, development of test cases and procedures, execution of those test	3-4	

ED 12B §	§ Purpose	ED-12B/DO 178B	Details	Coverage	Rationale
			procedures. The verification activities and the independence of those performing them are graded against software level. (See tables A.3 through A.7 in annex A)		
6.2	Activities	Combination of reviews, analysis, development of test cases and execution of test procedures. The verification process is performed as planned in the verification plan.	To assess accuracy, completeness and verifiability of the software requirements, architecture and source code (see reviews and analysis), and then to test the compliance with the requirements (see testing process). Guidance on outputs is provided. (11.13 & 11.14)	3-4	

2.5.1 SOFTWARE REVIEWS AND ANALYSIS

ED 12B §	§ Purpose	Topic	Details	Coverage	Rationale
6.3	General requirements	Analysis provide a repeatable evidence of correctness, reviews provide a qualitative assessment of correctness.	This subsection provides requirements for the reviews and analysis of high level requirements, low-level requirements, software architecture, software code. Outputs are recorded in the software verification results. The reviews and analysis, and the independence of those performing them are graded against software level. (See tables A.3 through A.5 in annex A)	3-4.2 3-6	
6.3.1	Review and analysis of high-level requirements	To detect and report requirements errors that may have been introduced during the software requirements process.		3-4.2	
6.3.2	Review and analysis of low-level requirements	To detect and report requirements errors that may have been introduced during the software design process.		3-4.2	No differentiation between high and low requirements in “COMPARISON”
6.3.3	Review and analysis of software architecture	To ensure that software architecture complies with the high-level requirements.	The process shall verify that architecture is consistent, verifiable and compatible with the target computer. It shall comply with the design standards. Guidance on partitioning integrity	3-4.2	Check how to apply Partitioning integrity
6.3.4	Review and analysis of the Source Code	To ensure that source code complies with the low-level requirements and the software architecture.	The process shall verify that the source code is verifiable, traceable and consistent. It shall comply with the code standards.	3-4.2	Traceability to requirements to be customised (COTS, HMI)
6.3.5	Review and analysis of integration process	To ensure that the integration process results are complete and correct.	Objective may be performed by a detailed examination of the linking and loading data and memory map.	3-4.2	
6.3.6	Review and analysis of	To ensure that code testing	Test cases, test procedures and	3-4.2	

ED 12B §	§ Purpose	Topic	Details	Coverage	Rationale
	the test cases, procedures and results	was developed and performed accurately and completely.	test results shall be reviewed.		

2.5.2 SOFWARE TESTING PROCESS

ED 12B §	§ Purpose	Topic	Details	Coverage	Rationale
6.4.1	Test environment	To identify and define test environments necessary to test SW as exhaustively as possible.	Different test environment may be considered: simulation, emulation, using target HW.	5-1.2	
6.4.2.1	Requirements-based test case selection	To establish requirements-based test cases using normal range test for normal inputs and conditions.	Test cases should be based primarily on the software requirements. Guidance is given to design them. The testing activities and the independence of those performing them are graded against software level. (See tables A.6 and A.7 in annex A)	3-4.2	
6.4.2.2	Requirements-based test case selection	To establish requirements-based test cases using robustness test cases for abnormal inputs and conditions.	Test cases should be based primarily on the software requirements. Guidance is given to design them.. The testing activities and the independence of those performing them are graded against software level. (See tables A.6 and A.7 in annex A)	3-4.2	
6.4.3	Requirements-based testing methods	To execute the three testing phases: -module testing, -software integration testing and -hardware/software integration testing.	The requirements-based testing methods are: -hardware/software integration testing -software integration testing -low-level testing. The hardware/software integration testing requires a specific environment or strategy. Testing the executable object code is not required for the lowest software level (see table A.6 in annex A).	3-4.2	
6.4.4.1	Test coverage analysis	Requirements-based test coverage analysis	The standard requires two test coverages, the first is: -requirements-based test coverage analysis (to determine how well the testing verified the implementation of the software requirements). They accomplish traceability between the implementation of the software requirements and their verification. Case of the highest software level and of unexecuted code.	3-4.2	
6.4.4.2	Test coverage analysis	Structural coverage analysis	The standard requires two test coverages, the second is: -structural coverage analysis (to determine how well the testing verified the code structure). They accomplish traceability between the implementation of the software requirements and their verification. Case of the highest software level and of unexecuted code.	3-4.2	CHECK how to apply to ANS (COTS, HMI)

ED 12B §	§ Purpose	Topic	Details	Coverage	Rationale
6.4.4.3	Test coverage analysis	Structural coverage analysis resolution	As structural coverage analysis may reveal code structure not tested, resolution requires additional verification.	3-4.2	CHECK how to apply to ANS (COTS, HMI)

2.6 SOFTWARE CONFIGURATION MANAGEMENT PROCESS

ED 12B §	§ Purpose	Topic	Details	Coverage	Rationale
7.1	SW Configuration Management process objectives	To define and control software configuration, manage configuration changes, control process inputs and outputs, establish baselines and aid the verification	Software configuration management (SCM) is applied in agreement with the planning process and the software configuration management plan. The SCM process includes the activities of configuration identification, change control, baseline establishment, and archiving of the software product, including the related software life cycle data. The SCM process does not stop when the product is accepted by the certification authority but continues throughout the service life. The depth of the SCM control (CC1 or CC2) placed on the data is specified in subsection 7.3. The configuration management activities and the independence of those performing them are graded against software level (See table A.8 in annex A).	3-2	
7.2.1	Configuration identification	To label unambiguously each configuration item and its successive versions so that a basis is established for the control and reference of configuration items.	Configuration identification should be done for life cycle data and for each configuration item. Configuration identification should be done before the use of configuration items and before implementation of change control and traceability data recording.	3-2	ED12B requests the identification at the level of COMPONENT
7.2.2	Baselines and traceability	To define a basis for further software life cycle activity and allow reference to, control of, and traceability between configuration items.	Baselines should be established for items used for certification credit. A baseline for the software product should be established and defined in an index (11.16). Baseline should be protected from change.	3-2	ED12B requests the baseline traceability and configuration item traceability. These 2 features applicability are to be checked for ANS (COTS).
7.2.3	Problem reporting, tracking and corrective action	To record and resolve process non-compliance with plan and standards, deficiencies of outputs and anomalous behaviour of products.	Problem resolution should be ensured in establishing reports. Problem reports that require corrective action of the software product or outputs of software life cycle processes should invoke the change control activity. Guidance on reports is given (11.17)	3-2 3-8	
7.2.4	Change control	Recording, evaluation, resolution and approval of changes throughout the life cycle.	Configuration items and baselines are protected against changes. There should be traceability between changes and their origin. Throughout the change activity, data affected by the change should be updated and records should be maintained.	3-2 3-8	
7.2.5	Change review	To ensure that changes	Confirmation that affected configuration	3-2	

ED 12B §	§ Purpose	Topic	Details	Coverage	Rationale
		are assessed, approved or disapproved and to control feedback	items are configuration identified. Feedback about safety-related changes is provided to the system safety assessment process.	3-8	
7.2.6	Configuration status accounting	To provide the status and history of configuration items.	The objective of the status accounting activity is to provide data for the configuration management with respect to configuration identification, baselines, problem reports, and change control.	3-2	
7.2.7	Archive, retrieval and release	To process life cycle data so that they could be retrieved and duplicated without errors and their integrity could be ensured. To control that only authorised software is used.	Archive and retrieval activities aim at ensuring that the life cycle data associated with the software product can be retrieved in case of a need to duplicate, regenerate, retest or modify the software product. Release activities aim at ensuring that only authorised software is used.	3-2	
7.2.8	Software load control	To ensure that the executable object code is loaded into the system with appropriate safeguards.	Procedures for part numbering and media identification shall be implemented Records should be kept that confirm software compatibility with the airborne system or hardware. Considerations about field-loadable software are provided. (2.5)	3-2	
7.2.9	Software life cycle environment control	To ensure that the tools used to produce (develop, control, build, verify and load) the software are identified, controlled and retrievable.	Configuration identification should be established for the executable object code of the tools used to develop, control, build, verify and load the software. Control Categories CC1 and CC2 apply to the qualified tools. At least CC2 is applied to the other tools.	3-2 4-2	
7.3	Data control categories	SW life cycle data can be assigned to one of two categories.	These categories are related to the configuration management controls placed on the data.	3-2	

2.7 SOFTWARE QUALITY ASSURANCE PROCESS

ED 12B §	§ Purpose	Topic	Details	Coverage	Rationale
8.1	SW Quality Assurance Process Objectives	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 the approved software plans and standards. The process is applied as defined by the software quality assurance plan.	Objectives of SQA is to obtain assurance that : - Software development processes and integral processes comply with approved plans and standards. - Transition criteria are satisfied - A conformity review of the software is conducted. The SQA activities and the independence of those performing them are graded against software level (see table A.9)	3-3 3-6	
8.2	SW Quality Assurance Process Activities	To perform audits and carry out all quality assurance procedures.	To provide assurance, with authority and independence, that plans and standards are developed and reviewed, that life cycle processes and products comply with all plans and standards by means of audits.	3-3 3-6 3-7	
8.3	SW Conformity	To obtain assurance, prior to the	Activities of the review are detailed.	3-3	Some activities are to be

ED 12B §	§ Purpose	Topic	Details	Coverage	Rationale
	Review	delivery of software products submitted as part of a certification application, that the software life cycle processes are complete, software life cycle data are complete, and the executable object code is controlled and can be regenerated.		3-6	checked for applicability to ANS (f:COTS, I)

2.8 CERTIFICATION LIAISON PROCESS

ED 12B §	§ Purpose	Topic	Details	Coverage	Rationale
9	Means of compliance and planning	To establish communication and understanding between the applicant and the certification authority throughout the software life cycle to assist the certification process.	The process is applied as defined by the planning process and the plan for software aspects of certification. Certification activities and the independence of those performing them are graded against software level (see table A10 in annex A).	N	CERTIFICATION NOT APPLICABLE TO ANS
9.1	Data submitted to the certification authority	To obtain agreement with the certification authority on this plan	To submit the plan and other requested data to the certification authority for review at a point in time when the effects of changes are minimal, to resolve issues identified by the certification authority.	N	CERTIFICATION NOT APPLICABLE TO ANS
9.2	Compliance substantiation	To provide evidence that the life cycle processes satisfy the plans	To arrange review of the life cycle processes, to submit the software accomplishment summary, the configuration index and other requested data, to resolve issues raised by the certification authority as a result of its reviews.	N	CERTIFICATION NOT APPLICABLE TO ANS
9.3	Minimum SW life cycle data submitted to Certification Authority	Minimum list of documents to be submitted.	Plan for SW aspects of certification SW configuration index SW accomplishment summary	N	CERTIFICATION NOT APPLICABLE TO ANS

2.9 OVERVIEW OF AIRCRAFT AND ENGINE CERTIFICATION

ED 12B §	§ Purpose	Topic	Details	Coverage	Rationale
10.1	Certification basis	Legal recognition by the certification authority that the software complies with the requirements.	The certification authority considers the software as a part of the system and does not approve it as a stand-alone product.	N	CERTIFICATION NOT APPLICABLE TO ANS
10.2	SW aspects of certification	Certification authority assesses the plan for SW aspects of certification for completeness and consistency with the means of		N	CERTIFICATION NOT APPLICABLE TO ANS

ED 12B §	§ Purpose	Topic	Details	Coverage	Rationale
		compliance.			
10.3	Compliance determination	Use of SW Accomplishment Summary to determine certification.		N	CERTIFICATION NOT APPLICABLE TO ANS

2.10 SOFTWARE LIFE CYCLE DATA

ED 12B §	§ Purpose	Topic	Details	Coverage	Rationale
11.0	SW life cycle data	Characteristics required.	Unambiguous, complete, verifiable, consistent, modifiable, traceable, form, control	3-1	
11.1	Plan for SW aspects of certification	Characteristics required.	System overview, SW overview, certification considerations, SW life cycle, SW life cycle data, schedule, additional considerations.	N	CERTIFICATION NOT APPLICABLE TO ANS
11.2	SW development plan	Characteristics required.	Standards, SW life cycle, SW development environment.	2-3.1.1 4-2	
11.3	SW verification plan	Characteristics required.	Organisation, independence, verification methods, verification environment, transition criteria, partitioning considerations, compiler assumptions, reverification guidelines, previously developed SW, multiple version dissimilar SW	3-4	
11.4	SW configuration management plan	Characteristics required.	Environment, SCM activities, transition criteria, SCM data, supplier control.	3-2	
11.5	SW quality assurance plan	Characteristics required.	Environment, authority, SQA activities, transition criteria, timing, SQA records, supplier control.	3-3	
11.6	SW requirements standards	Characteristics required.	Structured methods, notations to be used, use of tools, derive requirements to system process.	2-3.4	
11.7	SW design standards	Characteristics required.	Methods, naming conventions, use of tools, constraints, complexity restrictions.	2-3.5 2-3.6	
11.8	SW code standards	Characteristics required.	Syntax definition, source code presentation, naming conventions, constraints.	2-3.7	
11.9	SW requirements data	Characteristics required.	System requirements allocation to SW, mode of operation, precision, accuracy, timing, memory, HW/SW interfaces, failure detection, safety monitoring, partitioning.	2-3.4	
11.10	Design description	Characteristics required.	Algorithms, data structure, allocation to processors and tasks, input/output description, data flow, control flow, resource limitations, scheduling, inter-task/inter-procedure communications, partitioning.	2-3.5 2-3.6	
11.11	Source code	Characteristics required.	SW identification, version name, version date, compilers instructions to generate object code, linking and loading.	2-3.7	
11.12	Executable object code	Characteristics required.	SW loaded on the target.	2-3.8 2-3.9	

ED 12B §	§ Purpose	Topic	Details	Coverage	Rationale
11.13	SW verification cases and procedures	Characteristics required.	Review and analysis procedures, test cases, test procedures.	3-4.2	
11.14	SW verification results	Characteristics required.	Tests pass/fail results, configuration item identification, coverage and traceability analysis.	3-4.2	
11.15	SW life cycle environment configuration index	Characteristics required.	SW life cycle environment HW, SW development tools, test environment, qualified tools.	4-2	
11.16	SW configuration index	Characteristics required.	SW product, executable object code, source code, previously developed SW, SW life cycle data, archive and release media, instructions for building executable, reference to SW life cycle environment configuration index, data integrity checks.	3-2	
11.17	Problem reports	Characteristics required.	Identification of configuration item, problem description, corrective action description.	3-8	
11.18	SW configuration management records	Characteristics required.	Record of SCM process activities (baseline, configuration identification lists, change history reports, archive and release records, …).	3-2	
11.19	SW quality assurance records	Characteristics required.	Record of SQA process activities(review or audit reports, meeting minutes, conformity review records, …)	3-3	
11.20	SW accomplishment summary	Characteristics required.	Primary item for showing compliance for certification.	N	CERTIFICATION NOT APPLICABLE TO ANS

2.11 ADDITIONAL CONSIDERATIONS

2.11.1 USE OF PREVIOUSLY DEVELOPED SOFTWARE

ED 12B §	§ Purpose	Topic	Details	Coverage	Rationale
12.1	Use of previously developed software	To justify and control the use of previously developed software. To assess the issues associated with the use of previously developed software including modifications, change of installation, change of application environment...	The intention to use such software is stated in the plan for software aspects of requirements. Traceability from product and data of the previous application to the new application should be ensured. In general, the impact of any modification should be assessed against the objectives of the standard.	5-4	
12.1.1	Modification to previously developed SW	Guidance for analysis activities for proposed modifications to previously developed SW.	Review of system safety assessment process outputs, baseline upgrade, impact of requirements and architecture change, data flow and control flow analysis, timing and traceability analysis.	5-4	
12.1.2	Change of Aircraft installation	Guidance for new aircraft installations.	Determination of new SW assurance level and certification basis.	5-4	Customised to ANS
12.1.3	Change of	Guidance for change of	Development environment tools,	5-4	CUSTOMISED for ANS

ED 12B §	§ Purpose	Topic	Details	Coverage	Rationale
	application or development environment	application or development environment when using previously developed SW.	criteria for evaluation, compiler options, HW/SW interface, tests compatibility.		
12.1.4	Upgrading a development baseline	Guidelines intend to aid acceptance of COTS, previously developed SW at lower assurance level, SW developed against other guidelines or prior to existence of these guidelines;	Determination of areas of improvement, reverse engineering, product service history, strategy for upgrading.	5-4	CUSTOMISED for ANS
12.1.5	SW configuration management considerations	Guidance for upgrading SCM process when using previously developed SW.	Traceability from previously developed to new SW. Change control enabling SW components used in more than one application.	5-4	
12.1.6	SW quality assurance considerations	Guidance for upgrading SQA process when using previously developed SW.	Assurance that SW components satisfy SW level of new application, update of SW plan.	5-4	

2.11.2 TOOL QUALIFICATION

ED 12B §	§ Purpose	Topic	Details	Coverage	Rationale
12.2.1	Qualification criteria for development tools	Development tools can introduce errors, therefore, stringent criteria shall be applied to their qualification.	Tools comply with tool operational requirements.	5-1.1 5-1.2	CUSTOMISED for ANS
12.2.2	Qualification criteria for verification tools	The tool to be qualified should satisfy less stringent criteria because a verification tool cannot introduce errors, but may fail to detect them.	Tools comply with tool operational requirements.	5-1.1 5-1.2	CUSTOMISED for ANS
12.2.3	Tool Qualification data	The tool qualification process and data shall be described in a document.	Tool qualification plan, tool operational requirements.	5-1.1 5-1.2	CUSTOMISED for ANS
12.2.4	Tool Qualification agreement	Certification authority gives its agreement to the use of a tool.	Steps and documents to be produced.	5-1.1 5-1.2	CUSTOMISED for ANS

2.11.3 ALTERNATIVE METHODS

ED 12B §	§ Purpose	Topic	Details	Coverage	Rationale
12.3.1	Formal Methods	Method to improve the specification and verification of SW and to prevent and eliminate	Formal methods consider: Levels of design refinement, coverage of SW requirements and	N	

ED 12B §	§ Purpose	Topic	Details	Coverage	Rationale
		requirements, design and code errors throughout SW development process.	SW architecture, degree of rigor.		
12.3.2	Exhaustive input testing	This alternative method can be substituted for a SW verification process activity.	Used for simple and isolated equipment.	N	CHECK How to apply to ANS
12.3.3	Considerations for multiple version dissimilar SW verification	Guidelines concerning SW verification process as it applies to multiple-version dissimilar SW.	Independence, mulitple processor-related verification, source code verification, tool qualification, simulators.	N	NOT APPLICABLE TO ANS
12.3.4	SW reliability models	Use of SW reliability models for certification.	No guidance proposed for SW error rates, because no mature method available.	N	NOT APPLICABLE TO ANS
12.3.5	Product service history	Equivalence of safety for SW demonstrated by the use of SW's product service history.	This method depends on: Configuration management, effectiveness of problem reporting, stability and maturity of SW, relevance of product service history environment, actual error rates, impact of modifications.	N	CHECK How to apply to ANS

3. OMISSIONS OF ED 12B/DO 178B

The purpose of this paragraph is to highlight what is not covered by ED 12B/DO 178B, versus what is identified in Part I of this document.

So the purpose is to identify the objectives that are not addressed by ED 12B/DO 178B though recommended for an ANS Software Life Cycle.

Major ED 12B/DO 178B missing items are as follows:

- Management of functional safety (hazard and risk analysis, identification of safety/non-safety functions, functional safety assessment, safety validation, safety documentation). Due to the development of ED 12B/DO 178B before system safety assessment standard (ARP4754/4761); Chapter 2 of ED 12B/DO 178B, which is informative, provides some limited recommendations on how to perform a system safety assessment.
- Only a part of the safety lifecycle is defined by ED12B/DO178B (the part concerned with the development of software). No requirements are set concerning acquisition, maintenance, operation;
- Life cycle activities scheduling;

- Validation activities are not covered as ED 12B/DO 178B is certification oriented;
- Integration of the software product into the system on site;
- Software integration testing is not defined concurrently with the design/development phases;
- Configuration/adaptation data definition standard & verification is not considered (because not used by avionics software);
- Requirements to choose a programming language;
- Staff training, staff competence;
- Capacity for safe modifications (A margin for throughput (e.g., Input and Output (I/O) rate or Central Processing Unit (CPU) load) and memory usage);
- Software self monitoring of control flow and data flow;
- Some techniques and methods to verify outputs of different development phases;
- Project risk management;
- Use of Configuration Management tool;
- Tool selection criteria;
- Process improvement.

This page is intentionally left blank.

IEC 61508 COVERAGE

1. IEC 61508 STANDARD COVERAGE

This matrix intends to identify the applicability of IEC 61508 to ANS (Part I of this document, which recommends a set of ANS software lifecycle processes).

For that purpose, IEC 61508 Part I & III need to be referenced to address ANS software lifecycle scope.

1.1 DOCUMENTATION

Part III refers to Part I of IEC 61508 to address this topic.

61508 §	§ Purpose	Topic	Coverage	Rationale
I-5.1	Documentation Objectives	To specify the necessary information to be documented in order: - that all phases of the overall, E/E/PES and software safety lifecycles, - that the management of functional safety, verification and the functional safety assessment activities , can be effectively performed.	3-1	
I-5.2.1 to I-5.2.5	Documentation requirements	The documentation shall contain sufficient information, for each phase of the overall, E/E/PES and software safety lifecycles completed, necessary for effective performance of subsequent phases and verification activities, for functional safety management, for implementation of functional safety assessment.	3-1	
I-5.2.6 to I-5.2.11	Documentation requirements	All documents shall be revised, amended, approved according to la minimum list of criteria and be under control of an appropriate document control scheme.	3-1 3-4.2	

1.2 SOFTWARE QUALITY MANAGEMENT SYSTEM

1.2.1 MANAGEMENT OF FUNCTIONAL SAFETY

Part III refers to Part I of IEC 61508 to address this topic, but also lists some additional requirements.

61508 §	§ Purpose	Topic	Coverage	Rationale
I-6.1	Management of functional safety - Objectives	To specify the management and technical activities during the overall, E/E/PES and software safety lifecycle phases which are necessary for the achievement of the required functional safety of the E/E/PE safety-related systems. To specify the responsibilities of the persons, departments and organisations responsible for each overall, E/E/PES and software safety lifecycle phase or for activities within each phase.	1-1 1-3	
I-6.2.1.a to I-6.2.1.g	Management of functional safety - Requirements	Policy and strategy for achieving functional safety (evaluation, communication of safe working culture, responsibilities, measures and techniques, safety activities)	1-1 1-3.2	
I-6.2.1.h I-6.2.1.p	Management of functional safety	Staff training	4-4	
I-6.2.1. i	Management of functional safety	Hazard analysis and mitigation recommendations.	1-3.3	
I-6.2.1.j	Management of functional safety	Procedures for analysing operations and maintenance performance.	1-3.1 2-5	
I-6.2.1.k	Management of functional safety	Audits performance.	3-7	
I-6.2.1.l I-6.2.1.m	Management of functional safety	Modifications process.	3-8	
I-6.2.1.n	Management of functional safety	Potential hazards and safety-related systems information maintenance.	1-3.5	
I-6.2.1.o	Management of functional safety	Configuration management.	3-2	
I-6.2.2	Management of functional safety	Activities implementation and progress monitoring.	1-3.2 2-2	
I-6.2.3	Management of functional safety	Safety lifecycle activities output shall be reviewed by the organisations concerned and agreement reached.	3-6	
I-6.2.4	Management of functional safety	Responsibilities assignment.	1-3.2	
I-6.2.5	Management of functional safety	Quality assurance.	3-3	
6.2.2	Functional safety planning	Strategy for safety life cycle phases according to required safety integrity level.	1-3	

1.2.2 SOFTWARE CONFIGURATION MANAGEMENT

61508 §	§ Purpose	Topic	Coverage	Rationale
6.2.3.a	SW configuration management	Administrative and technical controls to manage SW changes.	3-2	
6.2.3.b	SW configuration management	Guarantee that all necessary operations have been carried out to demonstrate that the SW safety integrity has been achieved.	1-1	
6.2.3.c	SW configuration management	Configuration item identification.	3-2 4-2	
6.2.3.d	SW configuration management	Change-control procedure.	3-2	
6.2.3.e	SW configuration management	Configuration status, release status, justification for and approval of all modifications and details of modifications shall be documented to permit audits.	3-2 3-7	
6.2.3.f	SW configuration management	SW Release documentation. Archiving, retrieval and maintenance procedure.	3-2	

1.3 SOFTWARE SAFETY LIFECYCLE REQUIREMENTS

1.3.1 GENERAL REQUIREMENTS

61508 §	§ Purpose	Topic	Coverage	Rationale
7.1.1	General requirements	To structure the development of the SW into defined phases and activities.	2-3.1	
7.1.2.1	General requirements	A life cycle shall be selected.	2-3.1	
7.1.2.2	General requirements	Quality assurance procedure shall be integrated into life cycle activities.	3-3	
7.1.2.2	General requirements	Safety assurance procedure shall be integrated into life cycle activities.	1-3.4	
7.1.2.3 to 7.1.2.7	General requirements	Each phase of the SW life cycle shall be divided into elementary activities with the scope, inputs and outputs for each phase.	2-3.1	
7.1.2.8	General requirements	At any stage of the life cycle, if a change is required, pertaining to an earlier phase, then that earlier phase and the following shall be repeated.	2-3.1 1-3.3	

1.3.2 SOFTWARE SAFETY REQUIREMENTS SPECIFICATION

61508 §	§ Purpose	Topic	Coverage	Rationale
7.2.1	SW safety requirements specification	The objective is to specify the requirements in terms of functions and safety integrity.	2-3.1	

61508 §	§ Purpose	Topic	Coverage	Rationale
7.2.2	SW safety requirements specification	SW requirements shall be derived from system requirements and from safety planning. Requirements shall be expressed and structured according to criteria : safety, performance, interfaces, traceability, clarity, …	2-3.4	
7.2.2.5	SW safety requirements specification	Procedures shall be established for resolving any disagreement over the assignment of safety integrity level.	1-1	

1.3.3 SOFTWARE SAFETY VALIDATION PLANNING

61508 §	§ Purpose	Topic	Coverage	Rationale
7.3	SW safety validation planning	A validation plan shall include: planning, procedures, technical strategy, measures and techniques, pass/fail criteria, policies and procedures for evaluating results.	3-5	
7.3.2.5	SW safety validation planning	Pass/fail criteria.	3-5	

1.3.4 SOFTWARE DESIGN AND DEVELOPMENT

61508 §	§ Purpose	Topic	Coverage	Rationale
7.4.1	Objectives	Objectives are: - to create a software architecture that fulfils the specified requirements for software safety with respect to required SIL - to review and evaluate the requirements placed on the software by the hardware architecture - to select a suitable set of tools, including languages and compilers, for the required safety integrity level - is to design and implement software that fulfils the specified requirements for software safety with respect to required SIL - to verify that the requirements for software safety have been achieved.	2-3.5 3-6 2-3.1 4-2 2-3.5 2-3.6 2-3.7 3-4	
7.4.2	General requirements	List of requirements to define the SW design in accordance with the required safety integrity level.	2-3.5	
7.4.3	Requirements for SW architecture	List of requirements to define the SW architecture in accordance with the required safety integrity level. Techniques and measures include fault tolerance and fault avoidance.	2-3.5	
7.4.4	Requirements for support tools and programming languages	Support tools include languages, compilers, configuration management tools, automatic testing tools. Coding standards requirements are also listed.	4-2	
7.4.5	Requirements for detailed design and development	List of requirements to achieve SW detailed design in accordance with the required safety integrity level.	2-3.6	
7.4.6	Requirements for code implementation	List of requirements to achieve code implementation in accordance with the required safety integrity level.	2-3.7 3-6	
7.4.7	Requirements for	List of requirements to achieve SW module testing in accordance with the	2-3.7	

61508 §	§ Purpose	Topic	Coverage	Rationale
	SW module testing	required safety integrity level.	2-3.8	
7.4.8	Requirements for SW integration testing	List of requirements to achieve SW module testing in accordance with the required safety integrity level.	2-3.8	

1.3.5 PROGRAMMABLE ELECTRONICS INTEGRATION (HARDWARE AND SOFTWARE)

61508 §	§ Purpose	Topic	Coverage	Rationale
7.5.1	Objectives	To integrate SW onto the target HW, to combine them to ensure their compatibility and to meet the requirements of the intended safety integrity level.	2-3.10	
7.5.2	Requirements	List of requirements to achieve HW/SW integration in accordance with the required safety integrity level.	2-3.10	
7.5.2.6	Requirements	During the HW/SW integration testing, any modification or change to the integrated system shall be subject to an impact analysis, which shall determine all software modules impacted, and the necessary re-verification activities.	2-3.10	

1.3.6 SOFTWARE OPERATION AND MODIFICATION PROCEDURES

The requirements are given in IEC 61508 Part III §7.8 (Cf: Part I §3.8 of this document).

In this standard, software (unlike hardware) is not capable of being maintained: it is always modified.

1.3.7 SOFTWARE SAFETY VALIDATION

61508 §	§ Purpose	Topic	Coverage	Rationale
7.7	SW safety validation	To ensure that the integrated system complies with the specified requirements for SW safety at the intended safety integrity level. A SW safety validation document shall include results, validation activities, SW validation plan version, functions being validated, tools and equipment, discrepancies between expected and actual results.	2-3.9 3-5	
7.7.2.5	SW safety validation	When discrepancies occur between expected and actual results, the analysis made and the decisions taken on whether to continue the validation, or to issue a change request and return to an earlier part of the development lifecycle, shall be documented as part of the results of the software safety validation.	3-5	

61508 §	§ Purpose	Topic	Coverage	Rationale
7.7.2.6.a	SW safety validation	Animation and modelling may be used to supplement validation activities.	3-5	
7.7.2.7	SW tools qualification	All equipment used for validation shall be qualified.	5-1.1 5-1.2	Limitations identified.

1.3.8 SOFTWARE MODIFICATION

In this standard, software (unlike hardware) is not capable of being maintained: it is always modified. Consequently, references are made toward Chapter 3 where maintenance is considered.

61508 §	§ Purpose	Topic	Coverage	Rationale
7.8	SW modification	To make corrections, enhancements or adaptations to the **validated** SW, ensuring that the required safety integrity level is sustained.	2-5 3-8 1-1	
7.8.2.2	SW modification	The modification request analysis procedure shall detail the hazards which may be affected.	3-5 1-1	
7.8.2.3	SW modification	An analysis shall be carried out on the impact of the proposed software modification on the functional safety of the E/E/PE safety-related system; a) to determine whether or not a hazard and risk analysis is required; b) to determine which software safety lifecycle phases will need to be repeated.	3-5 1-1	

1.3.9 SOFTWARE VERIFICATION

61508 §	§ Purpose	Topic	Coverage	Rationale
7.9	SW verification	To test and evaluate the outputs from a given software safety lifecycle phase to ensure correctness and consistency with respect to the outputs and standards provided as input to that phase, , to the extent required by the safety integrity level	3-4	
7.9.2.13	Data verification	Data structures, application data, all modifiable parameters, plant interfaces, communications interfaces shall be verified.	3-4.2	

1.4 FUNCTIONAL SAFETY ASSESSMENT

61508 §	§ Purpose	Topic	Coverage	Rationale
8.1, I-8.1, I-8.2.1 to I-8.2.6	Functional safety assessment	To investigate and arrive at a judgement on the functional safety achieved by the E/E/PE safety-related systems.	1-3	
8.1, I-8.2.7 to I-8.2.11	Functional safety assessment	A functional safety assessment plan shall be documented and specify: those to undertake this activity, outputs from each assessment and its scope.	1-3.2	
8.2, I-8.2.12 to I-8.2.14	Functional safety assessment	The minimum level of independence of those carrying out the functional safety assessment shall be as specified.	1-3.4 (Partially)	The notion of independence in IEC 61508 includes three levels (people, unit, organisation)
8.3	Functional safety assessment	An assessment of functional safety may make use of the results of the techniques/measures.	N	The activities to achieve an objective are not part of the recommendations

1.5 HAZARD AND RISK ANALYSIS

61508 §	§ Purpose	Topic	Coverage	Rationale
I-7.4.1.1	Hazard and risk analysis Objectives	To determine the hazards and hazardous events of the EUC and the EUC control system (in all modes of operation), for all reasonably foreseeable circumstances including fault conditions and misuse (including all relevant human factors issues).	1-3.1	
I-7.4.1.2	Hazard and risk analysis Objectives	To determine the event sequences leading to the hazardous events determined.	1-3.1	
I-7.4.1.3	Hazard and risk analysis Objectives	To determine the EUC risks associated with the hazardous events determined.	1-3.1	
I-7.4.2.2	Hazard and risk analysis requirements	Consideration shall be given to the elimination of the hazards.	1-3.1 1-3.3	
I-7.4.2.5	Hazard and risk analysis requirements	The likelihood of the hazardous events shall be specified	1-3.1	
I-7.4.2.6	Hazard and risk analysis requirements	The potential consequences associated with hazardous events shall be determined.	1-3.1	
I-7.4.2.9	Hazard and risk analysis requirements	Lists of factors of how to apply techniques to perform hazard and risk analysis.	N	EATMP SAM provides only recommendations on limitations, advantages, drawbacks of techniques, not which should be used

61508 §	§ Purpose	Topic	Coverage	Rationale
I-7.4.2.11	Hazard and risk analysis requirements	Hazard and risk analysis shall be documented.	1-3.5	
I-7.4.2.12	Hazard and risk analysis requirements	The information and results which constitute the hazard and risk analysis shall be maintained for the EUC and the EUC control system throughout the overall safety lifecycle, from the hazard and risk analysis phase to the decommissioning or disposal phase.	1-3-5	

1.6 OVERALL SAFETY REQUIREMENTS

61508 §	§ Purpose	Topic	Coverage	Rationale
I-7.5.1 I-7.5.2	Overall safety requirements	To develop the specification for the overall safety requirements, in terms of the safety functions requirements and safety integrity requirements, for the E/E/PE safety-related systems, other technology safety-related systems and external risk reduction facilities, in order to achieve the required functional safety.	1-1 1-3.3	
I-7.5.2.1	Overall safety requirements	The safety functions necessary to ensure the required functional safety for each determined hazard shall be specified. This shall constitute the specification for the overall safety functions requirements.	1-3.1 1-3.3	
I-7.5.2.2	Overall safety requirements	The necessary risk reduction may be determined in a quantitative and/or qualitative manner.	1-3.1 1-3.3	
I-7.5.2.4	Overall safety requirements	The dangerous failure rate shall be taken into account to designate the EUC control system as a safety-related system.	1-3.1 1-3.3	
I-7.5.2.6	Overall safety requirements	The safety integrity requirements, in terms of the necessary risk reduction, shall be specified for each safety function. This shall constitute the specification for the overall safety integrity requirements.	1-3.3	
I-7.5.2.7	Overall safety requirements	The specification for the safety functions and the specification for the safety integrity requirements shall together constitute the specification for the overall safety requirements.	1-3.3	

1.7 SAFETY REQUIREMENTS ALLOCATION

61508 §	§ Purpose	Topic	Coverage	Rationale
I-7.6.1.1	Safety Requirements allocation objectives	To allocate the safety functions, contained in the specification for the overall safety requirements (both the safety functions requirements and the safety integrity requirements), to the designated E/E/PE safety-related systems, other technology safety-related systems and external risk reduction facilities.	1-3.3 2-1 2-3.2	
I-7.6.1.2	Safety Requirements allocation objectives	To allocate a safety integrity level to each safety function.	1-1 1-2 1-3.3	
I-7.6.2.2	Safety Requirements allocation	In allocating safety functions to the designated E/E/PE safety-related systems, other technology safety-related systems and external risk reduction facilities, the skills and resources available during all phases of the overall safety lifecycle shall be considered.	1-3.2 2.1	

61508 §	§ Purpose	Topic	Coverage	Rationale
I-7.6.2.3	Safety Requirements allocation	This allocation is iterative, and if it is found that the necessary risk reduction cannot be met, then the architecture shall be modified and the allocation repeated.	1-3.3 2.1	
I-7.6.2.4 To I-7.6.2.12	Safety Requirements allocation	The safety integrity requirements, for each safety function allocated to the E/E/PE safety-related system(s), shall be specified in terms of the safety integrity level and be qualified to indicate whether the target safety integrity parameter is either: - the average probability of failure to perform its design function on demand (for a low demand mode of operation); - the probability of a dangerous failure per hour (for a high demand or continuous mode of operation).	1-3.3 2.1	
I-7.6.2.13	Safety Requirements allocation	The information and results of the safety requirements allocation together with any assumptions and justifications made, shall be documented**.**	1-3.5 2.1	

1.8 OVERALL SAFETY VALIDATION PLANNING AND VALIDATION

This paragraph covers the system-related aspects of software validation.

61508 §	§ Purpose	Topic	Coverage	Rationale
I-7.8		A plan shall be developed to detail validation activities and specify validation strategy, techniques and measures, pass/fail criteria, policies and procedures for evaluating validation results.	2-3.11 3.5 1-3.4	
I-7.14		Validation activities shall be carried out in accordance with the validation plan. Information documented shall include validation activities, requirements specification version, functions being validated, results, tools and equipment, discrepancies between expected and actual results.	2.3.11 3.5 1-3.4	

1.9 OVERALL INSTALLATION PLANNING AND INSTALLATION

This paragraph covers the software-related aspects of installation and commissioning.

61508 §	§ Purpose	Topic	Coverage	Rationale
I-7.9	Overall installation planning	A plan for installation shall be developed to detail installation schedule, responsibilities, procedures and criteria for declaring installation complete.	2-3.12	
I-7.13	Overall installation	Installation shall be carried out in accordance with the installation plan. Information documented shall include installation activities and resolution of failures and incompatibilities.	2-3.12	

2. OMISSIONS OF IEC 61508

The purpose of this paragraph is to highlight what is not covered by IEC 61508 standard, versus what is identified in Part I of this document.

So the purpose is to identify the objectives that are not addressed IEC 61508 though recommended for an ANS Software Life Cycle.

Major IEC 61508 missing items are as follows:

- Configuration Management:
  - Tasks such as: baseline, traceability, configuration status accounting, software lifecycle environment, and software load control.
  - item to be configuration-managed (in IEC 61508 only at the software level, not at the level of software component) ,
  - control categories (No guidance is provided to perform different kind of control/configuration management (especially for data)),
  - use of a tool;
- Archive, retrieval and release requirements;
- Baseline management;
- Software Quality Assurance Process;
- Software Plans: identification, details, scheduling and configuration management;
- The use of user-modifiable, field-loadable software.
- Guidance for use of Commercial Off The Shelf software (COTS);
- Requirements for development planning (other than verification planning): resource, budget, staff, equipment and tools;
- Software standards (other than coding standards, such as requirements, design and integration);
- There is no consideration of tools' fitness for purpose (other than compilers/translators) in IEC 61508;
- Use of reverse engineering when using previously developed software;
- Tests cases, procedures and results verification;
- Exhaustive input testing guidance;
- The use of simulators or emulators for verification;
- Tools qualification (development and verification tools);
- Linking and loading actions;

- Requirements-based testing and structural coverage analysis (See Appendix A which s to be customised per domain);
- Lack of details for Transition criteria between processes/phases;
- Design description (processors/tasks allocation, HW resource management, scheduling, inter-task communication);
- No guidance on means to reduce software integrity level (architecture, isolation, partitioning, …);
- No guidance on the presence of deactivated code (only on unintended functions) or software patches;
- No constraints on presence of unreachable code (only on unintended functions);
- Project risk management;
- Process improvement;
- Safeguards against process hazards introduced through tools;
- Coverage metrics for testing purpose.

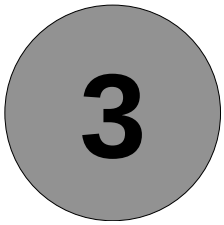
3. ISSUES WITH IEC 61508 SIL ALLOCATION PROCESS

IEC 61508 Part I §7.6.2.9 is often mis-used to allocate SIL the following way:

- a pure quantitative analysis (e.g. using Fault Tree Analysis) is performed that leads to allocate a quantitative pseudo “software failure rate”. Then this “software failure rate” is compared with the values claimed by Table 3 of §7.6.2.9 to allocate a SIL.

This process to allocate Software Assurance Level (and SIL) is totally unacceptable as not in accordance with “Recommendations for ANS SW”, not in accordance with IEC 61508 and assumes erroneously that a software failure rate can be assigned.

Therefore, refer to “Recommendations for ANS SW” as far as the SWAL allocation process is concerned, then equivalence between the allocated SWAL and its demonstration via a SIL can be achieved.



ISO/IEC 12207 COVERAGE

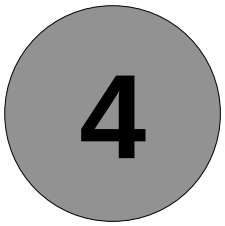
1. ISO/IEC 12207 STANDARD COVERAGE

As ISO/IEC 12207 has been used as the basis of the definition of the recommended Software Life Cycle for ANS, this standard is fully covered.

2. OMISSIONS OF ISO/IEC 12207

The major ISO/IEC 12207 missing items are as follows:

- Software Safety Assurance System: Management of functional safety (hazard and risk analysis, identification of safety/non-safety functions, functional safety assessment, safety validation, safety documentation);
- Test cases, procedures and results verification;
- Responsibilities definition;
- The use of user-modifiable, field-loadable software and Commercial Off The Shelf software (COTS);
- Transition criteria between processes/phases;
- Baseline management;
- Development standards.



ED 109/DO 278 COVERAGE

1. ED109/DO 278 STANDARD COVERAGE

This standard is a safety standard. It provides guidelines for the assurance of software in CNS/ATM systems used in ground or space-based applications. It is an adaptation of DO 178B to CNS/ATM systems.

ED109/DO 278 is not stand-alone; it is to be used with DO 178B and DO 248B.

This standard defines Assurance Levels that are closely linked with levels of DO 178B.

In ED109/DO 278, content of a Plan For Software Aspects of Approval is suggested, to be compared with the Plan for Software Aspects of Certification of ED12B/DO 178B.

It complements ED12B/DO178B by considering the use of COTS, and of adaptation data.

ED109/DO 278 §	Topic	Chap	Coverage	Rationale
P (Ref: 2 ; 4.1.3;5.1)	System aspects	2	3.1.1	System Overview
P (Ref: 2)	System aspects	2	3.2	System Requirements Analysis
P (Ref: 2)	System aspects	2	3.2	System Requirements Definition Criteria

ED109/DO 278 §	Topic	Chap	Coverage	Rationale
P(Ref: 2)	System aspects	2	2.1	Preliminary System Safety Assessment
·(Ref: 2.1)	Assurance levels	1	1	Assurance Rigour Objective
·(Ref: 2.1)	Assurance levels	1	1	Requirements Satisfaction Assurance
·(Ref: 2.1)	Assurance levels	1	3.4	Software Safety Assessment Verification
·(Ref: 2.1)	Assurance levels	3	3.4.1	Criticality Evaluation criteria
P(Ref: 2.1)	Assurance levels	2	3	System Architectural Design
·(Ref: 2.2)	Additional system considerations	1	3.1	System Description
P (Ref: 2.2)	Additional system considerations	2	3.3	System Architecture Definition
P(Ref: 2.2)	Additional system considerations	1	3.1	Operational Environment
P(Ref: 2.2)	Additional system considerations	1	3.1	System FHA & PSSA Output
P(Ref: 2.2)	Additional system considerations	2	3	System Requirements Analysis
·(Ref 3.1 Table A-1 line 4)	SW planning process	2	3.1	Non-Deliverable Items
·(Ref 3.1. Table A-1 line 3)	SW planning process	2	3.1	Environment Definition
·(Ref 3.1Table A-1 line 1)	SW planning process	2	3.1	Outputs Documentation
·(Ref 3.1Table A-1 line 1)	SW planning process	2	3.1	Development Plan
·(Ref 3.1Table A-1 line 3)	SW planning process	2	3.1	Lifecycle Definition
·(Ref 3.1Table A-1 line 5)	SW planning process	2	3.1	Development Standards
·(Ref: 3.1 Table A-1, Lines 1, 5, 7;4.1.4;4.1.9 line 3)	SW planning process	2	3	Software Development Plan
·(Ref: §3.1 Table A-1 lines 1, 2, 3, 4)	SW planning process	2	3.1.1	Software Integration Plan
·(Ref: 3.1 Table A-1 lines 1 to 7for COTS;4.1.9 Table A-10 lines 1, 2, 3)	SW planning process	2	3	Process Implementation
·(Ref: 3.1 Table A-1 lines 1, 2, 3, 4)	SW planning process	3	3.1	Process Implementation
·(Ref: 3.1 Table A-1 line 2)	SW planning process	3	3.4.1	Transition Criteria
·(Ref: 3.1 Table A-1 line 4)	SW planning process	3	3.4.1	Verification Environment Definition
·(Ref: 3.1 Table A-1 lines 1, 2, 3, 4)	SW planning process	3	3.4	Process implementation
·(Ref: 3.1 Table A-1 lines 1, 2, 3, 4)	SW planning process	3	3.4.1	Verification Plan
·(Ref: 3.1 Table A-1lines 6, 7;3.9 Table A-9 line 3)	SW planning process	3	3.3	Product assurance
·(Ref: 3.1 Tables A2.4 A2.5)	SW planning process	2	3.6	Software Detailed Design Standards
·(Ref: 3.1) Table A-1 line 1)	SW planning process	2	3	SW Integration
·(Ref: 3.1. Table A-1 line 3)	SW planning process	2	3.1.1	Software Development Environment
·(Ref: 3.1.2 ; 4.1.4.2 ;5.1)	SW planning process	2	3.1.1	Schedule
·(Ref: 3.1Table A-1 line 5;For COTS: 4.1.4.2)	SW planning process	2	3.1.1	Standards
·(Ref: 3.1Table A-1 lines 1, 2, 3)	SW planning process	3	3.2	Process Implementation
·(Ref: 3.1Table A-1 lines 1, 2, 3, 4)	SW planning process	3	3.3	Process implementation
·(Ref: 3.1Table A-1 lines 2, 3)	SW planning process	2	3.1.1	Software Lifecycle
P(Ref: 3.1 Table A-1 line 3 partial)	SW planning process	4	4.2	Process implementation
P(Ref: 3.1 Table A-1 lines 1, 2, 3, 4)	SW planning process	3	3.4	Verification
P(Ref: 3.1)	SW planning process	2	2.2	Planning
P(Ref: 3.1.1)	SW planning process	3	3.4.1	Verification Organisation Independence
·(Ref: 3.10 Table A-10 ; 5.1)	SW approval process	1	1	SW AL Assurance
·(Ref: 3.10 Table A-10 line 2 - 5.1)	SW approval process	1	3.1	Regulatory Framework
·(Ref: 3.10 Table A-10 lines 1, 2, 3)	SW approval process	3	3.4.2	Contract Verification
·(Ref: 5.1 - 3.10 Table A-10)	SW approval process	1	3.2	Software Safety Assessment Plan
·(Ref: 5.1 - 3.10 Table A-10)	SW approval process	1	3.2	Software Safety Assessment Plan Review
P(Ref: 5.1 - 3.10 Table A-10)	SW approval process	1	3.5	Software Safety Assessment Documentation Dissemination
·(Ref: 3.2 –Table A-2 (lines 1,2) Table	SW development	1	1	Requirements Correctness

ED109/DO 278 §	Topic	Chap	Coverage	Rationale
A-3 (lines 1, 2)	process			and Completeness
·(Ref: 3.2 Table A-2 line 1)	SW development process	2	3	SW Requirements Analysis
·(Ref: 3.2 Table A-2 line 3)	SW development process	2	3	SW Architectural Design
·(Ref: 3.2 Table A-2 line 3)	SW development process	2	3.5	Software Architectural Design Standards
·(Ref: 3.2 Table A-2 line 3)	SW development process	2	3.5	Software Architecture Definition Criteria
·(Ref: 3.2 Table A-2 line 6	SW development process	2	3.7	Coding Standards
·(Ref: 3.2 Table A-2 line 8)	SW development process	3	3.4.2	Adaptation data verification
·(Ref: 3.2 Table A-2 line3)	SW development process	2	3.5	Interfaces Design
·(Ref: 3.2 Table A-2 lines 1, 2)	SW development process	2	3	SW Detailed Design
·(Ref: 3.2 Tables A2.1, A2.2)	SW development process	2	3.4	Software Requirements Standards
·(Ref: 3.2, Table A.2line 3)	SW development process	3	3.1	Documentation (SW architectural design)
·(Ref: 3.2. Table A-2line 3)	SW development process	2	3.5	Top-Level Software Architecture Definition
·(Ref: 3.2Table A-2 lines 4, 5)	SW development process	2	3.6	Software Detailed Design Definition
·(Ref: 3.2Table A-2.line 3)	SW development process	2	3.5	Assurance Level Related Design
·(Ref: 3.2Tables A2.1, A2.2)	SW development process	2	3.4	Software Requirements Definition
·(Ref:3.2Table A-2 lines 4, 5)	SW development process	2	3.6	Software Detailed Design Definition Criteria
P(Ref: 3.2 Table A-2 lines 4, .5)	SW development process	2	3.6	Interfaces Design
P(Ref: 3.2 Table A-2 line 7)	SW development process	2	3.10	System Integration Definition
P(Ref: 3.2 Table A-2 line 7)	SW development process	2	3.10	Software Compatibility with target Hardware
·(Ref: 3.3 Table A-3 lines 1, 2; 3.4 Table A-4 lines 1, 2, 6)	Verification of outputs of SW requirements process	1	3.4	Software Safety Assessment Validation
·(Ref: 3.3 Table A-3 lines 1, 2, 3, 4, 5, 6, 7)	Verification of outputs of SW requirements process	3	3.4.2	Requirements Verification
·(Ref: 3.3 Table A-3, 3.4 Table A-4, 3.7 Table A-7)	Verification of outputs of SW requirements process	3	3.4.2	Process Verification
·(Ref: 3.3. Table A2.1, A2.2 , A-3 line 6)	Verification of outputs of SW requirements process	2	3.4	Software Requirements Definition Criteria
·(Ref: A3.6, A4.6, A5.6)	Testing	1	1	Requirements Traceability Assurance
·(Ref: 3.4 A2.1, A2.2)	Verification of outputs of SW design process	2	3.4	Assurance Level Related Requirements
·(Ref: 3.4 Table A-4 lines 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13 Over-compliant in line 13) (partitioning)	Verification of outputs of SW design process	3	3.4.2	Architectural Design Verification
·(Ref: 3.4 Table A-4 lines 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13 Over-compliant in line 13) (partitioning)	Verification of outputs of SW design process	3	3.4.2	Detailed design Verification
·(Ref: 3.4 Table A-4 lines 1, 2, 4, 5, 6;3.7 Table A-7 lines 2, 3, 4, 5, 6, 7, 8 partial)	Verification of outputs of SW design process	2	3.8	Software Integration Definition Criteria
·(Ref: 3.4, 3.5, 3.7)	Verification of outputs of SW design	3	3.4.1	Verification Process Implementation

ED109/DO 278 §	Topic	Chap	Coverage	Rationale
	process			
·(Ref: 3.5 Table A-5 lines 1, 2, 3, 4, 5, 6)	Verification of outputs of SW coding & integration processes	3	3.4.2	Source Code Verification
·(Ref: 3.5 Table A-5 lines 1, 2;3.6Table A-6 lines 1, 2, 3, 4)	Verification of outputs of SW coding & integration processes	2	3	SW Coding
P(Ref: 3.5 Table A-5 - 3.6 Table A-6)	Verification of outputs of SW coding & integration processes	2	3.7	Development & Documentation
P(Ref: 3.5 Table A-5; 3.6 Table A-6)	Verification of outputs of SW coding & integration processes	3	3.4.2	Development & Documentation
P(Ref: 3.5 Table A-5 line 7)	Verification of outputs of SW coding & integration processes	3	3.4.2	Integration Verification
·(Ref: 3.6 Table A-5 line 1)	Testing	1	1	Unintended Functions
·(Ref: 3.6 Table A-6 line 2)	Testing	2	3.8	Software Integration Standards
·(Ref: 3.6 Table A-6 lines 1, 2, 3, 4, 5 Over-compliant)	Testing	3	3.4.2	Executable Code Verification
·(Ref: 3.6 Table A-6 lines 3, 4, 5)	Testing	3	3.4.2	Software Units Testing
·(Ref: 3.6.3)	Testing	3	3.4.2	Software Units Test Definition
P(Ref: 3.6 Table A-6 lines 3, 4)	Testing	3	3.4.2	Module Testing Standards
·(Ref: 3.7 Table A-7 lines 1, 2, 3, 4, 5, 6, 7, 8 Over-compliant)	Verification of verification process	3	3.4.2	Verification Process Outputs Verification
P(Ref: 3.7 Table A-7 lines 2, 3)	Verification of verification process	3	3.4.2	System Qualification Testing Evaluation Criteria
P(Ref: 3.7 Table A-7)	Verification of verification process	2	3.7	Software Units Code definition Criteria
P(Ref: 3.7 Table A-7)	Verification of verification process	3	3.4.2	Software Units Tests definition Criteria
·(Ref 3.8Table A-8 line 1 to 6; For COTS: 4.1.7 Table 4-3 lines 1 to 4)	SW configuration management	2	3.1	Outputs Configuration Management
·(Ref 3.8Table A-8 line 3)	SW configuration management	2	3.1	Software Products Problems
·(Ref: 3.8 Table A-8 lines 3, 4)	SW configuration management	3	3.1	Maintenance
·(Ref: 3.8 Table A-8 line 3)	SW configuration management	3	3.8	Process implementation
·(Ref: 3.8 Table A-8 line 3)	SW configuration management	3	3.8	Problem resolution
·(Ref: 3.8 Table A-8 line 6	SW configuration management	4	4.2	Maintenance of the infrastructure
·(Ref: 3.8 Table A-8)	SW configuration management	1	1	Configuration ManagementAssurance
·(Ref: 3.8Table A-8 line 1)	SW configuration management	3	3.2	Configuration Identification
·(Ref: 3.8Table A-8 line 2)	SW configuration management	3	3.2	Baseline & Configuration Item Traceability
·(Ref: 3.8Table A-8 line 3)	SW configuration management	3	3.2	Configuration Control
·(Ref: 3.8Table A-8 line 3)	SW configuration management	3	3.2	Configuration Status Accounting
·(Ref: 3.8Table A-8 line 3)	SW configuration management	3	3.2	Configuration Evaluation
·(Ref: 3.8Table A-8 line 4)	SW configuration management	3	3.2	Release Management & Delivery
·(Ref: 3.8Table A-8 line 5)	SW configuration management	3	3.2	Software Load Control
·(Ref: 3.8Table A-8 line 6)	SW configuration management	3	3.2	Software Lifecycle Environment Control
P(Ref: 3.8 - 4.1.7)	SW configuration management	1	3.5	Software Safety Assessment Documentation Configuration Management

ED109/DO 278 §	Topic	Chap	Coverage	Rationale
P(Ref: 3.8 Table A-8 line 6 partial)	SW configuration management	4	4.2	Establishment of the infrastructure
·(Ref 3.9 Table A-9 line 1)	SW quality assurance	2	3.1	Support Process Compliance
·(Ref: 3.9 Table 9 Line 1)	SW quality assurance	3	3.4.1	Verification Results
·(Ref: 3.9 Table A-9 line 1)	SW quality assurance	3	3.3	Process assurance
·(Ref: 3.9 Table A-9)	SW quality assurance	1	3.4	Software Safety Assessment Process Assurance
·(Ref: 3.9, Table A-9)	SW quality assurance	2	2.2	Execution & control
P(Ref: 3.9 Table A-9 Line 1 partial)	SW quality assurance	3	3.6	Project management reviews
P(Ref: 3.9 Table A-9 Line 3 partial)	SW quality assurance	3	3.6	Process implementation
P(Ref: 3.9 Table A-9 Line 3 partial)	SW quality assurance	3	3.6	Technical reviews
·(Ref: 4)	Additional considerations	2	3.1.1	Additional considerations
·(Ref: 4.1.9 Table 4-1 line 1)	COTS	5	5.2.9	COTS planning
·(Ref: 4.1.9 Table 4-1 line 2)	COTS	5	5.2.9	COTS planning
·(Ref: 4.1.9 Table 4-1 line 3)	COTS	5	5.2.9	COTS planning
·(Ref: 4.1.9 Table 4-2 line 1)	COTS	5	5.2.9	COTS Acquisition
·(Ref: 4.1.9 Table 4-2 line 2)	COTS	5	5.2.9	COTS Acquisition
·(Ref: 4.1.9 Table 4-2 line 3)	COTS	5	5.2.9	COTS Acquisition
·(Ref: 4.1.9 Table 4-2 line 4)	COTS	5	5.2.9	COTS Acquisition
·(Ref: 4.1.9 Table 4-3 line 1)	COTS	5	5.2.9	COTS Configuration Management
·(Ref: 4.1.9 Table 4-3 line 2)	COTS	5	5.2.9	COTS Configuration Management
·(Ref: 4.1.9 Table 4-3 line 3)	COTS	5	5.2.9	COTS Configuration Management
·(Ref: 4.1.9 Table 4-3 line 4)	COTS	5	5.2.9	COTS Configuration Management
P (Ref: 4.1.4.2)	COTS	1	1	Software Modifications
P(Ref: 4.1.6.3)	COTS	1	1	SW AL Monitoring
P(Ref: For COTS 4.1.2)	COTS	3	3.1	Documentation (SW detailed design)
·(Ref: 5)	CNS/ ATM specific life cycle data	2	3.1.1	Software Lifecycle Data
P(Ref: 5)	CNS/ ATM specific life cycle data	1	3.5	Document Software Safety Assessment Process Results
·(Ref 5.1)	CNS/ ATM specific life cycle data	2	3.1.1	Software Overview
·(Ref: §5.1)	CNS/ ATM specific life cycle data	1	3.2	Software Safety Assessment Approach
P(Ref: 5.1)	CNS/ ATM specific life cycle data	1	3.2	Software Safety Assessment Plan Dissemination
P(Ref: 5.2)	CNS/ ATM specific life cycle data	5	5.1.1	Software Development Tool Qualification
P(Ref: 5.2)	CNS/ ATM specific life cycle data CNS/ ATM specific life cycle data	5	5.1.2	Software Development Tool Qualification

2. OMISSIONS

The purpose of this paragraph is to highlight what is not covered by ED 109/DO 278, versus what is identified in Part I of this document.

So the purpose is to identify the objectives that are not addressed by ED 109/DO 278 though recommended for an ANS Software Life Cycle.

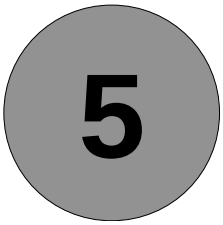
ED 109/DO 278 addresses mainly safety of software during development.

The major ED 12B/DO 178B missing items are as follows:

- There is no reference to system safety assessment standard;
- This standard does not provide guidance to allocate Assurance Levels;
- System aspects (architecture, integration, validation), HMI specifics are not covered;
- Documentation process is partially covered.

The major ED 12B/DO 178B missing items also apply to ED 109/DO 278:

- Only a part of the safety lifecycle is defined by ED12B/DO178B (the part concerned with the development of software). No requirements are set concerning acquisition, supply, installation, acceptance, maintenance, operation, decommissioning;
- Life cycle activities scheduling;
- Validation activities are not covered;
- Integration of the software product into the system on site;
- Software integration testing is not defined concurrently with the design/development phases;
- Requirements to choose a programming language;
- Staff training, staff competence;
- Capacity for safe modifications (A margin for throughput (e.g., Input and Output (I/O) rate or Central Processing Unit (CPU) load) and memory usage);
- Software self monitoring of control flow and data flow;
- Some techniques and methods to verify outputs of different development phases;
- Project risk management;
- Use of Configuration Management tool;
- Tool selection criteria;
- Process improvement.



CMMI[SM] V1.1 COVERAGE

1. CMMI[SM] STANDARD COVERAGE

This matrix intends to identify the relationship between the Capability Maturity Model Integrated (CMMI[SM]) and the ANS software Life ycle.

1.1 Summarized CMMI presentation

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 capabiliyty/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, Proiect Management and Supplier sourcing. The extension to other disciplines (e.g. hardware engineering, 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). 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.

1.2 SCOPE ANALYSIS COMPARED TO ANS SOFTWARE LIFE CYCLE

1. The CMMI is not designed for operation processes. It may be used for assessment of operation, as operation may be linked to a service concept, but it requires a specific interpretation.

   The ANS software life cycle is related to the full system life cycle, including operation aspects (typically operation risks, i.e. possible failures

analysis and flow-down to the software requirements, or operation feedback to the software assurance level).

2. The CMMI is intended to measure and to improve the process maturity of an organization, with different levels of organization: company, unit, department, unit, depending on the structure of the company. Except for large programmes where a part of the organization is dedicated to the programme, the CMMI is not designed to measure the process maturity of a project (at level 3, a large number of requirements are related to the organization, not (only) to the project).

   The ANS software life cycle is linked to a specific system to ensure that this system will operate safely. There is no concept of organizational maturity or organizational requirements.

3. The CMMI is designed for any type of development or services, and there are no specific safety “amplification” for safety-constrained development or services.

4. The CMMI highlight project organizational aspects (WBS, OBS, stakeholders involvement and commitment, etc.) and business objectives (organization performance), whereas ANS Software Life Cycle focus on expected result/system and intended operation in term of needed assurance level: the ANS Software Life Cycle process requirements are directly linked to the system, not on the organization needed to produce / maintain the system (except some specific independence requirements).

5. The CMMI doesn’t include requirement on specific criteria to be used for specify, design, develop or verify the products or services. It requires to specify such criteria depending on the business needs, i.e. contractual requirements, product line requirements, etc. (reminder: CMMI is general purpose model). The ANS Software Life Cycle specify such lists of criteria.

To which extend the CMMI copes with safety concerns?

1. The maturity level (staged representation) or the capability profile (continuous representation) of an organization provides a level of confidence on the ability to reach safety objectives: are activities “up to individuals” or specified & controlled or standardized or predictable …

2. A Safety & Security extension for integrated models has been developed to provide guidance on the interpretation of the models to deal with safety & security concerns. For example, the CMMI “Risk Management” PA is expected to covers the development/maintenance risks. If the scope is extended to operation, the “Risk Management” PA may be used as a reference to define an approach to identify and analyse impact of system operation failures, then identify mitigation means, etc.

3. If safety aspects (safety requirements, safety engineering, safety assurance) have been explicitly included in CMMI-based assessment, and a Company has been assessed at level 3, this means that the Company:

- has specified a standardized process to develop/maintain “safety-constrained” systems or software, that should be consistent with the external safety standards of the domain,
- is used to control adequate processes to deal with safety concerns ,
- has a recognized experience in developing/maintaining such systems or software.

CMMI			
	LEVEL 2		
Ref.	**Practice (Requirement)**	**ANS-Life Cycle** mapping	**Comments**
ReqM	**REQUIREMENT MANAGEMENT**		
The purpose of Requirements Management is to manage the requirements of the project's products and product components and to identify inconsistencies between those requirements and the project's plans and work products			
SGoal 1	**Requirements are managed and inconsistencies with project plans and work products are identified**		
SP 1.1	Develop an understanding with the requirements providers on the meaning of the requirements	2.2.2 2.3.1 3.3.4.2	*.1 Establish criteria for distinguishing appropriate requirements providers* *.2 Establish objective criteria for the acceptance of requirements* *.3 Analyze requirements to ensure that the established criteria are met* *.4 Reach an understanding of the requirements with the requirements provider so the project participants can commit to them*
SP 1.2	Obtain commitment to the requirements from the project participants	2.2.2	*.1 Assess the impact of requirements on existing commitments* *.2 Negotiate and record commitments*
SP 1.3	Manage changes to the requirements as they evolve during the project	3.3.4.2 3.3.8	*.1 Capture all requirements and requirements changes that are given to or generated by the project* *.2 Maintain the requirements change history with the rationale for the changes* *.3 Evaluate the impact of requirement changes from the standpoint of relevant stakeholders* *.4 Make the requirements and change data available to the project*
SP 1.4	Maintain bi-directional traceability among the requirements and the project plans and work products	1.1 2.3 2.3.2 2.3.3 2.3.4 2.3.5 2.3.6 2.3.7 2.3.8 3.3.4.2	*.1 Maintain requirements traceability to ensure that the source of lower-level (derived) requirements is documented* *.2 Maintain requirements traceability from a requirement to its derived requirements as well as to its allocation of functions, objects, people, processes, and work products* *.3 Maintain horizontal traceability from function to function and across interfaces* *.4 Generate the requirements traceability matrix*
SP 1.5	Identify inconsistencies between the project plans and work products and the requirements	3.3.4.2 4.4.1	*.1 Review the project's plans, activities, and work products for consistency with the requirements and the changes made to them* *.2 Identify the source of the inconsistency and the rationale* *.3 Identify changes that need to be made to the plans and work products resulting from changes to the requirements baseline* *.4 Initiate corrective actions*
PP	**PROJECT PLANNING**		
The purpose of Project Planning is to establish and maintain plans that define project activities			
SGoal 1	**Estimates of project planning parameters are established and maintained**		

CMMI			
	LEVEL 2		
Ref.	**Practice (Requirement)**	**ANS-Life Cycle** mapping	**Comments**
SP 1.1	Establish a top-level work breakdown structure (WBS) to estimate the scope of the project	2.3	*.1 Develop a WBS based on the product architecture* *.2 Identify the work packages in sufficient detail to specify estimates of project tasks, responsibilities, and schedule* *.3 Identify work products (or components of work products) that will be externally acquired* *.4 Identify work products that will be reused*
SP 1.2	Establish and maintain estimates of the attributes of the work products and tasks	None	*.1 Determine the technical approach for the project* *.2 Use appropriate methods to determine the attributes of the work products and tasks that will be used to estimate the resource requirements* *.3 Estimate the attributes of the work products and tasks* *.4 Estimate, as appropriate, the labor, machinery, materials, and methods that will be required by the project*
SP 1.3	Define the project life-cycle phases upon which to scope the planning effort	2.2.2 2.3 2.3.1 2.3.1.1 3.3.4.1	
SP 1.4	Estimate the project effort and cost for the work products and tasks based on estimation rationale	4.4.1	*.1 Collect the models or historical data that will be used to transform the attributes of the work products and tasks into estimates of the labor hours and cost* *.2 Include supporting infrastructure needs when estimating effort and cost* *.3 Estimate effort and cost using models and/or historical data*
SGoal 2	**A project plan is established and maintained as the basis for managing the project**		
SP 2.1	Establish and maintain the project's budget and schedule	2.3.1.1 3.3.6 4.4.1	*.1 Identify major milestones* *.2 Identify schedule assumptions* *.3 Identify constraints* *.4 Identify task dependencies* *.5 Define the budget and schedule* *.6 Establish corrective action criteria*
SP 2.2	Identify and analyze project risks	4.4.1	*.1 Identify risks* *.2 Document the risks* *.3 Review and obtain agreement with relevant stakeholders on the completeness and correctness of the documented risks* *.4 Revise the risks as appropriate*
SP 2.3	Plan for the management of project data	3.3.1	*.1 Establish requirements and procedures to ensure privacy and security of the data* *.2 Establish a mechanism to archive data and to access archived data* *.3 Determine the project data to be identified, collected, and distributed*
SP 2.4	Plan for necessary resources to perform the project	2.3.1 2.3.1.1 3.3.4.2 4.4.1 4.4.2	*.1 Determine process requirements* *.2 Determine staffing requirements* *.3 Determine facilities, equipment, and component requirements*
SP 2.5	Plan for knowledge and skills needed to perform the project	4.4.4	*.1 Identify the knowledge and skills needed to perform the project* *.2 Assess the knowledge and skills available* *.3 Select mechanisms for providing needed knowledge and skills* *.4 Incorporate selected mechanisms in the project plan*
SP 2.6	Plan the involvement of identified stakeholders	none	
SP 2.7	Establish and maintain the overall project plan content	2.3.1.1	
SGoal 3	**Commitments to the project plan are established and maintained**		
SP 3.1	Review all plans that affect the project to understand project commitments	3.3.4.2	

CMMI			
	LEVEL 2		
Ref.	**Practice (Requirement)**	**ANS-Life Cycle** mapping	**Comments**
SP 3.2	Reconcile the project plan to reflect available and estimated resources	4.4.1	
SP 3.3	Obtain commitment from relevant stakeholders responsible for performing and supporting plan execution	None	*.1 Identify needed support and negotiate commitments with relevant stakeholders* *.2 Document all organizational commitments, both full and provisional, ensuring appropriate level of signatories* *.3 Review internal commitments with senior management as appropriate* *.4 Review external commitments with senior management as appropriate*
PMC	**PROJECT MONITORING & CONTROL**		
The purpose of Project Monitoring and Control is to provide an understanding of the project's progress so that appropriate corrective actions can be taken when the project's performance deviates significantly from the plan			
SGoal 1	**Actual performance and progress of the project are monitored against the project plan**		
SP 1.1	Monitor the actual values of the project planning parameters against the project plan	2.3.1 33.3.7 4.4.1 4.4.2 4.4.4	*.1 Monitor progress against the schedule* *.2 Monitor the project's cost and expended effort* *.3 Monitor the attributes of the work products and tasks* *.4 Monitor resources provided and used* *.5 Monitor the knowledge and skills of project personnel* *.6 Document the significant deviations in the project planning parameters*
SP 1.2	Monitor commitments against those identified in the project plan	None	*.1 Regularly review commitments (both external and internal)* *.2 Identify commitments that have not been satisfied or which are at significant risk of not being satisfied* *.3 Document the results of the commitment reviews*
SP 1.3	Monitor risks (project management risks) against those identified in the project plan	None	*.1 Periodically review the documentation of the risks in the context of the project's current status and circumstances* *.2 Revise the documentation of the risks, as additional information becomes available, to incorporate changes* *.3 Communicate risk status to relevant stakeholders*
SP 1.4	Monitor the management of project data against the project plan	3.3.1 4.4.1	*.1 Periodically review data management activities against their description in the project plan* *.2 Identify and document significant issues and their impacts* *.3 Document the results of data management activity reviews*
SP 1.5	Monitor stakeholder involvement against the project plan	2.2.2	*.1 Periodically review the status of stakeholder involvement* *.2 Identify and document significant issues and their impacts* *.3 Document the results of the stakeholder involvement status reviews*
SP 1.6	Periodically review the project's progress, performance, and issues	2.3.1 3.3.6	*.1 Regularly communicate status on assigned activities and work products to relevant stakeholders* *.2 Review the results of collecting and analyzing measures for controlling the project* *.3 Identify and document significant issues and deviations from the plan* *.4 Document change requests and problems identified in any of the work products and processes* *.5 Document the results of the reviews* *.6 Track change requests and problem reports to closure*
SP 1.7	Review the accomplishments and results of the project at selected project milestones	None	*.1 Conduct reviews at meaningful points in the project's schedule, such as the completion of selected stages, with relevant stakeholders* *.2 Review the commitments, plan, status, and risks of the project* *.3 Identify and document significant issues and their impacts* *.4 Document the results of the review, action items, and decisions* *.5 Track action items to closure*

CMMI			
	LEVEL 2		
Ref.	**Practice (Requirement)**	**ANS-Life Cycle** mapping	**Comments**
SGoal 2	**Corrective actions are managed to closure when the project's performance or results deviate significantly from the plan**		
SP 2.1	Collect and analyze the issues and determine the corrective actions necessary to address the issues	None	*.1 Gather issues for analysis* *.2 Analyze issues to determine need for corrective action*
SP 2.2	Take corrective action on identified issues	None	*.1 Determine and document the appropriate actions needed to address the identified issues* *.2 Review and get agreement with relevant stakeholders on the actions to be taken* *.3 Negotiate changes to internal and external commitments*
SP 2.3	Manage corrective actions to closure	None	*.1 Monitor corrective actions for completion* *.2 Analyze results of corrective actions to determine the effectiveness of the corrective actions* *.3 Determine and document appropriate actions to correct deviations from planned results for corrective actions*
SAM	**SUPPLIER AGREEMENT MANAGEMENT**		
The purpose of Supplier Agreement Management is to manage the acquisition of products from suppliers for which there exists a formal agreement			
SGoal 1	**Agreements with the suppliers are established and maintained**		
SP 1.1	Determine the type of acquisition for each product or product component to be acquired	2.2.1	
SP 1.2	Select suppliers based on an evaluation of their ability to meet the specified requirements and established criteria	2.2.1 3.3.4.2	*.1 Establish and document criteria for evaluating potential suppliers* *.2 Identify potential suppliers and distribute solicitation material and requirements to them* *.3 Evaluate proposals according to evaluation criteria* *.4 Evaluate risks associated with each proposed supplier* *.5 Evaluate proposed suppliers' ability to perform the work* *.6 Select the supplier*
SP 1.3	Establish and maintain formal agreements with the supplier	None	*.1 Revise the requirements to be fulfilled by the supplier to reflect negotiations with the supplier when necessary* *.2 Document what the project will provide to the supplier* *.3 Document the supplier agreement* *.4 Ensure all parties to the agreement understand and agree to all requirements before implementing the agreement* *.5 Revise the supplier agreement as necessary* *.6 Revise the project's plans and commitments as necessary to reflect the supplier agreement*
SGoal 2	**Agreements with the suppliers are satisfied by both the project and the supplier**		
SP 2.1	Review candidate COTS products to ensure they satisfy the specified requirements that are covered under a supplier agreement	2.2.1 5.5.2.9	*.1 Develop criteria for evaluating COTS products* *.2 Evaluate candidate COTS products against the associated requirements and criteria* *.3 Evaluate the impact of candidate COTS products on the project's plans and commitments* *.4 Assess the suppliers' performance and ability to deliver* *.5 Identify risks associated with the selected COTS product and the supplier agreement* *.6 Select the COTS product to be acquired* *.7 Plan for the maintenance of the COTS product*
SP 2.2	Perform activities with the supplier as specified in the supplier agreement	2.2.1 2.2.2	*.1 Monitor supplier progress and performance (schedule, effort, cost, and technical performance) as defined in the supplier agreement* *.2 Monitor selected supplier processes and take corrective action when necessary* *.3 Conduct reviews with the supplier as specified in the supplier agreement* *.4 Conduct technical reviews with the supplier as defined in the supplier agreement* *.5 Conduct management reviews with the supplier as defined in the supplier agreement*

CMMI			
	LEVEL 2		
Ref.	**Practice (Requirement)**	**ANS-Life Cycle** mapping	**Comments**
			.6 Use the results of reviews to improve the supplier's performance and to establish and nurture long-term relationships with preferred suppliers *.7 Monitor risks involving the supplier and take corrective action as necessary* *.8 Revise the supplier agreement and project plans and schedules as necessary*
SP 2.3	Ensure that the supplier agreement is satisfied before accepting the acquired product	2.2.1 2.3 5.5.2.9	*.1 Define the acceptance procedures* *.2 Review and obtain agreement with relevant stakeholders on the acceptance procedures before the acceptance review or test* *.3 Verify that the acquired products satisfy their requirements* *.4 Confirm that the non-technical commitments associated with the acquired work product are satisfied* *.5 Document the results of the acceptance review or test* *.6 Establish and obtain supplier agreement on an action plan for any acquired work products that do not pass their acceptance review or test* *.7 Identify, document, and track action items to closure*
SP 2.4	Transition the acquired products from the supplier to the project	2.2.2	*.1 Ensure that there are appropriate facilities to receive, store, use, and maintain the acquired products* *.2 Ensure that appropriate training is provided for those involved in receiving, storing, using, and maintaining the acquired products* *.3 Ensure that storing, distributing, and using the acquired products are performed according to the terms and conditions specified in the supplier agreement or license*
M&A	**MEASUREMENT & ANALYSIS**		
The purpose of Measurement and Analysis is to develop and sustain a measurement capability that is used to support management information needs			
SGoal 1	**Measurement objectives and activities are aligned with identified information needs and objectives**		
SP 1.1	Establish and maintain measurement objectives that are derived from identified information needs and objectives	None	*.1 Document information needs and objectives* *.2 Prioritize information needs and objectives* *.3 Document, review, and update measurement objectives* *.4 Provide feedback for refining and clarifying information needs and objectives as necessary* *.5 Maintain traceability of the measurement objectives to the identified information needs and objectives*
SP 1.2	Specify measures to address the measurement objectives	None	*.1 Identify candidate measures based on documented measurement objectives* *.2 Identify existing measures that already address the measurement objectives* *.3 Specify operational definitions for the measures* *.4 Prioritize, review, and update measures*
SP 1.3	Specify how measurement data will be obtained and stored	None	*.1 Identify existing sources of data that are generated from current work products, processes, or transactions* *.2 Identify measures for which data are needed, but are not currently available* *.3 Specify how to collect and store the data for each required measure* *.4 Create data collection mechanisms and process guidance* *.5 Support automatic collection of the data where appropriate and feasible* *.6 Prioritize, review, and update data collection and storage procedures* *.7 Update measures and measurement objectives as necessary*
SP 1.4	Specify how measurement data will be analyzed and reported	None	*.1 Specify and prioritize the analyses that will be conducted and the reports that will be prepared* *.2 Select appropriate data analysis methods and tools* *.3 Specify administrative procedures for analyzing the data and communicating the results* *.4 Review and update the proposed content and format of the specified analyses and reports* *.5 Update measures and measurement objectives as necessary* *.6 Specify criteria for evaluating the utility of the analysis results, and of the conduct of the measurement and analysis activities*
SGoal 2	**Measurement results that address identified information needs and objectives are provided**		
SP 2.1	Obtain specified measurement data	None	*.1 Obtain the data for base measures* *.2 Generate the data for derived measures* *.3 Perform data integrity checks as close to the source of the data as*

CMMI			
	LEVEL 2		
Ref.	**Practice (Requirement)**	**ANS-Life Cycle** mapping	**Comments**
			possible
SP 2.2	Analyze and interpret measurement data	None	*.1 Conduct initial analyses, interpret the results, and draw preliminary conclusions .2 Conduct additional measurement and analysis as necessary, and prepare results for presentation .3 Review the initial results with relevant stakeholders .4 Refine criteria for future analyses*
SP 2.3	Manage and store measurement data, measurement specifications, and analysis results	None	*.1 Review the data to ensure their completeness, integrity, accuracy, and currency .2 Make the stored contents available for use only by appropriate groups and personnel .3 Prevent the stored information from being used inappropriately*
SP 2.4	Report results of measurement and analysis activities to all relevant stakeholders	None	*.1 Keep relevant stakeholders apprised of measurement results on a timely basis .2 Assist relevant stakeholders in understanding the results*
PPQA	**PROCESS & PRODUCT QUALITY ASSURANCE**		
The purpose of Process and Product Quality Assurance is to provide staff and management with objective insight into processes and associated work products			
SGoal 1	**Adherence of the performed process and associated work products and services to applicable process descriptions, standards, and procedures is objectively evaluated**		
SP 1.1	Objectively evaluate the designated performed processes against the applicable process descriptions, standards, and procedures	3.3.3 3.3.4.2	*.1 Promote an environment (created as part of project management) that encourages employee participation in identifying and reporting quality issues .2 Establish and maintain clearly stated criteria for the evaluations .3 Use the stated criteria to evaluate performed processes for adherence to process descriptions, standards, and procedures .4 Identify each noncompliance found during the evaluation .5 Identify lessons learned that could improve processes for future products and services*
SP 1.2	Objectively evaluate the designated work products and services against the applicable process descriptions, standards, and procedures	3.3.3 3.3.4.2	*.1 Select work products to be evaluated, based on documented sampling criteria if sampling is used .2 Establish and maintain clearly stated criteria for the evaluation of work products .3 Use the stated criteria during the evaluations of work products .4 Evaluate work products before they are delivered to the customer .5 Evaluate work products at selected milestones in their development .6 Perform in-progress or incremental evaluations of work products and services against process descriptions, standards, and procedures .7 Identify each case of noncompliance found during the evaluations .8 Identify lessons learned that could improve processes for future products and services*
SGoal 2			
SP 2.1	Communicate quality issues and ensure resolution of noncompliance issues with the staff and managers	3.3.3	*.1 Resolve each noncompliance with the appropriate members of the staff where possible .2 Document noncompliance issues when they cannot be resolved within the project .3 Escalate noncompliance issues that cannot be resolved within the project to the appropriate level of management designated to receive and act on noncompliance issues .4 Analyze the noncompliance issues to see if there are any quality trends that can be identified and addressed .5 Ensure that relevant stakeholders are aware of the results of evaluations and the quality trends in a timely manner .6 Periodically review open noncompliance issues and trends with the manager designated to receive and act on noncompliance issues .7 Track noncompliance issues to resolution*
SP 2.2	Establish and maintain records of the quality assurance activities	3.3.3	*.1 Record process and product quality assurance activities in sufficient detail such that status and results are known .2 Revise the status and history of the quality assurance activities as necessary*
CM	**CONFIGURATION MANAGEMENT**		
SGoal 1			

CMMI			
	LEVEL 2		
Ref.	**Practice (Requirement)**	**ANS-Life Cycle** mapping	**Comments**
SP 1.1	Identify the configuration items, components, and related work products that will be placed under configuration management	1.3.5	*.1 Select the configuration items and the work products that compose them based on documented criteria* *.2 Assign unique identifiers to configuration items* *.3 Specify the important characteristics of each configuration item* *.4 Specify when each configuration item is placed under configuration management* *.5 Identify the owner responsible for each configuration item*
SP 1.2	Establish and maintain a configuration management and change management system for controlling work products	3.3.2 5.5.2.9	*.1 Establish a mechanism to manage multiple control levels of configuration management* *.2 Store and retrieve configuration items in the configuration management system* *.3 Share and transfer configuration items between control levels within the configuration management system* *.4 Store and recover archived versions of configuration items* *.5 Store, update, and retrieve configuration management records* *.6 Create configuration management reports from the configuration management system* *.7 Preserve the contents of the configuration management system* *.8 Revise the configuration management structure as necessary*
SP 1.3	Create or release baselines for internal use and for delivery to the customer	2.3.11 2.3.9 2.5 3.3.2 3.3.6	*.1 Obtain authorization from the configuration control board (CCB) before creating or releasing baselines of configuration items* *.2 Create or release baselines only from configuration items in the configuration management system* *.3 Document the set of configuration items that are contained in a baseline* *.4 Make the current set of baselines readily available*
SGoal 2			
SP 2.1	Track change requests for the configuration items	3.3.4.1 3.3.7 5.5.2.9	*.1 Initiate and record change requests in the change request system* *.2 Analyze the impact of changes and fixes proposed in the change requests* *.3 Review change requests that will be addressed in the next baseline with those who will be affected by the changes and get their agreement* *.4 Track the status of change requests to closure*
SP 2.2	Control changes to the configuration items	5.5.2.9	*.1 Control changes to configuration items throughout the life of the product* *.2 Obtain appropriate authorization before changed configuration items are entered into the configuration management system* *.3 Check in and check out configuration items from the configuration management system for incorporation of changes in a manner that maintains the correctness and integrity of the configuration items* *.4 Perform reviews to ensure that changes have not caused unintended effects on the baselines (e.g., ensure that the changes have not compromised the safety and/or security of the system)* *.5 Record changes to configuration items and the reasons for the changes as appropriate*
SGoal 3	**Integrity of baselines is established and maintained**		
SP 3.1	Establish and maintain records describing configuration items	3.3.2	*.1 Record configuration management actions in sufficient detail so the content and status of each configuration item is known and previous versions can be recovered* *.2 Ensure that relevant stakeholders have access to and knowledge of the configuration status of the configuration items* *.3 Specify the latest version of the baselines* *.4 Identify the version of configuration items that constitute a particular baseline* *.5 Describe the differences between successive baselines*

CMMI			
	LEVEL 2		
Ref.	**Practice (Requirement)**	**ANS-Life Cycle** mapping	**Comments**
			.6 Revise the status and history (i.e., changes and other actions) of each configuration item as necessary
SP 3.2	Perform configuration audits to maintain integrity of the configuration baselines	3.3.2 3.3.7	*.1 Assess the integrity of the baselines* *.2 Confirm that the configuration records correctly identify the configuration of the configuration items* *.3 Review the structure and integrity of the items in the configuration management system* *.4 Confirm the completeness and correctness of the items in the configuration management system* *.5 Confirm compliance with applicable configuration management standards and procedures* *.6 Track action items from the audit to closure*

	LEVEL 3		
Ref.	**Practice (Requirement)**	**ANS-Life Cycle** mapping	**Comments**
RD	**REQUIREMENTS DEVELOPMENT**		
The purpose of Requirements Development is to produce and analyze customer, product, and product-component requirements			
SGoal 1	**Stakeholder needs, expectations, constraints, and interfaces are collected and translated into customer requirements**		
SP 1.1	Elicit stakeholder needs, expectations, constraints, and interfaces for all phases of the product life cycle	1.1 1.3.1 2.2.2 2.3.2	*.1 Engage relevant stakeholders using methods for eliciting needs, expectations, constraints, and external interfaces*
SP 1.2	Transform stakeholder needs, expectations, constraints, and interfaces into customer requirements	None	*.1 Translate the stakeholder needs, expectations, constraints, and interfaces into documented customer requirements* *.2 Define constraints for verification and validation*
SGoal 2			
SP 2.1	Establish and maintain product and product-component requirements, which are based on the customer requirements	1.3.3 2.3 2.3.4	*.1 Develop requirements in technical terms necessary for product and product component design* *.2 Derive requirements that result from design decisions* *.3 Establish and maintain relationships between requirements for consideration during change management and requirements allocation*
SP 2.2	Allocate the requirements for each product component	2.3 2.3.3 2.3.5 5.5.2.9	*.1 Allocate requirements to functions* *.2 Allocate requirements to product components* *.3 Allocate design constraints to product components* *.4 Document relationships among allocated requirements*
SP 2.3	Identify interface requirements	1.3.1	*.1 Identify interfaces both external to the product and internal to the product (i.e., between functional partitions or objects)* *.2 Develop the requirements for the identified interfaces*
SGoal 3			
SP 3.1	Establish and maintain operational concepts and associated scenarios	1.3.1 2.3.2	failure situations are interpreted as specific operational scenarios
SP 3.2	Establish and maintain a definition of required functionality	1.3.1	*.1 Develop operational concepts and scenarios that include functionality, performance, maintenance, support, and disposal as appropriate* *.2 Define the environment the product will operate in, including boundaries and constraints* *.3 Review operational concepts and scenarios to refine and discover requirements* *.4 Develop a detailed operational concept, as products and product components are selected, that defines the interaction of the product, the end user, and the environment, and that satisfies the operational, maintenance, support, and disposal needs*
SP 3.3	Analyze requirements to ensure that they are necessary and sufficient	1.3.4 2.3.4 3.3.4.2	*.1 Analyze and quantify functionality required by end users* *.2 Analyze requirements to identify logical or functional partitions (e.g., subfunctions)* *.3 Partition requirements into groups, based on established criteria (e.g., similar functionality, performance, or coupling), to facilitate and focus the requirements analysis* *.4 Consider the sequencing of time-critical functions both initially and subsequently during product component development* *.5 Allocate customer requirements to functional partitions, objects,*

	LEVEL 3		
Ref.	**Practice (Requirement)**	**ANS-Life Cycle** mapping	**Comments**
			people, or support elements to support the synthesis of solutions *.6 Allocate functional and performance requirements to functions and subfunctions*
SP 3.4	Analyze requirements to balance stakeholder needs and constraints	None	*.1 Analyze stakeholder needs, expectations, constraints, and external interfaces to remove conflicts and to organize into related subjects* *.2 Analyze requirements to determine whether they satisfy the objectives of higher-level requirements* *.3 Analyze requirements to ensure that they are complete, feasible, realizable, and verifiable* *.4 Identify key requirements that have a strong influence on cost, schedule, functionality, risk, or performance* *.5 Identify technical performance measures that will be tracked during the development effort* *.6 Analyze operational concepts and scenarios to refine the customer needs, constraints, and interfaces and to discover new requirements*
SP 3.5	Validate requirements to ensure the resulting product will perform as intended in the user's environment using multiple techniques as appropriate	None	*.1 Use proven models, simulations, and prototyping to analyze the balance of stakeholder needs and constraints* *.2 Perform a risk assessment on the requirements and functional architecture* *.3 Examine product life-cycle concepts for impacts of requirements on risks*

	LEVEL 3		
Ref.	**Practice (Requirement)**	**ANS-Life Cycle** mapping	**Comments**
TS	**TECHNICAL SOLUTION**		
The purpose of Technical Solution is to design, develop, and implement solutions to requirements. Solutions, designs, and implementations encompass products, product components, and product-related life-cycle processes either singly or in combinations as			
SGoal 1	**Product or product component solutions are selected from alternative solutions**		
SP 1.1	Develop detailed alternative solutions and selection criteria	2.3.5 3.3.4.2	*.1 Identify screening criteria to select a set of alternative solutions for consideration* *.2 Identify technologies currently in use and new product technologies for competitive advantage* *.3 Generate alternative solutions* *.4 Obtain a complete requirements allocation for each alternative* *.5 Develop the criteria for selecting the best alternative solution* *.6 Develop timeline scenarios for product operation and user interaction for each alternative solution*
SP 1.2	Evolve the operational concept, scenarios, and environments to describe the conditions, operating modes, and operating states specific to each product component	1.3.1	*.1 Evolve the operational concepts and scenarios to a degree of detail appropriate for the product component* *.2 Evolve the operational environments for the product components*
SP 1.3	Select the product component solutions that best satisfy the criteria established	None	*.1 Evaluate each alternative solution/set of solutions against the selection criteria established in the context of the operating concepts, operating modes, and operating states* *.2 Based on the evaluation of alternatives, assess the adequacy of the selection criteria and update these criteria as necessary* *.3 Identify and resolve issues with the alternative solutions and requirements* *.4 Select the best set of alternative solutions that satisfy the established selection criteria* *.5 Establish the requirements associated with the selected set of alternatives as the set of allocated requirements to those product components* *.6 Identify the product component solutions that will be reused or*

	LEVEL 3		
Ref.	**Practice (Requirement)**	**ANS-Life Cycle** mapping	**Comments**
			acquired *.7 Establish and maintain the documentation of the solutions, evaluations, and rationale*
SGoal 2			
SP 2.1	Develop a design for the product or product component	1.3.3 2.3 2.3.3 2.3.5 2.3.6 2.3.8 3.3.1 3.3.4.2	*.1 Establish and maintain criteria against which the design can be evaluated* *.2 Identify, develop, or acquire the design methods appropriate for the product* *.3 Ensure that the design adheres to applicable design standards and criteria* *.4 Ensure that the design adheres to allocated requirements* *.5 Document the design*
SP 2.2	Establish and maintain a technical data package	None	*.1 Determine the number of levels of design and the appropriate level of documentation for each design level* *.2 Base detailed design descriptions on the allocated product component requirements, architecture, and higher-level designs* *.3 Document the design in the technical data package* *.4 Document the rationale for key (i.e., significant effect on cost, schedule, or technical performance) decisions made or defined* *.5 Revise the technical data package as necessary*
SP 2.3	Design comprehensive product component interfaces in terms of established and maintained criteria	1.3.1 2.3.5 2.3.6	*.1 Define interface criteria* *.2 Apply the criteria to the interface design alternatives* *.3 Document the selected interface designs and the rationale for the selection*
SP 2.4	Evaluate whether the product components should be developed, purchased, or reused based on established criteria	2.2.1 5.5.2.9	*.1 Develop criteria for the reuse of product component designs* *.2 Analyze designs to determine if product components should be developed, reused, or purchased* *.3 When purchased or non-developmental (COTS, government off-the-shelf, and reuse) items are selected, plan for their maintenance*
SGoal 3			
SP 3.1	Implement the designs of the product components	2.3 2.3.7 3.3.1 3.3.4.2 3.3.7 5.5.1.1	*.1 Use effective methods to implement the product components* *.2 Adhere to applicable standards and criteria* *.3 Conduct peer reviews of the selected product components* *.4 Perform unit testing of the product component as appropriate* *.5 Revise the product component as necessary*
SP 3.2	Develop and maintain the end-use documentation	3.3.7	*.1 Review the requirements, design, product, and test results to ensure that issues affecting the installation, operation, and maintenance documentation are identified and resolved* *.2 Use effective methods to develop the installation, operation, and maintenance documentation* *.3 Adhere to the applicable documentation standards* *.4 Develop preliminary versions of the installation, operation, and maintenance documentation in early phases of the project life cycle for review by the relevant stakeholders* *.5 Conduct peer reviews of the installation, operation, and maintenance documentation* *.6 Revise the installation, operation, and maintenance documentation as necessary*
PI	**PRODUCT INTEGRATION**		
The purpose of Product Integration is to assemble the product from the product components, ensure that the product, as integrated, functions properly, and deliver the product			
SGoal 1	**Preparation for product integration is conducted**		

	LEVEL 3		
Ref.	**Practice (Requirement)**	**ANS-Life Cycle** mapping	**Comments**
SP 1.1	Determine the product component integration sequence	2.3 2.3.8	*.1 Identify the product components to be integrated* *.2 Identify the product integration verifications to be performed using the definition of the interfaces between the product components* *.3 Identify alternative product component integration sequences* *.4 Select the best integration sequence* *.5 Periodically review the product integration sequence and revise as needed* *.6 Record the rationale for decisions taken and deferred*
SP 1.2	Establish and maintain the environment needed to support the integration of the product components	2.3.10 2.3.8	*.1 Identify verification criteria and procedures for the product integration environment* *.2 Decide whether to make or buy the needed product integration environment* *.3 Develop an integration environment if a suitable environment cannot be acquired* *.4 Maintain the product integration environment throughout the project* *.5 Dispose of those portions of the environment that are no longer useful*
SP 1.3	Establish and maintain procedures and criteria for integration of the product components	2.3 2.3.10	*.1 Establish and maintain product integration procedures for the product components* *.2 Establish and maintain criteria for product component integration and evaluation* *.3 Establish and maintain criteria for validation and delivery of the integrated product*
SGoal 2	**The product component interfaces, both internal and external, are compatible**		
SP 2.1	Review interface descriptions for coverage and completeness	2.3.5 2.3.6	*.1 Review interface data for completeness and ensure complete coverage of all interfaces* *.2 Ensure that product components and interfaces are marked to ensure easy and correct connection to the joining product component* *.3 Periodically review the adequacy of interface descriptions*
SP 2.2	Manage internal and external interface definitions, designs, and changes for products and product components	3.3.4.2	*.1 Ensure the compatibility of the interfaces throughout the life of the product* *.2 Resolve conflict, noncompliance, and change issues* *.3 Maintain a repository for interface data accessible to project participants*
SGoal 3	**Verified product components are assembled and the integrated, verified, and validated product is delivered**		
SP 3.1	Confirm, prior to assembly, that each product component required to assemble the product has been properly identified, functions according to its description, and that the product component interfaces comply with the interface descriptions	3.3.4.2	*.1 Track the status of all product components as soon as they become available for integration* *.2 Ensure that product components are delivered to the product integration environment in accordance with the product integration sequence and available procedures* *.3 Confirm the receipt of each properly identified product component* *.4 Ensure that each received product component meets its description* *.5 Check the configuration status against the expected configuration* *.6 Perform pre-check (for example, by a visual inspection and using basic measures) of all the physical interfaces before connecting product components together*
SP 3.2	Assemble product components according to the product integration sequence and available procedures	2.3.8 3.3.4.2	*.1 Ensure the readiness of the product integration environment* *.2 Ensure that the assembly sequence is properly performed* *.3 Revise the product integration sequence and available procedures as appropriate*
SP 3.3	Evaluate assembled product components for interface compatibility	None	*.1 Conduct the evaluation of assembled product components following the product integration sequence and available procedures* *.2 Record the evaluation results*
SP 3.4	Package the assembled product or product component and deliver it to the appropriate customer	2.2.2 2.3.11 2.3.12 2.3.9 2.5 3.3.4.2	*.1 Review the requirements, design, product, verification results, and documentation to ensure that issues affecting the packaging and delivery of the product are identified and resolved* *.2 Use effective methods to package and deliver the assembled product* *.3 Satisfy the applicable requirements and standards for packaging and delivering the product* *.4 Prepare the operational site for installation of the product* *.5 Deliver the product and related documentation and confirm receipt* *.6 Install the product at the operational site and confirm correct operation*
Ver	**VERIFICATION**		
The purpose of Verification is to ensure that selected work products meet their specified requirements			
SGoal 1	**Preparation for verification is conducted**		

	LEVEL 3		
Ref.	**Practice (Requirement)**	**ANS-Life Cycle** mapping	**Comments**
SP 1.1	Select the work products to be verified and the verification methods that will be used for each	1.3.3 3.3.7	*.1 Identify work products for verification* *.2 Identify the requirements to be satisfied by each selected work product* *.3 Identify the verification methods that are available for use* *.4 Define the verification methods to be used for each selected work product* *.5 Submit for integration with the project plan the identification of work products to be verified, the requirements to be satisfied, and the methods to be used*
SP 1.2	Establish and maintain the environment needed to support verification	3.3.4.1 3.3.4.2 5.5.1.1 5.5.1.2	*1. Identify verification environment requirements* *2. Identify verification resources that are available for reuse and modification* *3. Identify verification equipment and tools* *.4 Acquire verification support equipment and an environment, such as test equipment and software*
SP 1.3	Establish and maintain verification procedures and criteria for the selected work products	2.3.1.1 3.3.4.1 3.3.4.2 3.3.7	*.1 Generate the set of comprehensive, integrated verification procedures for work products and any commercial off-the-shelf products, as necessary* *.2 Develop and refine the verification criteria when necessary* *.3 Identify the expected results, any tolerances allowed in observation, and other criteria for satisfying the requirements* *.4 Identify any equipment and environmental components needed to support verification*
SGoal 2	**Peer reviews are performed on selected work products**		
SP 2.1	Prepare for peer reviews of selected work products	1.3.4	*.1 Determine what type of peer review will be conducted* *.2 Define requirements for collecting data during the peer review* *.3 Establish and maintain entry and exit criteria for the peer review* *.4 Establish and maintain criteria for requiring another peer review* *.5 Establish and maintain checklists to ensure that the work products are reviewed consistently* *.6 Develop a detailed peer review schedule, including the dates for peer review training and for when materials for peer reviews will be available* *.7 Ensure that the work product satisfies the peer review entry criteria prior to distribution* *.8 Distribute the work product to be reviewed and its related information to the participants early enough to enable participants to adequately prepare for the peer review* *.9 Assign roles for the peer review as appropriate* *.10 Prepare for the peer review by reviewing the work product prior to conducting the peer review*
SP 2.2	Conduct peer reviews on selected work products and identify issues resulting from the peer review	None	*.1 Perform the assigned roles in the peer review* *.2 Identify and document defects and other issues in the work* *.3 Record the results of the peer review, including the action items* *.4 Collect peer review data* *.5 Identify action items and communicate the issues to relevant stakeholders* *.6 Conduct an additional peer review if the defined criteria indicate the need* *.7 Ensure that the exit criteria for the peer review are satisfied*

	LEVEL 3		
Ref.	**Practice (Requirement)**	**ANS-Life Cycle** mapping	**Comments**
SP 2.3	Analyze data about preparation, conduct, and results of the peer reviews	None	*.1 Record data related to the preparation, conduct, and results of the peer reviews* *.2 Store the data for future reference and analysis* *.3 Protect the data to ensure that peer review data are not used inappropriately* *.4 Analyze the peer review data*
SGoal 3	**Selected work products are verified against their specified requirements**		
SP 3.1	Perform verification on the selected work products	None	*.1 Perform verification of selected work products against their requirements* *.2 Record the results of verification activities* *.3 Identify action items resulting from verification of work products* *.4 Document the "as-run" verification method and the deviations from the available methods and procedures discovered during its performance*
SP 3.2	Analyze the results of all verification activities and identify corrective action	None	*.1 Compare actual results to expected results* *.2 Based on the established verification criteria, identify products that have not met their requirements or identify problems with the methods, procedures, criteria, and verification environment* *.3 Analyze the verification data on defects* *.4 Record all results of the analysis in a report* *.5 Use verification results to compare actual measurements and performance to technical performance parameters* *.6 Provide information on how defects may be resolved (including verification methods, criteria, and verification environment) and formalize it in a plan*
Val	**VALIDATION**		
The purpose of Validation is to demonstrate that a product or product component fulfills its intended use when placed in its intended environment			
SGoal 1	**Preparation for validation is conducted**		
SP 1.1	Select products and product components to be validated and the validation methods that will be used for each	3.3.5	*.1 Identify the key principles, features, and phases for product or product component validation throughout the life of the project* *.2 Determine which categories of user needs (operational, maintenance, training, or support) are to be validated* *.3 Select the product and product components to be validated* *.4 Select the evaluation methods for product or product component validation* *.5 Review the validation selection, constraints, and methods with relevant stakeholders*
SP 1.2	Establish and maintain the environment needed to support validation	3.3.5	*.1 Identify validation environment requirements* *.2 Identify customer-supplied products* *.3 Identify reuse items* *.4 Identify test equipment and tools* *.5 Identify validation resources that are available for reuse and modification* *.6 Plan the availability of resources in detail*
SP 1.3	Establish and maintain procedures and criteria for validation	3.3.4.2	*.1 Review the product requirements to ensure that issues affecting validation of the product or product component are identified and resolved* *.2 Document the environment, operational scenario, procedures, inputs, outputs, and criteria for the validation of the selected product or product component* *.3 Assess the design as it matures in the context of the validation environment to identify validation issues*
SGoal 2	**The product or product components are validated to ensure that they are suitable for use in their intended operating environment**		
SP 2.1	Perform validation on the selected products and product components	None	
SP 2.2	Analyze the results of the validation activities and identify issues	None	*.1 Compare actual results to expected results* *.2 Analyze the validation data for defects* *.3 Based on the established validation criteria, identify products and product components that do not perform suitably in their intended operating environments, or identify problems with the methods, criteria, and/or environment* *.4 Record the results of the analysis and identify issues* *.5 Use validation results to compare actual measurements and performance to intended use or operational need*

	LEVEL 3		
Ref.	**Practice (Requirement)**	**ANS-Life Cycle** mapping	**Comments**
OPF	**ORGANIZATIONAL PROCESS FOCUS**		
The purpose of Organizational Process Focus is to plan and implement organizational process improvement based on a thorough understanding of the current strengths and weaknesses of the organization's processes and process assets.			
SGoal 1	**Strengths, weaknesses, and improvement opportunities for the organization's processes are identified periodically and as needed**		
SP 1.1	Establish and maintain the description of the process needs and objectives for the organization	None	*. Identify the policies, standards, and business objectives that are applicable to the organization's processes . Examine relevant process standards and models for best practices . Determine the organization's process performance objectives . Define the essential characteristics of the organization's processes . Document the organization's process needs and objectives . Revise the organization's process needs and objectives as needed*
SP 1.2	Appraise the processes of the organization periodically and as needed to maintain an understanding of their strengths and weaknesses	4.4.3	*. Obtain sponsorship of the process appraisal from senior management . Define the scope of the process appraisal . Determine the method and criteria for process appraisal . Plan, schedule, and prepare for the process appraisal . Conduct the process appraisal . Document and deliver the appraisal's activities and findings*
SP 1.3	Identify improvements to the organization's processes and process assets	4.4.3	*. Determine candidate process improvements . Prioritize the candidate process improvements . Identify and document the process improvements that will be implemented . Revise the list of planned process improvements to keep it current*
SGoal 2	**Strengths, weaknesses, and improvement opportunities for the organization's processes are identified periodically and as needed**		
SP 2.1	Establish and maintain process action plans to address improvements to the organization's processes and process assets	4.4.3	*. Identify strategies, approaches, and actions to address the identified process improvements . Establish process action teams to implement the actions . Document process action plans . Review and negotiate process action plans with relevant stakeholders . Review process action plans as necessary*
SP 2.2	Implement process action plans across the organization	4.4.3	*. Make process action plans readily available to relevant stakeholders . Negotiate and document commitments among the process action teams and revise their process action plans as necessary . Track progress and commitments against process action plans . Conduct joint reviews with the process action teams and relevant stakeholders to monitor the progress and results of the process actions . Plan pilots needed to test selected process improvements . Review the activities and work products of process action teams . Identify, document, and track to closure issues in implementing process action plans . Ensure that the results of implementing process action plans satisfy the organization's process improvement objectives*
SP 2.3	Deploy organizational process assets across the organization	None	*. Deploy organizational process assets and associated methods and tools . Deploy the changes that were made to the organizational process assets . Document the changes to the organizational process assets . Provide guidance and consultation on the use of the organizational process assets*
SP 2.4	Incorporate process-related work products, measures, and improvement information derived from planning and performing the process into the organizational process assets	None	*. Conduct periodic reviews of the effectiveness and suitability of the organization's set of standard processes and related organizational process assets relative to the organization's business objectives . Obtain feedback about the use of the organizational process assets . Derive lessons learned from defining, piloting, implementing, and deploying the organizational process assets . Make lessons learned available to the people in the organization as appropriate . Analyze the organization's common set of measures . Appraise the processes, methods, and tools in use in the organization and develop recommendations for improving the organizational process assets . Make the best use of the organization's processes, methods, and tools available to the people in the organization as appropriate . Manage process improvement proposals*

	LEVEL 3		
Ref.	**Practice (Requirement)**	**ANS-Life Cycle** mapping	**Comments**
			. Establish and maintain records of the organization's process improvement activities
OPD	**ORGANIZATIONAL PROCESS DEFINITION**		
The purpose of Organizational Process Definition is to establish and maintain a usable set of organizational process assets			
SGoal 1	**A set of organizational process assets is established and maintained**		
SP 1.1	Establish and maintain the organization's set of standard processes	None	*. Decompose each standard process into constituent process elements to the detail needed to understand and describe the process . Specify the critical attributes of each process element . Specify the relationships of the process elements . Ensure that the organization's set of standard processes adheres to applicable policies; process standards and models; and product standards . Ensure that the organization's set of standard processes satisfies the process needs and objectives of the organization . Ensure that there is appropriate integration among the processes that are included in the organization's set of standard processes . Document the organization's set of standard processes . Conduct peer reviews on the organization's set of standard processes . Revise the organization's set of standard processes as necessary*
SP 1.2	Establish and maintain descriptions of the life-cycle models approved for use in the organization	None	*. Select life-cycle models based on the needs of projects and the organization . Document the descriptions of the life-cycle models . Conduct peer reviews on the life-cycle models . Revise the descriptions of the life-cycle models as necessary*
SP 1.3	Establish and maintain the tailoring criteria and guidelines for the organization's set of standard processes	4.4.3	*. Specify the selection criteria and procedures for tailoring the organization's set of standard processes . Specify the standards for documenting the defined processes . Specify the procedures for submitting and obtaining approval of waivers from the requirements of the organization's set of standard processes . Document the tailoring guidelines for the organization's set of standard processes . Conduct peer reviews on the tailoring guidelines . Revise the tailoring guidelines as necessary*
SP 1.4	Establish and maintain the organization's measurement repository	None	*. Determine the organization's needs for storing, retrieving, and analyzing measurements . Define a common set of process and product measures for the organization's set of standard processes . Design and implement the measurement repository . Specify the procedures for storing, updating, and retrieving measures . Conduct peer reviews on the definitions of the common set of measures and the procedures for storing and retrieving measures . Enter the specified measures into the repository . Make the contents of the measurement repository available for use by the organization and projects as appropriate . Revise the measurement repository, common set of measures, and procedures as the organization's needs change*
SP 1.5	Establish and maintain the organization's process asset library	None	*. Design and implement the organization's process asset library, including the library structure and support environment . Specify the criteria for including items in the library . Specify the procedures for storing and retrieving items . Enter the selected items into the library and catalog them for easy reference and retrieval . Make the items available for use by the projects . Periodically review the use of each item and use the results to maintain the library contents . Revise the organization's process asset library as necessary*
OT	**ORGANIZATIONAL TRAINING**		
The purpose of Organizational Training is to develop the skills and knowledge of people so they can perform their roles effectively and efficiently			

	LEVEL 3		
Ref.	**Practice (Requirement)**	**ANS-Life Cycle** mapping	**Comments**
SGoal 1	**A training capability that supports the organization's management and technical roles is established and maintained**		
SP 1.1	Establish and maintain the strategic training needs of the organization	4.4.4	*. Analyze the organization's strategic business objectives and process improvement plan to identify potential future training needs* *. Document the strategic training needs of the organization* *. Determine the roles and skills needed to perform the organization's set of standard processes* *. Document the training needed to perform the roles in the organization's set of standard processes* *. Revise the organization's strategic needs and required training as necessary*
SP 1.2	Determine which training needs are the responsibility of the organization and which will be left to the individual project or support group	None	*. Analyze the training needs identified by the various projects and support groups* *. Negotiate with the various projects and support groups on how their specific training needs will be satisfied* *. Document the commitments for providing training support to the projects and support groups*
SP 1.3	Establish and maintain an organizational training tactical plan	4.4.4	*. Establish plan content* *. Establish commitments to the plan* *. Revise plan and commitments as necessary*
SP 1.4	Establish and maintain training capability to address organizational training needs	4.4.4	*. Select the appropriate approaches to satisfy specific organizational training needs* *. Determine whether to develop training materials internally or acquire them externally* *. Develop or obtain training materials* *. Develop or obtain qualified instructors* *. Describe the training in the organization's training curriculum*
SGoal 2	**Training necessary for individuals to perform their roles effectively is provided**		
SP 2.1	Deliver the training following the organizational training tactical plan	4.4.4	*. Select the people who will receive the training* *. Schedule the training, including any resources, as necessary (e.g., facilities and instructors)* *. Conduct the training* *. Track the delivery of training against the plan*
SP 2.2	Establish and maintain records of the organizational training	4.4.4	*. Keep records of all students who successfully complete each training course or other approved training activity as well as those who are unsuccessful* *. Keep records of all staff who have been waived from specific training* *. Keep records of all students who successfully complete their designated required training* *. Make training records available to the appropriate people for consideration in assignments*
SP 2.3	Assess the effectiveness of the organization's training program	None	*. Assess in-progress or completed projects to determine whether staff knowledge is adequate for performing project tasks* *. Provide a mechanism for assessing the effectiveness of each training course with respect to established organizational, project, or individual learning (or performance) objectives* *. Obtain student evaluations of how well training activities met their needs*
IPM	**INTEGRATED PROJECT MANAGEMENT**		
The purpose of Integrated Project Management is to establish and manage the project and the involvement of the relevant stakeholders according to an integrated and defined process that is tailored from the organization's set of standard processes			
SGoal 1	**The project is conducted using a defined process that is tailored from the organization's set of standard processes**		
SP 1.1	Establish and maintain the project's defined process	2.3.1 3.3.4.2	*.1 Select a life-cycle model from those available from the organizational process assets* *.2 Select the standard processes from the organization's set of standard processes that best fit the needs of the project* *.3 Tailor the organization's set of standard processes and other organizational process assets according to the tailoring guidelines to produce the project's defined process* *.4 Use other artifacts from the organization's process asset library as appropriate* *.5 Document the project's defined process*

	LEVEL 3		
Ref.	**Practice (Requirement)**	**ANS-Life Cycle** mapping	**Comments**
			.6 Conduct peer reviews of the project's defined process *.7 Revise the project's defined process as necessary*
SP 1.2	Use the organizational process assets and measurement repository for estimating and planning the project's activities	None	*.1 Base the activities for estimating and planning on the tasks and work products of the project's defined process* *.2 Use the organization's measurement repository in estimating the project's planning parameters*
SP 1.3	Integrate the project plan and the other plans that affect the project to describe the project's defined process	2.3.1.1 3.3.4.1 3.3.4.2 5.5.2.9	*.1 Integrate other plans that affect the project with the project plan* *.2 Incorporate into the project plan the definitions of measures and measurement activities for managing the project* *.3 Identify and analyze product and project interface risks* *.4 Schedule the tasks in a sequence that accounts for critical development factors and project risks* *.5 Incorporate the plans for performing peer reviews on the work products of the project's defined process* *.6 Incorporate the training needed to perform the project's defined process in the project's training plans* *.7 Establish objective entry and exit criteria to authorize the initiation and completion of the tasks described in the work breakdown structure (WBS)* *.8 Ensure that the project plan is appropriately compatible with the plans of relevant stakeholders* *.9 Identify how conflicts will be resolved that arise among relevant stakeholders*
SP 1.4	Manage the project using the project plan, the other plans that affect the project, and the project's defined process	2.3.1 3.3.4.1	*.1 Implement the project's defined process using the organization's process asset library* *.2 Monitor and control the project's activities and work products using the project's defined process, project plan, and other plans that affect the project* *.3 Obtain and analyze the selected measures to manage the project and support the organization's needs* *.4 Periodically review the adequacy of the environment to meet the project's needs and to support coordination* *.5 Periodically review and align the project's performance with the current and anticipated needs, objectives, and requirements of the organization, customer, and end users, as appropriate*
SP 1.5	Contribute work products, measures, and documented experiences to the organizational process assets	None	*.1 Propose improvements to the organizational process assets* *.2 Store process and product measures in the organization's measurement repository* *.3 Submit documentation for possible inclusion in the organization's process asset library* *.4 Document lessons learned from the project for inclusion in the organization's process asset library*
SGoal 2	**Coordination and collaboration of the project with relevant stakeholders is conducted**		
SP 2.1	Manage the involvement of the relevant stakeholders in the project	None	*1. Coordinate with the relevant stakeholders who should participate in the project's activities* *2. Ensure that work products that are produced to satisfy commitments meet the requirements of the recipient projects* *3. Develop recommendations and coordinate the actions to resolve misunderstandings and problems with the product and product-component requirements, product and product-component architecture, and product and product-component design*
SP 2.2	Participate with relevant stakeholders to identify, negotiate, and track critical dependencies	None	*.1 Conduct reviews with relevant stakeholders* *.2 Identify each critical dependency* *.3 Establish need dates and plan dates for each critical dependency based on the project schedule* *.4 Review and get agreement on the commitments to address each critical dependency with the people responsible for providing the work product and the people receiving the work product* *.5 Document the critical dependencies and commitments* *.6 Track the critical dependencies and commitments and take corrective action as appropriate*
SP 2.3	Resolve issues with relevant stakeholders	None	*.1 Identify and document issues* *.2 Communicate issues to the relevant stakeholders* *.3 Resolve issues with the relevant stakeholders* *.4 Escalate to the appropriate managers those issues not resolvable with the relevant stakeholders* *.5 Track the issues to closure* *.6 Communicate with the relevant stakeholders on the status and resolution of the issues*

	LEVEL 3		
Ref.	**Practice (Requirement)**	**ANS-Life Cycle** mapping	**Comments**
SGoal 3	**The project is conducted using the project's shared vision**		
SP 3.1	Identify expectations, constraints, interfaces, and operational conditions applicable to the project's shared vision	None	*.1 Identify expectations, constraints, interfaces, and operational conditions about the organization and project that affect the project's shared vision* *.2 Elicit project members' perspectives and aspirations for the project* *.3 Create a description of the project's shared vision context*
SP 3.2	Establish and maintain a shared vision for the project	None	*.1 Hold meetings or workshops to create the project's shared vision* *.2 Articulate the project's shared vision in terms of purpose or mission, vision, values, and objectives* *.3 Reach consensus on the project's shared vision* *.4 Establish a strategy to communicate the project's shared vision both externally and internally* *.5 Make presentations suitable for the various audiences that need to be informed about the project's shared vision* *.6 Check that project and individual activities and tasks are aligned with the project's shared vision*
SGoal 4	**The integrated teams needed to execute the project are identified, defined, structured, and tasked**		
SP 4.1	Determine the integrated team structure that will best meet the project objectives and constraints	None	*.1 Determine the risks in the products and product suite* *.2 Determine likely resource requirements and availability* *.3 Establish work-product-based responsibilities* *.4 Consider organizational process assets for opportunities, constraints, and other factors that might influence integrated team structure* *.5 Develop an understanding of the organization's shared vision, the project's shared vision, and the organization's standard processes and organizational process assets applicable to teams and team structures* *.6 Identify alternative integrated team structures* *.7 Evaluate alternatives and select an integrated team structure*
SP 4.2	Develop a preliminary distribution of requirements, responsibilities, authorities, tasks, and interfaces to teams in the selected integrated team structure	None	.1 Assemble requirements and interfaces for integrated teams .2 Check that the preliminary distribution of requirements and interfaces covers all specified product requirements and other requirements .3 Define responsibilities and authorities for integrated teams .4 Designate the sponsor for each integrated team
SP 4.3	Establish and maintain teams in the integrated team structure	None	.1 Choose integrated team leaders .2 Allocate responsibilities and requirements to each integrated team .3 Allocate resources to each integrated team .4 Create each integrated team .5 Integrated team composition and structures are periodically evaluated and modified to best reflect project needs .6 When a change of team leader or a significant change of membership of the team occurs, review the integrated team composition and its place in the integrated team structure .7 When a change in team responsibility occurs, review the team composition and its tasking .8 Manage the overall performance of the teams
RskM	**RISK MANAGEMENT**		
The purpose of Risk Management is to identify potential problems before they occur, so that risk-handling activities may be planned and invoked as needed across the life of the product or project to mitigate adverse impacts on achieving objectives			
SGoal 1	**Preparation for risk management is conducted**		
SP 1.1	Determine risk sources and categories	None	*.1 Determine risk sources* *.2 Determine risk categories*
SP 1.2	Define the parameters used to analyze and categorize risks, and the parameters used to control the risk management effort	None	*.1 Define consistent criteria for evaluating and quantifying risk likelihood and severity levels* *.2 Define thresholds for each risk category* *.3 Define bounds on the extent to which thresholds are applied against or within a category*
SP 1.3	Establish and maintain the strategy to be used for risk management	None	
SGoal 2	**Risks are identified and analyzed to determine their relative importance**		
SP 2.1	Identify and document the risks	None	*.1 Identify the risks associated with cost, schedule, and performance in all appropriate product life-cycle phases* *.2 Review environmental elements that may impact the project* *.3 Review all elements of the WBS as part of identifying risks to help ensure that all aspects of the work effort have been considered* *.4 Review all elements of the project plan as part of identifying risks to*

	LEVEL 3		
Ref.	**Practice (Requirement)**	**ANS-Life Cycle** mapping	**Comments**
			help ensure that all aspects of the project have been considered *.5 Document the context, conditions, and potential consequences of the risk* *.6 Identify the relevant stakeholders associated with each risk*
SP 2.2	Evaluate and categorize each identified risk using the defined risk categories and parameters, and determine its relative priority	4.4.1	*.1 Evaluate the identified risks using the defined risk parameters* *.2 Categorize and group risks according to the defined risk categories* *.3 Prioritize risks for mitigation*
SGoal 3	**Risks are handled and mitigated, where appropriate, to reduce adverse impacts on achieving objectives**		
SP 3.1	Develop a risk mitigation plan for the most important risks to the project, as defined by the risk management strategy	None	*.1 Determine the levels and thresholds that define when a risk becomes unacceptable and triggers the execution of a risk mitigation plan or a contingency plan* *.2 Identify the person or group responsible for addressing each risk* *.3 Determine the cost-benefit ratio of implementing the risk mitigation plan for each risk* *.4 Develop an overall risk mitigation plan for the project to orchestrate the implementation of the individual risk mitigation and contingency plans* *.5 Develop contingency plans for selected critical risks in the event their impacts are realized*
SP 3.2	Monitor the status of each risk periodically and implement the risk mitigation plan as appropriate	None	*.1 Monitor risk status* *.2 Provide a method for tracking open risk-handling action items to closure* *.3 Invoke selected risk-handling options when monitored risks exceed the defined thresholds* *.4 Establish a schedule or period of performance for each risk-handling plan or activity that includes the start date and anticipated completion date* *.5 Provide continued commitment of resources for each plan to allow successful execution of the risk-handling strategy* *.6 Collect performance measures on the risk handling activities*
DAR	**DECISION ANALYSIS & RESOLUTION**		
The purpose of Decision Analysis and Resolution is to analyze possible decisions using a formal evaluation process that evaluates identified alternatives against established criteria			
SGoal 1	**Decisions are based on an evaluation of alternatives using established criteria**		
SP 1.1	Establish and maintain guidelines to determine which issues are subject to a formal evaluation process	None	*. Establish guidelines* *. Incorporate the use of the guidelines into the defined process where appropriate*
SP 1.2	Establish and maintain the criteria for evaluating alternatives, and the relative ranking of these criteria	None	*. Define the criteria for evaluating alternative solutions* *. Define the range and scale for ranking the evaluation criteria* *. Rank the criteria* *. Assess the criteria and their relative importance* *. Evolve the evaluation criteria to improve their validity* *. Document the rationale for the selection and rejection of evaluation criteria*
SP 1.3	Identify alternative solutions to address issues		*. Perform a literature search* *. Identify alternatives for consideration in addition to those that may be provided with the issue* *. Document the proposed alternatives*
SP 1.4	Select the evaluation methods	None	*. Select the methods based on the purpose for analyzing a decision and on the availability of the information used to support the method* *. Select evaluation methods based on their ability to focus on the issues at hand without being overly influenced by side issues* *. Determine the measures needed to support the evaluation method*
SP 1.5	Evaluate alternative solutions using the established criteria and methods	None	*. Evaluate the proposed alternative solutions using the established evaluation criteria and selected methods* *. Evaluate the assumptions related to the evaluation criteria and the evidence that supports the assumptions* *. Evaluate whether uncertainty in the values for alternative solutions affects the evaluation and address as appropriate* *. Perform simulations, modeling, prototypes, and pilots as necessary to exercise the evaluation criteria, methods, and alternative solutions* *. Consider new alternative solutions, criteria, or methods if the proposed alternatives do not test well; repeat the evaluations until alternatives do test well* *. Document the results of the evaluation*

	LEVEL 3		
Ref.	**Practice (Requirement)**	**ANS-Life Cycle** mapping	**Comments**
SP 1.6	Select solutions from the alternatives based on the evaluation criteria	None	*. Assess the risks associated with implementing the recommended solution* *. Document the results and rationale for the recommended solution*

This page is intentionally left blank.