



The European Organisation for Civil Aviation Equipment
L'Organisation Européenne pour l'Équipement de l'Aviation Civile

GUIDELINES FOR ANS SOFTWARE SAFETY ASSURANCE

This document is the intellectual and commercial property of EUROCAE.

It is presently commercialised by EUROCAE

This electronic copy is delivered to your company/organisation for internal use exclusively.

In no case, may it be re-sold, or hired, lent or exchanged outside your company/organisation.

ED-153

August 2009

GUIDELINES FOR ANS SOFTWARE SAFETY ASSURANCE

This document is the intellectual and commercial property of EUROCAE.

It is presently commercialised by EUROCAE

This electronic copy is delivered to your company/organisation for internal use exclusively.

In no case, may it be re-sold, or hired, lent or exchanged outside your company/organisation.

ED-153

August 2009

FOREWORD

1. This document, prepared by EUROCAE Working Group 64 (SWAL CS), was accepted by the Council of EUROCAE on 28th August 2009 taking note of the concerns expressed by UK CAA and DFS, which are summarized in the Attachment to this document (Appendix III).
2. EUROCAE is an international non-profit making organisation. Membership is open to European manufacturers of equipment for aeronautics, trade associations, national civil aviation administrations, users, and non-European organisations. Its work programme is principally directed to the preparation of performance specifications and guidance documents for civil aviation equipment, for adoption and use at European and worldwide levels.
3. EUROCAE performance specifications are recommendations only. EUROCAE is not an official body of the European Governments; its recommendations are valid as statements of official policy only when adopted by a particular government or conference of governments. The inclusion of references to standards published by other standardisation bodies, or extracts from those standards, does not imply any safety or regulatory liability.
4. Copies of this document may be obtained from:

EUROCAE

102, rue Etienne Dolet

92240 MALAKOFF

France

Telephone: 33 1 40 92 79 30

Fax: 33 1 46 55 62 65

E-mail: eurocae@eurocae.net

Website: www.eurocae.net

TABLE OF CONTENTS

1	INTRODUCTION.....	1
1.0	Purpose	1
1.1	Scope.....	2
1.2	Organisation of This Document.....	3
1.2.0	Structure	3
1.2.1	Mandating and recommending phrases.....	3
1.2.2	Table legend.....	4
1.3	Target Audience	4
1.4	Readership	5
1.5	References	5
1.6	Glossary.....	6
1.7	Abbreviations.....	10
2	DOCUMENT STRATEGY	11
2.0	Introduction	11
2.0.0	Relationship to EC Regulations	11
2.1	Strategy for SSAS:.....	13
2.1.0	SSAS Regulatory Requirements hierarchy	13
2.1.1	SSAS Regulatory Requirements and Objectives hierarchy	14
2.1.2	SSAS Implementation	16
2.1.3	Production of Argument.....	16
2.1.4	Requirement Correctness	16
2.1.5	Traceability	16
2.1.6	Unintended functions.....	17
2.1.7	Requirement satisfaction.....	17
2.1.8	Configuration Management.....	17
2.1.9	Assurance and Demonstration to NSA	17
2.1.10	Changes to software and to specific software	17
2.2	Strategy for Software lifecycle:	18
3	SOFTWARE SAFETY ASSURANCE SYSTEM	20
3.0	Software Safety Assurance System Overall Objectives.....	20
3.1	Software Safety Assessment Initiation	24
3.2	Software Safety Assessment Planning.....	26
3.3	Software Safety Requirements Specification	27
3.4	Software Safety Assessment Validation, Verification and Process Assurance	28
3.5	Software Safety Assessment Completion	29
3.6	Software Assurance Level.....	30
3.6.0	Introduction.....	30
3.6.1	Supporting Information for SWAL Allocation Process.....	30
3.6.2	SWAL Allocation Process	33
3.6.3	Examples of SWAL Allocation.....	36
3.6.4	Grading Policy	38
3.6.5	Similarity of Levels throughout Various Standards	40

4	PRIMARY LIFECYCLE PROCESSES.....	42
4.0	Introduction	42
4.1	Acquisition Process	43
4.2	Supply Process.....	46
4.3	Development Process.....	49
4.4	Operation Process	57
4.5	Maintenance Process	59
5	SUPPORTING LIFECYCLE PROCESSES	61
5.0	Introduction	61
5.1	Documentation Process.....	63
5.2	Configuration Management Process	64
5.3	Quality Assurance Process	68
5.4	Verification Process	70
5.4.0	Software Verification	71
5.4.1	Other Verifications.....	79
5.5	Validation Process	80
5.6	Joint Review Process	80
5.7	Audit Process.....	82
5.8	Problem/Change Resolution Process.....	86
6	ORGANISATIONAL LIFECYCLE PROCESSES.....	88
6.0	Introduction	88
6.1	Management Process.....	88
6.2	Infrastructure Process.....	91
6.3	Improvement Process.....	92
6.4	Training Process.....	93
7	ADDITIONAL ANS SOFTWARE LIFECYCLE OBJECTIVES	94
7.0	Introduction	94
7.1	Software Development Environment.....	95
7.2	COTS.....	95
7.2.0	Scope of COTS Section	95
7.2.1	System Aspects Relating to COTS in ANS.....	96
7.2.2	COTS Planning Process	96
7.2.3	COTS Acquisition Process.....	99
7.2.4	COTS Verification Process	102
7.2.5	COTS Configuration Management Process.....	104
7.2.6	COTS Quality Assurance	106
7.3	Tool Qualification	106
8	SOFTWARE SAFETY FOLDER	108
8.0	Introduction	108
8.1	Software Safety Folder Structure – Option 1.....	109
8.2	Software Safety Folder Structure – Option 2.....	111
8.3	ESARR6 and EC Regulation 482/2008 mapping	116

A	REFERENCE TO EXISTING SOFTWARE STANDARDS	117
A.1	Purpose	117
A.2	Software Safety Assurance System	118
A.3	Primary Lifecycle Processes	126
A.4	Supporting Lifecycle Processes	154
A.5	Organisational Lifecycle Processes.....	174
A.6	Additional ANS Software Lifecycle Objectives	179
A.7	ED109/DO-278 Omissions	181
B	ROLES AND RESPONSIBILITIES SCENARIOS	182
B.1	First Scenario – Contracted Major System Project	182
B.2	Second Scenario – Contracted SW	183
B.3	Third Scenario – ANSP Internal Development	185
C	TRACEABILITY WITH ESARR6	187
	APPENDIX I WG-64 MEMBERSHIP	193
	APPENDIX II HISTORY AND TERMS OF REFERENCE OF WG-64.....	194
	APPENDIX III ATTACHMENT TO ED-153	195
	APPENDIX IV IMPROVEMENT SUGGESTION FORM	196

CHAPTER 1

INTRODUCTION

1.0

PURPOSE

An increasing proportion of safety-critical Air Navigation Service (ANS) functions are being supported by software. This shift towards more automated ANS functions assumes at least equal, if not improved, levels of safety and efficiency provided by the overall system. Therefore, it is necessary to offer guidance on how to assure that the risk associated with deploying the software is reduced to a tolerable level.

This document provides:

- Recommendations and requirements on the major processes necessary to provide safety assurance for software in ANS systems, including:
 - A Software Assurance Level (SWAL) allocation process;
 - A list of objectives to be satisfied per SWAL;
 - A SWAL grading policy, ie the definition of a policy and its rationale to justify and substantiate the stringency of the objectives to be met per SWAL;
 - The identification of some appropriate activities (techniques or methods) to achieve these objectives, principally through referencing existing standards that offer guidance on how to provide evidence and confidence that these objectives are achieved and the SWAL is satisfied;

This document also provides:

- A recommended ANS Software lifecycle and its associated activities in support of achieving the objectives identified herein;
- A reference to other standards (focusing on ED-109/DO-278, ISO/IEC 12207, IEC 61508, CMMi and ED-12B/DO-178B) that relate to the identified objectives;

NOTE: *IEC12207, ED-12B/DO-178B, ED-109/DO-278 and IEC61508 consider a system as being hardware and software and consequently, the people and procedure aspects of a system are not taken into account by these four standards.*

- An assessment of the referenced standards' coverage of the recommended lifecycle and its associated activities for the development, operation and maintenance of ANS software.
- Guidance towards satisfying ESARR6 and EC Regulation 482/2008 (see Annex C). This guidance does not address the generation of the required safety arguments. Generation of these safety arguments that use the evidence generated by the SWAL processes are outside the scope of this document. Formal compliance with ESARR6 cannot be claimed as it is under the responsibility of Eurocontrol Safety Regulation Commission (SRC) to state it.

Figure 1 below encapsulates the main purposes of this document and shows how those purposes and other relevant documents are related.

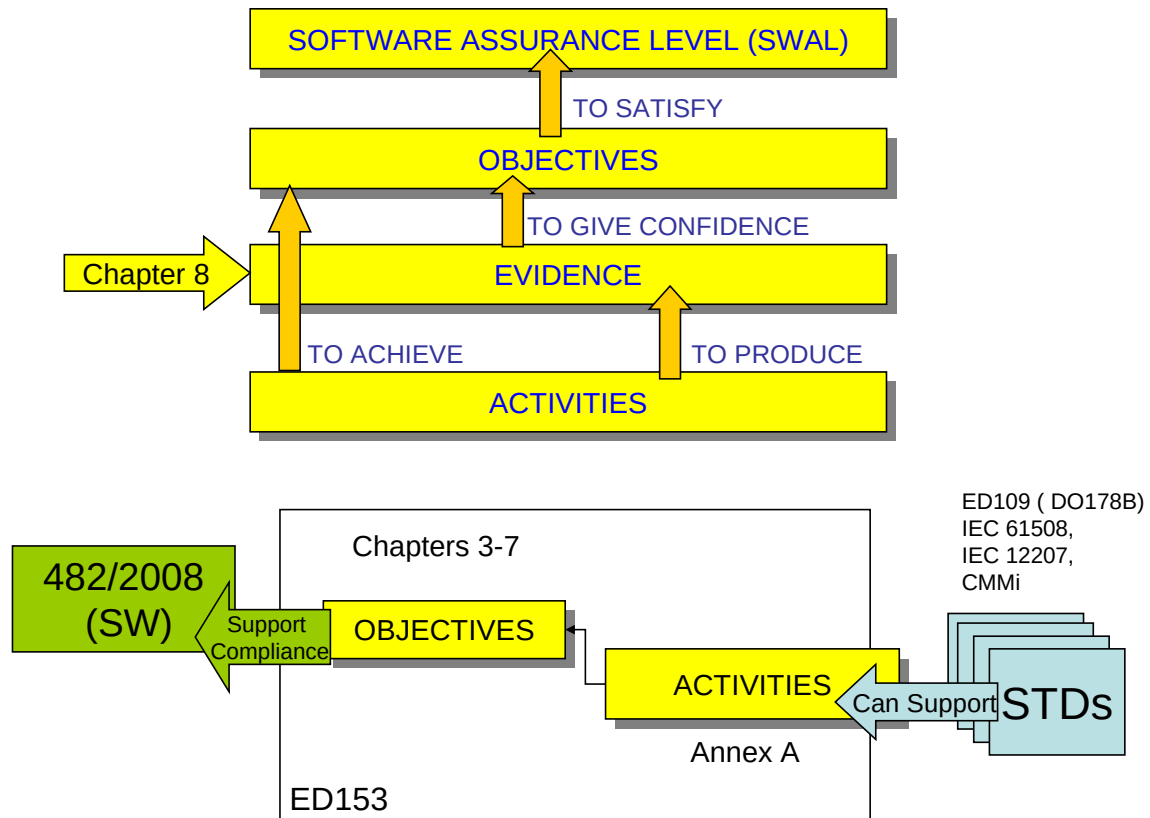


FIGURE 1: PURPOSE AND DOCUMENT RELATIONSHIPS RELEVANT TO THIS STANDARD

NOTE: Whilst the objectives described in this document support the achievement of many (but not all) of the articles within EC Regulation 482/2008, in no way can compliance with any such articles be claimed as this is the responsibility of regulatory and legislative authorities.

1.1

SCOPE

This document applies to software that forms part of an ANS system. The scope of this extends to the overall lifecycle of software within an ANS system, however this document considers aircraft software out of scope and is therefore limited to the “ground” segment of ANS.

This document assumes that a risk assessment and mitigation process has been undertaken along with an a priori system (where system includes people, procedure and equipment) safety assessment (eg a SAM-FHA and SAM-PSSA) with the results forming an input to this document.

This document is limited to software safety assurance and any references to software lifecycle data are made solely within the context of software **safety assurance**. Documentation not related to software lifecycle data is therefore out of scope.

This document covers:

- Guidance for an ANSP to establish a software safety assurance system;
- Guidance for software suppliers on the necessary software safety assurance regarding products and processes;
- A reference against which stakeholders can assess their own practices for software safety assurance of: specification, design, development, operation, maintenance, and decommissioning;
- A software assurance process that will promote interoperability through its common application to ANS software development.

1.2 ORGANISATION OF THIS DOCUMENT

1.2.0 Structure

This document includes the following chapters:

- Chapter 2: Document Strategy described each step of the strategy for the determination of software safety assurance.
- Chapter 3: Software Safety Assurance System lists objectives to set-up a Software Safety Assurance System including Software Assurance Level (SWAL) aspects.
- Chapter 4: Primary Lifecycle Processes lists the objectives per SWAL that belong to primary lifecycle processes.
- Chapter 5: Supporting Lifecycle Processes lists the objectives per SWAL that belong to supporting lifecycle processes.
- Chapter 6: Organisational Lifecycle Processes lists the objectives per SWAL that belong to organisational lifecycle processes.
- Chapter 7: Additional ANS Software Lifecycle Objectives lists additional objectives per SWAL that have been added due to ANS specific needs, such as the analysis of safety oriented standards, ANS particularities, and omissions by existing standards.
- Chapter 8: Software Safety Folder proposes two options for a Software Safety Folder (SSF) structure and identifies which objective output contributes to fill in this SSF.

The document is completed by the following annexes

- Annex A: Provides a mapping between objectives in this document and existing standards (ED-109/DO-278, ISO/IEC 12207, IEC 61508, CMMi and ED-12B/DO-178B)
- Annex B: Provides examples of how roles and responsibilities may be allocated between the ANSP and supplier per objective.
- Annex C: Proposes a mapping between ESARR6 and ED-153 objectives.

1.2.1 Mandating and recommending phrases

Each objective contains at least one “**shall**” statement, where “**shall**” means that this requirement must be satisfied to demonstrate compliance with this document.

Where more than one “**shall**” statement is included as part of an objective then, to achieve that objective, all of the “**shall**” requirements must be satisfied.

This document is not a regulatory requirement. It implies that, in case of a “shall” statement not being satisfied, alternative means sustained with rationale and evidence can be chosen and agreed with the NSA.

1.2.2 Table legend

The tables including Objective allocation per SWAL have to be read as follows (Cf: 3.6.4.0-criteria):

Legend (*used throughout this document*):

- | | |
|-------|---|
| ● | The objective to be achieved with independence. |
| ○ | The objective to be achieved. |
| Blank | Achievement of objective is at organisation's discretion. |

1.3 TARGET AUDIENCE

This document is specifically targeted at:

- **Safety Practitioners:** are responsible for establishing the link between the programme/project and the risk assessment and mitigation process. They provide the methodological support to the different steps of the risk assessment and mitigation process and the integration within the organisation's Safety Management System (SMS).

For example, the safety practitioners have to ensure that SWAL is allocated in accordance with material provided in Chapter 2, and that the allocated SWAL is validated.
- **Software Teams:** should apply this document to the relevant software.

For example, the software team is responsible for the implementation, verification and compliance with many of the objectives defined in this document.
- **Project/Programme Manager or Safety Manager:** Responsible for all or part of the Software Lifecycle (development, maintenance, operation...). Also responsible for allocating resources to assurance activities.
- **System Designers:** Responsible for system (people, procedures and equipment) architectural design including software. They will also allocate the SWAL.
- **Test and Integration Team:** Responsible for the development and conduct of the test and integration activities.

1.4 READERSHIP

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

Section \ Audience	Software Team	Safety Practitioner	Other roles (eg Programme/project Manager, Safety Manager)	System Designers	Test and Integration Team
Chapter 1 – Introduction	✓	📖	✓	✓	✓
Chapter 2 – Document Strategy	✓	📖	✓	✓	✓
Chapter 3 – Software Safety Assurance System	✓	📖	📖	📖	📖
Chapter 4 – Primary Lifecycle Processes	📖	📖	✓	✓	✓
Chapter 5 – Supporting Lifecycle Processes	📖	📖	📖	N/A	📖
Chapter 6 – Organisational Lifecycle Processes	✓	📖	📖	N/A	N/A
Chapter 7 – Additional ANS Software Lifecycle Objectives	✓	📖	✓	N/A	N/A
Chapter 8 – Software Safety Folder	📖	📖	✓	✓	✓
Annex A - Reference To Existing Software Standards	📖	✓	✓	N/A	N/A
Annex B – Roles and Responsibilities Scenarios	✓	✓	📖	N/A	N/A
Annex C – Traceability with ESARR6	N/A	📖	📖	N/A	N/A

📖: Detailed knowledge;

✓: Aware;

N/A: Not Applicable.

1.5 REFERENCES

- [1] ANS Software Lifecycle, by SAM-Software Task Force, SAF.ET1.STO1.1000-REP-01-00, edition 3.0 (21st December 2005).
- [2] Air Navigation System Safety Assessment Methodology, by Safety Assessment Methodology Task Force, SAF.ET1.STO1.1000-MAN-01-00, edition 2.1 (10/2006).
- [3] Recommendations For ANS Software, SAF.ET1.ST03.1000.GUI-01-00, edition 1.0, 21st December 2005.
- [4] ED-109/DO-278, Guidelines for the communication, navigation surveillance, and air traffic management (CNS/ATM) systems software integrity assurance (March, 2002), EUROCAE ED-109 & RTCA DO-278.
- [5] ISO/IEC 12207: Information technology - Software lifecycle processes, 1995.
- [6] IEC 61508: Functional safety of electrical/electronic/programmable electronic safety-related systems, December 1998.
- [7] RTCA/DO-178B/ED-12B, Software Considerations in Airborne Systems and Equipment Certification, RTCA & EUROCAE, 1992.
- [8] CMMI for Development (CMMI-DEV), Version 1.2, August 2006.
- [9] ESARR6, Software in ATM Systems, Version 1.0, 06/11/2003
- [10] ESARR4, Risk assessment and mitigation in ATM, Version 1.0, 05/04/2001

- [11] EUROCAE/ED-94B and RTCA/DO-248B, "Final Report for Clarification of ED-12B/DO-178B 'Software Consideration in Airborne Systems and Equipment Certification.
- [12] EC Regulation 482/2008
- [13] EC Regulation 552/2004

1.6

GLOSSARY

Acceptably Safe	Acceptably safe defines the target risk for an ANSP and is more demanding than tolerably safe.
Acquirer	An entity in charge of acquiring SW and/or SW lifecycle data from another entity so-called "Supplier". There are multiple type of relationships between Acquirer and Supplier as they can be part of the same organisation or from different organisations (contractual relationships). In some cases such formal relationships may not exist (e.g. in-house development) and so some requirements relating to the Acquirer may be justified as not applicable.
Adaptation Data	Data used to customise elements of the Air Traffic Management System for their designated purpose. <i>Note: 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:</i> - 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. <i>Examples include, but are not limited to:</i> 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. <i>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.</i> <i>Where appropriate, the generation of adaptation data should seek compliance with the guidance given in ADI DAL (Aeronautical Data Integrity Data Assurance Level). Except for generation related processes, compliance with this document should be shown to the same assurance level as the ANS code that uses them.</i>

ANS	Air Navigation Service
Approval	A means by which an authorised body so-called "the Approval Authority" gives formal recognition that a product, process, service, or operation conforms to applicable requirements.. <i>Note: For example, approval is a generic term to refer to certification, commissioning, qualification and initial operational capability, etc.</i>
Approval Authority	The relevant body responsible for the approval in accordance with applicable approval requirements.
Capacity	This term should be defined in the framework of the SW under consideration (the definition itself may vary depending on the type of SW eg real-time, database, HMI, ...).
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) <i>Note: See adaptation data</i>
Consequential failure	A SW failure or malfunction caused by another SW failure or malfunction.
Commercial Off The Shelf (COTS)	COTS software encompasses a wide range of software, including purchased software, Non-Developmental Items (NDI), and software previously developed without consideration of ED-153. 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.
Defined	When used in this document in the context of an objective, "defined" means "defined and documented".
External Mitigation Means (EMM)	Barriers or lines of defence, defined from the risk mitigation strategy, taken to control or prevent or reduce either: the likelihood of a hazard contributing to or causing an end effect; or the severity of an end effect. They may take various forms, including operational, procedural, functional, performance, and interoperability requirements or environment characteristics.
HMI	Human Machine Interface
Improvement	A modification has to be performed because the software performance, though compliant with requirements, has to be improved.
Internal Mitigation Means (IMM)	Barriers or lines of defence, defined from the risk mitigation strategy, taken to control or prevent or reduce the likelihood of an event contributing to or causing a hazard. They may take various forms, including operational, procedural, functional, performance, and interoperability requirements or environment characteristics.
Isolation	A design concept that ensures that the failure of one item does not cause a failure of another item. (derived from JAR AMJ 25.1309)

Mitigation Means	Barriers or lines of defence, defined from the risk mitigation strategy, taken to control, prevent or reduce either: the likelihood of an event developing further such that it can contribute to or cause an end effect; or the severity of an end effect. They may take various forms, including operational, procedural, functional, performance and interoperability requirements or environment characteristics.
Non-developmental software	Software not developed for the current demonstration.
Organisation	An administrative structure in which people collectively manage one or more projects as a whole, and whose projects share a senior manager and operate under the same policies.
Overload Tolerance	This term should be defined in the framework of the software under consideration as the definition itself may vary depending on the type of software (eg real-time, database, HMI). The definition may include, if appropriate, the SW performance range for some levels of load beyond some limits to be defined.
Perform Completely	In accordance with the approved software safety plan, in conformance with ANSP Safety Management System and in compliance with applicable safety regulatory requirements.
Resource Usage	The amount of resources within the computer system that can be used by the application software. Note: Resources may include main memory of various categories (such as static data, stack and heap), disc space and communications bandwidth and may include internal software resources, such as the number of files which may be simultaneously open.
Safety Requirement	A risk-mitigation means, defined from the risk-mitigation strategy that achieves a particular safety objective, including organisational, operational, procedural, functional, performance, and interoperability requirements or environment characteristics;
Software (SW)	Computer programs and corresponding configuration data, including non-developmental software (eg 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 Assurance Level (SWAL)	The software assurance level (SWAL) defines the level of rigour of the software assurances throughout the software lifecycle, supporting the demonstration that the EATMN software is acceptability safe.
Software Component	A component can be seen as a building block that can be fitted or connected together with other reusable blocks of software to combine and create a custom software application. In the framework of this document, it was found necessary to further developed the definition of “Software component” by providing the following information: A software component is the result of the first level of decomposition of the software architecture, so that requirements, actions, objects, input and output flows can be associated to that software component. Therefore, it can be a process if the application is based on a multi-process architecture or a thread if the architecture is mono or multi-process and multi thread or a set of actions or a set of objects with their associated methods or a state and its associated actions of a finite state machine.

Software Failure	The inability of a program to perform a required function.
Software Lifecycle data	Data that is produced during the software lifecycle to plan, direct, explain, define, record, or provide evidence of activities (including the software product itself). This data enables the software lifecycle processes, system or equipment approval and post-approval modification of the software product.
Software Malfunction	The inability of a program to perform a required function correctly..
Software Requirements Validity	Confirmation by examination and provision of objective evidence that software requirements are complete and correct, and are also compliant with the system safety requirements. ([9] Para 4.2)
Software Robustness	The behaviour of the software in the event of unexpected inputs, hardware faults and power supply interruptions, either in the computer system itself or in connected devices.
Software Timing Performances	The time allowed for the software to respond to given inputs or to periodic events, and/or the performance of the software in terms of transactions or messages handled per unit time.
Software Unit (SW Unit)	An element specified in the design of a Software Component that is separately testable. A software unit (SW Unit) is a low level of decomposition of the software architecture (can be the lowest). Requirements, actions, objects, input and output flows can be associated to that software unit that can be verified (and more specifically be tested). Therefore, it can be a file or a module or a single object with its associated methods or an interrupt or device handler.
Supplier	An organisation that enters into relationship with the acquirer for the supply of a software product. In some cases such formal relationships may not exist (e.g. in-house development) and so some requirements relating to the Supplier may be justified as not applicable.
System	An Air Navigation System is composed of People, Procedures and Equipment (Software, Hardware and HMI)
Tolerably Safe	Tolerably safe defines the target risk for a National Supervisory Authority (NSA) and is less demanding than acceptably safe.
Validation	Confirmation by examination and provision of objective evidence that the particular requirements for a specific intended use are fulfilled. [10]
Verification	Confirmation by examination of evidence that a product, process or service fulfils specified requirements. [9]

1.7**ABBREVIATIONS**

ADI	Aeronautical Data Integrity
ANSP	Air Navigation Service Supplier
ATM	Air Traffic Management
CM	Configuration Management
CMM	Capability Maturity Model
COTS	Commercial Off the Shelf Software
DAL	Data Assurance Level
EC	European Commission
EMM	External Mitigation Means
ER	Essential Requirement
ESARR	EUROCONTROL Safety Regulatory Requirement
FFP	Functional Failure Path
FHA	Functional Hazard Assessment
HMI	Human Machine Interface
IEC	International Electrotechnical Commission
IMM	Internal Mitigation Means
ISO	International Standard Organisation
MC/DC	Modified Conditions / Decision Coverage
NSA	National Supervisory Authority
PSSA	Preliminary System Safety Assessment
RBT	Requirement-Based Testing
SAM	Safety Assessment Methodology
SSA	System Safety Assessment
SSAS	Software Safety Assurance System
SSF	Software Safety Folder
SW	Software
SWAL	SoftWare Assurance Level

CHAPTER 2

DOCUMENT STRATEGY

2.0 INTRODUCTION

This section provides the strategy that has been used and that can be built upon by the users to relate the document purpose with the list of objectives (Chapter 3 to 7) and the means proposed by various standards to satisfy these objectives as listed in Appendix A.

2.0.0 Relationship to EC Regulations

This document provides guidance aiming to partially satisfy EC regulations (552/2004, 482/2008).

Therefore, both safety regulatory requirements (482/2008) as well as Interoperability (552/2004) aspects of Software are addressed by this document.

This section also gives the rationale for the current structure of this document.

It helps the reader to understand the various levels of guidance of this document:

- The Software Safety Assurance System (SSAS) implementation is described in Section 2.1 (to show partial compliance with 482/2008 and to 552/2004 Essential Requirement (ER) 3)
- The Software Lifecycle implementation is described in sections 3.6, chapter 5

SUPPORTING LIFECYCLE PROCESSES, chapter 6

ORGANISATIONAL LIFECYCLE PROCESSES, chapter 7

ADDITIONAL ANS SOFTWARE LIFECYCLE OBJECTIVES (to show partial compliance with 552/2004 Essential Requirement 7).

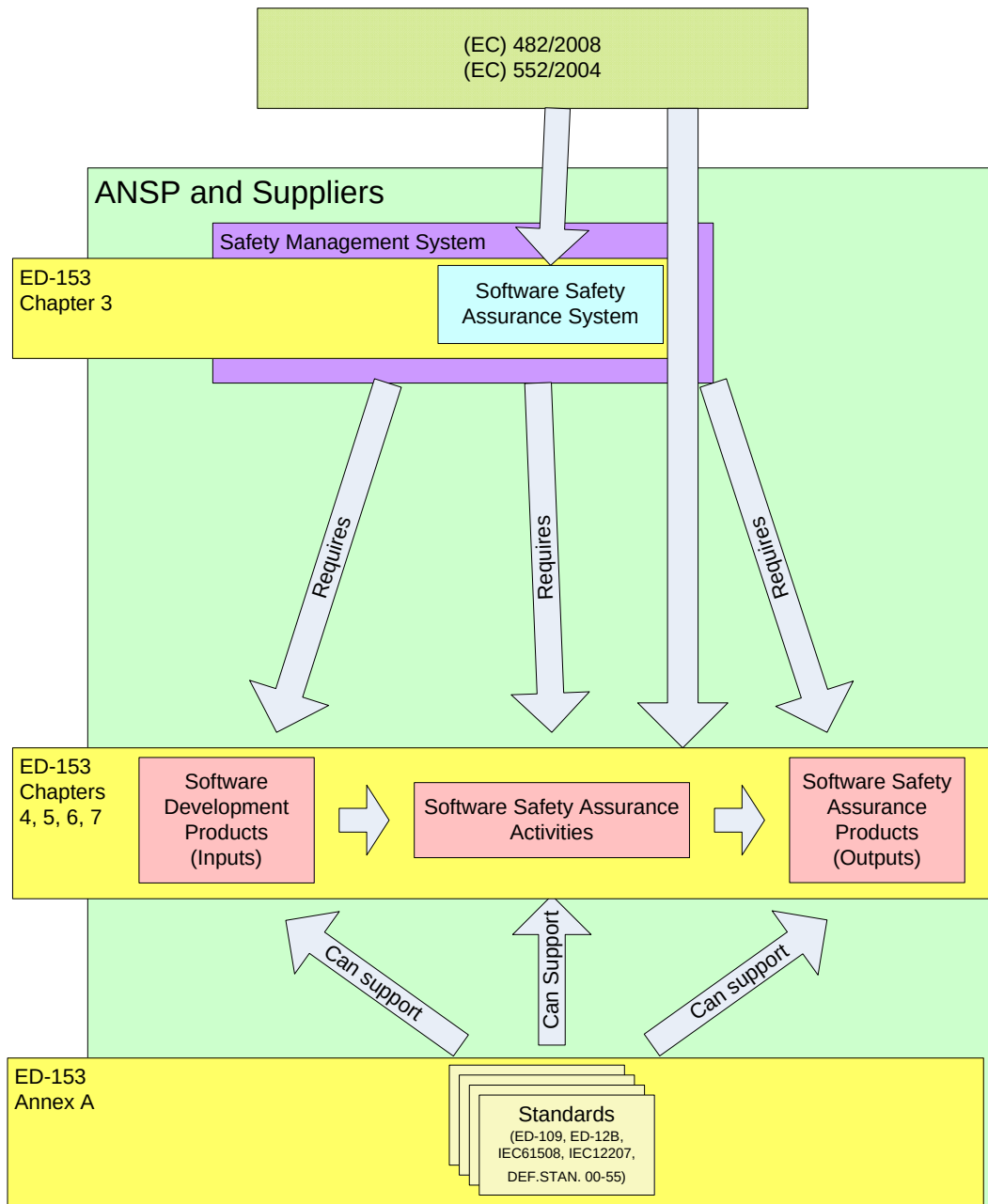


FIGURE 2: VARIOUS LEVELS OF GUIDANCE PROVIDED BY ED-153

2.1 STRATEGY FOR SSAS:

2.1.0 SSAS Regulatory Requirements hierarchy

This document provides guidance aiming to partially satisfy EC regulations (552/2004, 482/2008). The approach considers 482/2008 and refines the 552/2004 ER3 for software. It implies that demonstrating compliance with 482/2008 through the implementation of a SSAS subsequently supports the demonstration of compliance with 552/2004 ER3 for the software part of a constituent.

High level articles and objectives address ‘what’ has to be done and low level articles and objectives address ‘how’ it has to be done.

The table below provides the hierarchy of the regulatory requirements applicable to the SSAS.

482/2008 and 552/2004										
SSAS definition & implementation										
Article 3.1										
Name	SSAS Implementation	SWAL	Argument	Requirement Correctness	Traceability	Unintended functions	Requirement satisfaction	Configuration Management	Demo to NSA	Specific SW
482/2008 articles “high-level”	Article 4.1	Article 4.2;	Article 3.2 (1 st sentence)	Article 3.2.a	Article 3.2.b	Article 3.2.c	Article 3.2.d	Article 3.2.e	Article 3.3	
482/2008 articles “low-level”		Articles 4.4; 4.5; Annex I		Article 4.3.a; Annex II Part A	Article 4.3.d; Annex II Part D		Article 4.3.b; Annex II Part B	Article 4.3.c; Annex II Part C		Article 5

TABLE 1: 482/2008 REGULATORY REQUIREMENTS HIERARCHY

Example of Table interpretation: To implement the Configuration Management aspects of the SSAS, Article 3.2.e of EC Regulation 482/2008 has to be used as a reference. Article 3.2.e Requirements are refined into Article 4.3.c and Annex II Part C.

2.1.1 SSAS Regulatory Requirements and Objectives hierarchy

ED-153 includes various levels of objectives aiming to address some 482/2008 requirements. The table below provides the relationship between the various levels of 482/2008 requirements and ED-153 objectives.

482/2008 and ED-153 objectives mapping										
Name	SSAS Implementation		Argument	Requirement Correctness	Traceability	Unintended functions	Requirement satisfaction	Configuration Management	Demo to NSA	Changes to software and to specific software
482/2008 articles "high-level"	Article 4.1		Article 3.2 (1 st sentence)	Article 3.2.a	Article 3.2.b	Article 3.2.c	Article 3.2.d	Article 3.2.e	Article 3.3	
ED-153 Objectives "high-level"	3.0.1	3.0.15	3.0.17	3.0.2	3.0.3	3.0.4	3.0.6	3.0.7	3.0.8 3.0.16	
482/2008 articles "low-level"				Article 4.3.a Annex II Part A	Article 4.3.d Annex II Part D	-	Article 4.3.b Annex II Part B	Article 4.3.c Annex II Part C		Article 5
ED-153 Objectives "low-level"	See Table 3	4.4.5		4.3.4	4.3.15		Chap 5.4	Chap 5.2		3.0.12 3.0.13 Chap 7.2
ED-153 Objectives complementary "low-level"				3.1.1; 3.1.5; 3.3.1; 3.3.2; 3.3.3; 3.3.4; 3.4.X; 4.3.1; 4.3.2; 4.3.4; 4.3.5; 4.3.6; 4.3.9; 4.3.10; 4.3.11; 4.3.12; 4.3.13; 4.3.14; 4.3.20; 5.2.5; 5.4.3; 5.4.5; 5.4.6; 5.4.8; 5.4.9; 5.4.11; 5.7.2;	5.4.10	3.3.2; 3.3.3; 3.3.4; 4.4.2; 5.4.3; 5.8.3	3.4.4, 4.3.13; 4.3.14; 4.4.5; 4.5.3; 5.4.X;	5.2.X; 5.4.13;	4.1.7 Argument (note2) + (SSF 2 informativ e)	7.2.X

TABLE 2: 482/2008 REQUIREMENTS AND ED-153 OBJECTIVES MAPPING

NOTE 1: The objectives above are applicable as per the allocated SWAL.

NOTE 2: This document does not propose any guidance to develop the software argument or any example of it, except guidance to develop the safety argument for the system change (See SAM- Part IV GM I –Safety Case Development Manual). The system safety argument is the master argument embedding the software argument.

Example of table interpretation: ED-153 “high-level” objectives 3.0.7 aims at addressing (complying?) the “high-level” Article 3.2.e 482/2008 requirement. ED-153 Chapter 5.2 aims to address 482/2008 requirements. ED-153 Chapter 5.2 contains “low-level” objectives aiming to address 482/2008 “low-level” requirements stated in Article 4.3.c and in Annex II Part C

The table below expands the ED-153 objective 3.0.1 hierarchy.

ED-153 Objective 3.0.1					
482/2008 articles “low-level”	Article 4.1	Article 4.2 Annex I	Article 4.4.a	Article 4.4.b	Article 4.5
ED-153 Objectives “low-level”	3.0.1 4.4.2	3.0.2 3.0.5 3.0.14 Chap 3.6	3.0.9	3.0.8 3.0.10	3.0.11
ED-153 Objectives complementary “low-level”	3.1.X; 3.2.X; 3.3.X; 3.4.X; 3.5.X; 5.1.X;	4.4.5; 4.5.2; “shall” of Chapter 3.6	Those requirements associated with the derived SWAL	4.4.5; 4.5.2; 4.5.3;	4.4.X; 4.5.2; 4.5.3; 5.8.X;

TABLE 3: ED-153 OBJECTIVE 3.0.1 HIERARCHY (FROM TABLE 2)

NOTE: The low level objectives is not an exhaustive list as many low-level objectives contribute to many low level articles. It may be the role of an role of the argument to identify the complete set of low-level objectives for each article.

2.1.2 SSAS Implementation

This section provides an explanation on how to set up an SSAS (independently of the regulatory requirements). Therefore references are made to both the high level and low level objectives identified in tables 2 and 3 above.

In order to implement an SSAS (at organisation level as part of the Safety Management System), the following features should be addressed:

- Establishment of a SSAS: This is addressed in this document through Objective 3.0.1.
- Documented Processes: This is addressed in this document through Objective 3.2.1.
- Continuous improvement of processes: This is addressed in this document through Objectives 6.3.X.
- SWAL Allocation: This is addressed in this document through Objectives 3.0.5 and Chapter 3.6.
- Assurance rigour: This is addressed in this document through Objectives 3.0.8 and 3.0.9
- Software Safety Assurance Planning: This is addressed in this document through Objectives 3.2.X.
- Evidence of assurance activities having been undertaken: This is addressed in this document through Objectives 3.5.X and Software Safety Folder (informative).
- Independence of assurance and evidence: This is addressed in this document through Objectives 3.0.9 and SWAL tables
- Risk Assessment and Mitigation : This is addressed in this document through Objectives 3.3.X
- Risk Mitigations implemented in Software: This is addressed in this document through Objective 3.3.4
- In Service Monitoring: This is addressed in this document through Objectives 3.0.11 and 4.4.5.
- Software experience feedback: This is addressed in this document through Objectives 5.8.X
- Resolution of recorded Software malfunction or failures: This is addressed in this document through Objectives 5.8.X
- Modifications: This is addressed in this document through Objectives 3.0.12 and 5.8.2.
- Use of Tools (Software Development Tools, Verification Tools or Support tools): This is addressed in this document through Objectives 4.3.12, 4.3.17-19 and 7.3.X.

2.1.3 Production of Argument

No further refinement of the strategy relating to this high level objective in this version.

Chapter 2 as well as chapter 8 provide some basic strategy information that can be used to produce the argument.

2.1.4 Requirement Correctness

No further refinement of the strategy relating to this high level objective.

2.1.5 Traceability

No further refinement of the strategy relating to this high level objective.

2.1.6 Unintended functions

Unintended functions are functions in the software that are either performed in addition to those required or that are not performed on demand. They can result from, but are not limited to, the following examples:

- Execution of unspecified code including unintentional execution of Deactivated code: code which is present in the executable software but which is not intended to be executed in the given context. For example, different configurations for different sites may enable/disable different functions of the software;
- Incorrect implementation of SW specification ('Software fault / malfunction') leading to the execution of 'dead code';
- Code execution not performed at the right time or not on demand
- Incorrect SW specification capture (at any level, top level down to detailed design level)

This high level objective does not mean that, ultimately, there is not any unintended function in the software. It means that these functions are not activated or that the consequences of them being activated are justified by the safety analysis or that an effort has been made to produce evidence for this objective to be met with a level of rigor commensurate with the SWAL.

Ideally unintended functions should be identified. If this proves impossible, an alternative approach has to be defined.

In both cases, unintended functions have to be addressed by:

- Ensuring software requirements validity with a level of confidence commensurate with the SWAL. This is addressed in this document through Objective 4.3.4.
- Ensuring software requirements satisfaction with a level of confidence commensurate with the SWAL. This is addressed in this document through Objective 5.4.X.
- Ensuring that both the above, are appropriately associated with the software version being assessed (eg by configuration management). This is addressed in this document through Objective 5.2.X.

2.1.7 Requirement satisfaction

No further refinement of the strategy relating to this high level objective

2.1.8 Configuration Management

No further refinement of the strategy relating to this high level objective.

2.1.9 Assurance and Demonstration to NSA

The overall set of objectives aims to:

- provide confidence or assurance to the organisation that the software can be operated safely (This is addressed in this document in SSF-1 or in SSF2), and
- demonstrate to the NSA that the Software complies with the related regulatory requirements. This is addressed in this document in SSF-2.

2.1.10 Changes to software and to specific software

No further refinement of the strategy relating to this low level objective.

2.2 STRATEGY FOR SOFTWARE LIFECYCLE:

The approach chosen to address EC regulations related to the construction of systems aspects, namely 552/2004 ER7, considers that the software lifecycle has to be based on sound engineering principles. It implies that the use of best practice, extracted from published international standards (ISO/IEC12207) augmented with domain specific practices, supports the demonstration of compliance with 552/2004 ER7 for the software part of a constituent.

This strategy is complemented by listing the objectives of a Software lifecycle in this document in addition to the lifecycle related to the SSAS.

The table below provides the hierarchy of best practice that is applicable to software lifecycle. This includes both the high level objectives required to set-up a process for each lifecycle activity; and the low level objectives that refine the process further.

SW Lifecycle and ED-153 objectives mapping					
ED-153 categories	4 PRIMARY LIFECYCLE PROCESSES	5 SUPPORTING LIFECYCLE PROCESSES	6 ORGANISATIONAL LIFECYCLE PROCESSES	7 ADDITIONAL ANS SOFTWARE LIFECYCLE OBJECTIVES	8 SOFTWARE SAFETY FOLDER
ED-153 Process	4.1 Acquisition Process 4.2 Supply Process 4.3 Development Process 4.4 Operation Process 4.5 Maintenance Process	5.1 Documentation Process 5.2 Configuration Management Process 5.3 Quality Assurance Process 5.4 Verification Process 5.6 Joint Review Process 5.7 Audit Process 5.8 Problem/Change Resolution Process	6.1 Management Process 6.2 Infrastructure Process 6.3 Improvement Process 6.4 Training Process	7.1 Software Development Environment 7.2 COTS 7.3 Tool Qualification	8.1 Software Safety Folder Structure – Option 1 8.2 Software Safety Folder Structure – Option 2
ED-153 Objectives “high-level”	4.1.1; 4.2.1; 4.3.3; 4.4.1; 4.5.1	5.1.1; 5.2.1; 5.3.1; 5.4.1; 5.6.1; 5.7.1; 5.8.1	6.1.1; 6.2.1; 6.3.1; 6.4.1	7.1.1; 7.2.1; 7.3.1	
ED-153 Objectives “low-level”	4.1.X; 4.2.X; 4.3.X; 4.4.X; 4.5.X (excluding the high level objectives listed above)	5.1.X; 5.2.X; 5.3.X; 5.4.X; 5.6.X; 5.7.X; 5.8.X (excluding the high level objectives listed above)	6.1.X; 6.2.X; 6.3.X; 6.4.X (excluding the high level objectives listed above)	7.1.X; 7.2.X; 7.3.X (excluding the high level objectives listed above)	

TABLE 4: SW LIFECYCLE AND ED-153 OBJECTIVES MAPPING

The objectives above are applicable as per the allocated SWAL.

CHAPTER 3

SOFTWARE SAFETY ASSURANCE SYSTEM

Introduction

The Software Safety Assurance System encompasses the following tasks:

- Software Safety Assurance System overall Objectives specification
- Software Assurance Level allocation
- Software Safety Assessment
 - Software Safety Assessment Initiation
 - Software Safety Assessment Planning
 - Software Safety Requirements Specification
 - Software Safety Assessment Validation, Verification & Process Assurance
 - Software Safety Assessment Completion

3.0

SOFTWARE SAFETY ASSURANCE SYSTEM OVERALL OBJECTIVES

The aim of the Software Safety Assurance System (SSAS) is to ensure that appropriate processes are in place such that the risk associated with deploying the software is reduced to a tolerable level (ie the objectives of (EC) 482/2008 Article 3 are satisfied). Consequently, the Software Safety Assurance System is required to ensure requirements validity, satisfaction, traceability, non-interference and the configuration consistency of all data used in claiming compliance with (EC) 482/2008. The Software Safety Assurance System is required to be a part of an organisation's Safety Management System and consequently applies to any software under the responsibility of that organisation. The implementation of a SSAS is described in section 2.1.2.

Commission Regulation (EC) No 482/2008 sets some high level criteria for the Software Safety Assurance System and mandates that a SWAL allocation process be implemented that complies with the given criteria. Consequently, the following objectives have to be achieved by a Software Safety Assurance System.

NOTE: *The Software Safety Folder is defined and described in Chapter 8 of this document.*

Obj N°	ALs Objectives Title/Topic	OBJECTIVES	Output
3.0.1	Implementation	Annex A Section A.2.1 A Software Safety Assurance System shall ^{3.0.1.1} be defined and implemented (as part of the overall System Safety Assessment Documentation).	Software Assurance Manual (Part of the Safety Management System)
3.0.2	Requirements Correctness and Completeness	Annex A Section A.2.1 The software requirements shall ^{3.0.2.1} correctly and completely state what is required by the software, in order to meet the system safety objectives and system safety requirements as identified by the risk.	SSF1-Part VII SSF2 – Part I
3.0.3	Requirements Traceability Assurance	Annex A Section A.2.1 All software requirements shall ^{3.0.3.1} be traced to the level required by the SWAL.	SSF1 -Part VII SSF2 – Part I
3.0.4	Unintended Functions	Annex A Section A.2.1 The software implementation shall ^{3.0.4.1} not contain functions which may adversely affect safety or whose effect is not consistent with the safety analysis. <i>Note: This objective does not mean that there are no unintended functions in the software, but that these functions are not activated or that the consequences of them being activated is justified by the safety analysis.</i>	SSF1 -Part VII SSF2 – Part I
3.0.5	SWAL Allocation	Annex A Section A.2.1 The ANSP shall ^{3.0.5.1} ensure, as a minimum, that the Software Safety Assurance System allocates a SWAL to all operational ground ANS software. <i>Note: Refer to Chapter 2 Section 2 of this document for SWAL definitions and the allocation process.</i>	SSF1 - Part VII SSF2 – Part II
3.0.6	Requirements Satisfaction Assurance	Annex A Section A.2.1 The ANS software shall ^{3.0.6.1} satisfy its requirements with a level of confidence which is consistent with the SWAL allocated during risk assessment and mitigation eg PSSA.	SSF1 - Part VII SSF2 – Part I

Obj N°	ALs Objectives Title/Topic	OBJECTIVES	Output
3.0.7	Configuration Management Assurance	Annex A Section A.2.1 Any Assurance shall 3.0.7.1 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 (including specifications) that have been used in the production of that version.	SSF1 - Part VII SSF2 – Part I
3.0.8	Assurance Rigour Objective	Annex A Section A.2.1 The SWAL shall 3.0.8.1 give sufficient confidence that the ANS software can be operated, as a minimum, acceptably safely.	SSF1 - Part VII SSF2 – Part I
3.0.9	Assurance Rigour Criteria	Annex A Section A.2.1 The variation in rigour of the assurances per SWAL shall 3.0.9.1 include the following criteria: • required to be achieved with independence, • required to be achieved, • not required.	Software Assurance Manual (Part of the Safety Management System) SSF2 – Part II
3.0.10	SWAL Assurance	Annex A Section A.2.1 Assurance shall 3.0.10.1 provide confidence that SWAL is achieved. Note: Assurance may be based on direct or indirect arguments and evidence.	SSF1 - Part VII SSF2 – Part I
3.0.11	SWAL Monitoring	Annex A Section A.2.1 Assurance shall 3.0.11.1 be provided that once in operation the software meets its requirements with a degree of confidence commensurate with the SWAL, through monitoring. Feedback of ANS software experience shall 3.0.11.2 be used to confirm that the Software Safety Assurance System and the assignment of SWALs are appropriate. For this purpose, the effect resulting from any reported software malfunction or failure from ANS operational experience (reported according to the relevant requirements on reporting and assessment of safety occurrences), shall 3.0.11.3 be assessed in respect of their mapping to SWAL definition as per Chapter 2 of this document. (Reported Software malfunction or failures are output of the ANS occurrence reporting system as part of the ANSP Safety Management System)	SSF1 - Part VII SSF2 – Part II

Obj N°	ALs Objectives Title/Topic	OBJECTIVES	Output
3.0.12	Software Modifications	Annex A Section A.2.1 Any change to the software shall 3.0.12.1 lead first to the re-assessment of the safety impact of such a change on the system and then, depending on the impact, shall 3.0.12.2 lead to the re-assessment of the SWAL allocated to this software.	SSF1 - Part VII SSF2 – Part II
3.0.13	COTS	The same level of confidence, through any means chosen and agreed with the Approval Authority, shall 3.0.13.1 be provided for the same software assurance level for developmental and non-developmental ANS software (eg Commercial Off The Shelf software, etc). These means shall 3.0.13.2 give sufficient confidence that the software meets the safety objectives and requirements, as identified by the safety risk assessment and mitigation process.	SSF1 -Part VI SSF2 – Part II
3.0.14	Isolation	ANS software components that cannot be shown to be isolated from one another shall 3.0.14.1 be allocated the SWAL of the most critical (most demanding SWAL) of the components.	SSF1-Part II SSF2 – Part III
3.0.15	All on-line aspects of SW operational changes	The Software Safety Assurance System shall 3.0.15.1 deal specifically with software related aspects, including all on-line software operational changes (eg cutover/hot swapping).	SS1 -Part II SSF2 – Part I
3.0.16	Demonstration to NSA	The organisation shall 3.0.16.1 make available the required assurances, to the National Supervisory Authority, demonstrating that the objectives as per the allocated SWAL have been satisfied.	SSF1 -All SSF2 – All
3.0.17	Argument Production	The organisation shall 3.0.17.1 produce an argument demonstrating that the objectives of the allocated SWAL, as per this document, have been satisfied. <i>Note: No explicit guidance. Chapter 2 as well as chapter 8 provide some basic strategy information that can be used to produce the argument.</i>	No explicit guidance in this version

TABLE 5: SOFTWARE SAFETY ASSESSMENT SYSTEM OVERALL OBJECTIVES

3.1 SOFTWARE SAFETY ASSESSMENT INITIATION

Risk assessment and mitigation process (eg Functional Hazard Assessment-FHA and Preliminary System Safety Assessment-PSSA) assumptions and output should be confirmed as far as software can impact them.

Obj N°	ALs Objectives Title/Topic	OBJECTIVES	SWAL				Output
			1	2	3	4	
3.1.1	System Description	Annex A Section A.2.3.1 The Software purpose shall 3.1.1.1 be defined. Operational scenarios shall 3.1.1.2 be defined (eg HMI: Operator Handbook which defines the mode of operation and the human-machine interface; nominal mode, degraded mode). The Software and System functions and their relationships shall 3.1.1.3 be defined. Software boundaries shall 3.1.1.4 be defined (eg operational, time) Software external interfaces shall 3.1.1.5 be described.	●	●	○	○	SS1- Part I SSF2 – Part IV
3.1.2	Operational Environment	Annex A Section A.2.3.1 The Software and its environment (physical, operational, control functions, legislative etc) shall 3.1.2.1 be described in sufficient detail to enable the safety lifecycle tasks to be satisfactorily carried out.	●	●	○	○	SSF1 - Part I SSF2 – Part XX
3.1.3	Regulatory Framework	Annex A Section A.2.3.1 Applicable safety regulatory objectives and requirements shall 3.1.3.1 be identified.	○	○	○	○	SSF1 - Part I SSF2 – Part XX
3.1.4	Applicable Processes and guidance	Annex A Section A.2.3.1 Processes and Guidance applicable to the Software Assurance shall 3.1.4.1 be agreed.	○	○	○	○	SSF1 - Part I SSF2 – Part XX
3.1.5	Risk Assessment and Mitigation process Output	Annex A Section A.2.3.1 The system level risk assessment and mitigation identification shall 3.1.5.1 be reassessed at the software level to ensure it is consistent with the software architecture/design.	○	○	○	○	SSF1 - Part I SSF2 – Part XX

TABLE 6: SOFTWARE SAFETY ASSESSMENT INITIATION OBJECTIVES

NOTE: A system (People, procedures, and equipment) safety assessment has to be performed first, then its output re-assessed during the equipment safety assessment and then software safety assessment.

3.2 SOFTWARE SAFETY ASSESSMENT PLANNING

Obj N°	ALs Objectives Title/Topic	OBJECTIVES	SWAL				Output
			1	2	3	4	
3.2.1	Software Safety Assessment Approach	Annex A Section A.2.3.2 The overall approach for the Software Safety Assessment across Software Lifecycle shall 3.2.1.1 be defined.	●	●	○	○	SS1 - Part III SSF2 – Part II
3.2.2	Software Safety Assessment Plan	Annex A Section A.2.3.2 A plan describing the software safety assessment steps shall 3.2.2.1 be produced (eg approach, relations between safety assessment and software lifecycle, deliverables (content and date of delivery), 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)	●	●	●	●	SS1 - Part III SSF2 – Part II
3.2.3	Software Safety Assessment Plan Review	Annex A Section A.2.3.2 The Software Safety Assessment plan shall 3.2.3.1 be reviewed and commented for approval by NSA.	●	●	●	●	SS1 - Part III SSF2 – Part II
3.2.4	Software Safety Assessment Plan Dissemination	Annex A Section A.2.3.2 The Software Safety Assessment plan shall 3.2.4.1 be disseminated to the impacted parties.	●	●	●	○	SS1 - Part III SSF2 – Part II

TABLE 7: SOFTWARE SAFETY ASSESSMENT PLANNING OBJECTIVES

NOTE: These objectives recommend neither a specific packaging of a SW safety assessment plan, nor a separate plan (the plan can be part of a (sub-)system plan).

3.3 SOFTWARE SAFETY REQUIREMENTS SPECIFICATION

The objectives in this section describe the Software Safety Assessment defined in objective 3.2.1.

Obj N°	ALs Objectives Title/Topic	OBJECTIVES	SWAL				Output
			1	2	3	4	
3.3.1	Failure Identification	Annex A Section A.2.3.3 Potential failures shall _{3.3.1.1} be identified by considering various ways Software can fail and by considering the sequence of events that lead to the occurrence of the failure. A list of single, consequential and common modes of failure shall _{3.3.1.2} be drawn up. <i>Note: Guidance on Common Mode Analysis can be found in SAM and ED-79.</i>	●	●	●	●	SSF1 - Part II SSF2 – Part I
3.3.2	Failure Effects	Annex A Section A.2.3.3 The effects of failure occurrence shall _{3.3.2.1} be evaluated. The hazards associated with software failure occurrences shall _{3.3.2.2} be identified in order to further complete the list of hazards initiated during Risk Assessment and Mitigation process (eg FHA and further completed during PSSA).	●	●	●	●	SSF1 - Part II SSF2 – Part I
3.3.3	Assessment of Risk	Annex A Section A.2.3.3 The initial Risk Assessment and Mitigation process (eg FHA and further completed during PSSA) shall _{3.3.3.1} be revisited based upon the outcome of 3.3.1 and 3.3.2.	●	●	●	●	SSF1 - Part II SSF2 – Part I
3.3.4	Software Requirements Setting	Annex A Section A.2.3.3 Software Requirements shall _{3.3.4.1} be compliant with the Safety Objectives to which the Software contributes and System Safety Requirements. <i>Note: The definition of “compliant” has to be developed as part of the argument sustaining the demonstration of this objective. This definition should include the traceability with the above level of requirements, the demonstration of the necessity, sufficiency, appropriateness and relevance of the requirements to satisfy the above level of requirements.</i>	●	●	●	●	SSF1 - Part II SSF2 – Part I

TABLE 8: SOFTWARE SAFETY REQUIREMENTS SPECIFICATION OBJECTIVES

NOTE: Software “safety” requirements are not limited to the SWAL. See Chapter 4 Section 3 Objective 4.3.4. Consequently, in this document, the term “software requirements” is used, to reflect the fact that the same requirement can be classified as, for example a performance, security, interoperability or safety requirement. It is the role of the System Safety Assessment and of the Software Safety Assurance System to ensure that software requirements, that are necessary and sufficient to ensure an acceptably safe operation, have been identified and verified.

3.4 SOFTWARE SAFETY ASSESSMENT VALIDATION, VERIFICATION AND PROCESS ASSURANCE

Obj N°	ALs Objectives Title/Topic	OBJECTIVES	SWAL				Output
			1	2	3	4	
3.4.1	Software Safety Assessment Validation	Annex A Section A.2.3.4 The Software Safety Assurance System shall _{3.4.1.1} provide an approach to justify that Software Requirements are complete and correct.	●	●	●	●	SSF1 - Part III SSF2 – Part I
3.4.2	Software Safety Assessment Verification	Annex A Section A.2.3.4 The software Requirements shall _{3.4.2.1} be consistent with functions to mitigate the effects of the hazard and the Safety Objective of the hazards.	●	●	●	●	SSF1 - Part III SSF2 – Part I
3.4.3	Software Safety Assessment Process Assurance	Annex A Section A.2.3.4 The software Safety Assessment shall _{3.4.3.1} be performed completely. <i>Note: In accordance with the approved SW safety plan, in conformance with ANSP Safety Management System and in compliance with applicable safety regulatory requirements.</i>	●	●	●	●	SSF1 - Part III SSF2 – Part I
3.4.4	Software Safety Assurance	Demonstration and Assurance that SW requirements are satisfied shall be provided.	●	●	●	●	SSF1 - Part VII SSF2 – Part I

TABLE 9: SOFTWARE SAFETY ASSESSMENT VALIDATION, VERIFICATION AND PROCESS ASSURANCE OBJECTIVES

3.5 SOFTWARE SAFETY ASSESSMENT COMPLETION

Obj N°	ALs Objectives Title/Topic	OBJECTIVES	SWAL				Output
			1	2	3	4	
3.5.1	Document Software Safety Assessment Process Results	Annex A Section A.2.3.5 The Software Safety Assessment process results shall _{3.5.1.1} be documented.	☐	☐	☐	☐	SSF1 – Part XX SSF2 – Part II
3.5.2	Software Safety Assessment Documentation Configuration Management	Annex A Section A.2.3.5 Software Safety Assessment documentation shall _{3.5.2.1} be put under configuration management.	☐	☐	☐	☐	SSF1 – Part XX SSF2 – Part II
3.5.3	Software Safety Assessment Documentation Dissemination	Annex A Section A.2.3.5 Software Safety Assessment documentation shall _{3.5.3.1} be disseminated to impacted parties. <i>Note: This document does not presume who the impacted parties are. They are defined in accordance with the approved SW safety plan, in conformance with ANSP Safety Management System and in compliance with applicable safety regulatory requirements.</i>	☐	☐	☐	☐	SSF1 – Part XX SSF2 – Part II

TABLE 10: SOFTWARE SAFETY ASSESSMENT COMPLETION OBJECTIVES

3.6 SOFTWARE ASSURANCE LEVEL

3.6.0 Introduction

A SWAL scheme is a strategic project management aid that is intended to ensure that appropriate software safety assurance processes are used throughout the lifecycle of safety related software. SWALs identify sets of objectives to be applied to the software lifecycle processes and products.

A SWAL indicates the recommended rigour of the assurance processes to be used to generate the assurance evidence that will show whether the Software Requirements have been satisfied with a degree of rigour commensurate with the risk presented by the software. A SWAL provides a uniform level of rigor to which the development, operational transfer, maintenance, decommissioning and product functional assurances for the safety related software are produced. It is allocated upon the potential consequences of anomalous behaviour of software (the severity of the end effect of the software malfunction or failure and the likelihood of occurrence of the end effect) as determined by the system safety assessment process.

A SWAL does not replace Safety Requirements assigned to the Software. It is a Software Safety Requirement that has to be derived from the risk assessment and mitigation process.

Compliance with a SWAL requires the software team to follow planned and systematic actions to provide product, process or other evidence that can be argued to show that an adequate level of confidence and assurance exists that the software satisfies its requirements.

The allocation of a Software Assurance Level (SWAL) for software is part of the risk assessment and mitigation process e.g. it is identified during the Preliminary System Safety Assessment (SAM-PSSA) and confirmed during the System Safety Assessment (SAM-SSA). However, if the risk assessment and mitigation process does not define a SWAL allocation process it must be defined in the Software Safety Assurance System.

The SWAL allocation process shall ^{3.6.0.1} be performed during the design of an ANS system as part of a risk assessment and mitigation process.

The SWAL shall ^{3.6.0.2} be allocated by the “System team”

NOTE: *The term “system team” should be taken to include system designers and safety practitioners (see section 1.2.1).*

The software team can offer support and increased confidence that the allocated SWAL is appropriate by providing further understanding of the Software role, contribution, interference and interactions with the overall ANS system architecture.

3.6.1 Supporting Information for SWAL Allocation Process

This section includes information necessary to perform the SWAL allocation process described in section 3.6.2.

NOTE: *The reader is encouraged to note the definitions, provided in section 1.6, of internal and external mitigation means which are of particular use in this section.*

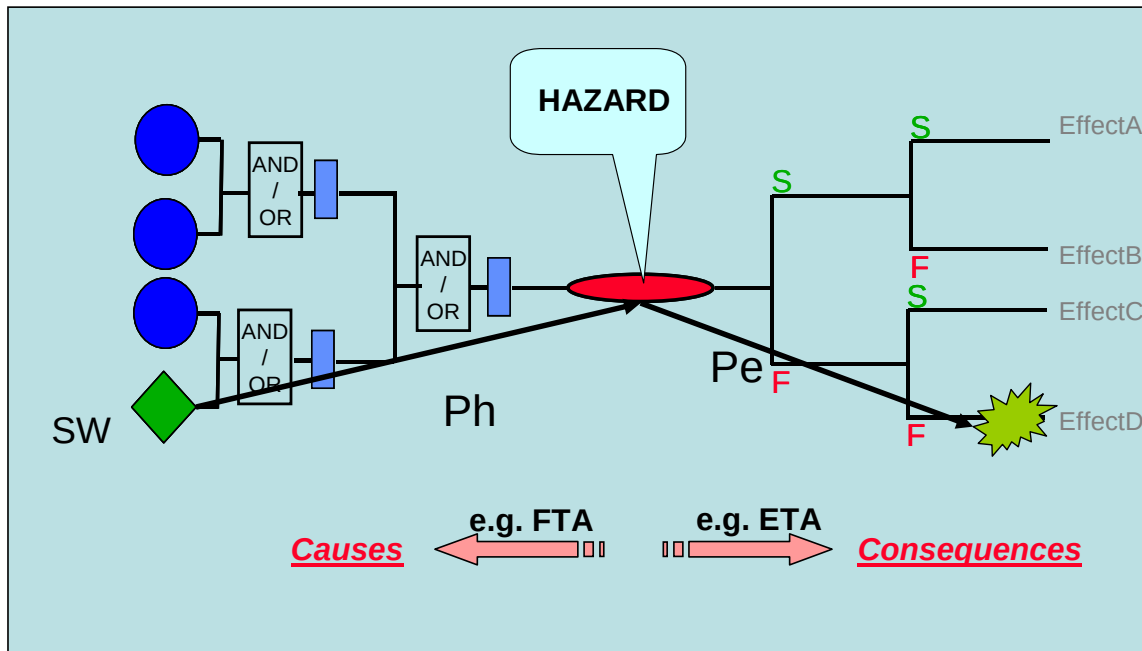


FIGURE 3: RELATIONSHIP BETWEEN SW MALFUNCTION OR FAILURE, HAZARD AND EFFECTS

Figure 3 illustrates the likelihood ($Ph \times Pe$) that, once software malfunctions or fails, this malfunction/failure could result in a certain effect, where Ph and Pe are defined as:

- Ph (identified during the safety assessment eg PSSA) is the likelihood that once the software has malfunctioned or failed, this malfunction/failure generates a hazard. Ph is commensurate with the ability of the remaining part of the architecture to mitigate the software malfunction/failure;
- Pe (identified during the safety assessment eg FHA) is the likelihood that the hazard generates an effect having a certain severity.

For each hazard, depending on the method used to set Safety Objectives, there can be:

- many likelihoods Pe (one Pe per effect of the hazard), to be assessed for each individual effect or;
- only one likelihood Pe (one for the Worst Credible effect).

The SWAL allocation process illustrated here is based on the second case, given above, and considers the likelihood of a given software behaviour generating the Worst Credible effect that it can cause.

The SWAL allocation process's sensitivity to inaccurate quantification needs to be reduced to an acceptable level due to:

- The lack of accuracy on Pe . Pe may include some un-quantifiable human, procedural and equipment aspects as well as quantifiable mitigation means or models such as Collision Risk Models (CRM). Thus Pe may not always be quantified precisely and may be stated to a given confidence eg $\pm 10\%$;
- The lack of accuracy on Ph . Ph may include some un-quantifiable human, procedural and equipment aspects as well as quantifiable mitigation means. Thus Ph may not always be quantified precisely and may be stated to a given confidence e.g. $\pm 10\%$;

- The difficulty in providing a failure rate for the Software. Whilst it is assumed that Software can fail and that a target failure rate objective may be set for the combination of hardware and software by the Risk Assessment and Mitigation process, it may only be possible to claim the failure rate for the combination of hardware to a given accuracy.

As it is difficult to quantify accurately and precisely these probabilities, common sense, expert judgement and other means (knowledgebase, lessons learned, incidents reports, equivalent field service experience) should be used to assess those probabilities.

As part of the safety assessment process (eg SAM-SSA), appropriate monitoring has to be put in place to ensure that these values are satisfied as P_e and P_h should be transposed into:

- Assumptions captured through Safety Requirements on the Operational Environment (P_e) (eg during SAM-FHA) and;
- Safety Requirements on the elements of the ANS System (P_h) (eg during SAM-PSSA).

Consequently, it is important to document the rationale for the allocated SWAL (end effect and the likelihood of generating such an effect, overall system design, operational environment) so that the satisfaction of the safety requirements and any assumptions may be verified during the software lifecycle.

3.6.1.0

Example of a SWAL Allocation (Air-Ground Datalink)

Example: Air-Ground Datalink Safety Requirements on Software

- Hazard: Undetected Corruption of a CPDLC message used for separation.
- Scope of the hazard: CPDLC/ACL Service (Controller Pilot Datalink Communication/ATC Clearance)
- Outcome of the System safety assessment process (e.g.. FHA)
 - Operational Environment: European Continental Domestic En-Route airspace (assuming airspace class definition, type and volume of traffic,...)
 - External Mitigation Means (EMM) to mitigate the consequence of the hazard once occurred:
 - EMM1: VHF **shall** be the primary means of communication,
 - EMM2: Radar surveillance **shall** be available,
 - EMM3: The airspace to which the clearance pertains is protected until the response is received.
 - Effects of the hazard:
 - Various effects can be identified depending on the scenario, thus depending on the EMM effectiveness: Severity Class 1 (e.g. mid-air collision), Severity Class 2 (serious incident such as serious loss of separation), Severity Class 3 (major incident such as large reduction of separation with ATCO or aircrew controlling the situation), Severity Class 4 (significant incident such as significant increase of workload of Aircrew and ATCO)
- Safety Objective (various methods can be applied, See SAM [2]) :
 - Either:
 - Worst Credible Effect: Severity Class 3 in a considering some External Mitigation Means

- Safety Objective for this hazard: The likelihood that an undetected corruption of a CPDLC message used for separation shall be no greater than Occasional (Occasional was set as 10-6/message due to a Safety Target for Severity Class 3 ST3 = 1E-4/fh, a Pe = 1/10 and 100 ANS hazards having a Worst Credible Effect of Severity Class3).
- Or:
 - Assess Pe for each couple (Hazard – Effect) by taking into consideration the effectiveness of the EMM involved in the Hazard - Effect scenario (not developed in this example).
- Outcome of the System safety assessment process (e.g. PSSA)
- Safety Requirements assigned to the SW:
 - o SR-ACL-4: response message shall indicate to which message it refers (Ph = 1).
 - o SR-ACL-8: Any processing (data entry/encoding/ transmitting/ decoding/ displaying) shall not affect the intent of the CPDLC message (Ph = 1).
 - o SR-ACL-18: The aircraft/ATSU shall be capable of detecting a corrupted CPDLC message (Ph = 1).

SWAL: The likelihood (Pe x Ph) that a SW failure (SW data corruption) could generate an effect having a Severity Class 3 was considered as “Very Possible” as:

- overall no effective and independent significant mitigation means (EMM) exist: Ph was considered as equal to 1 (no internal system architectural means exist to mitigate some SW data corruption) and;
- the External Mitigation Means (captured via Pe) are not very efficient once the pilot has started executing the clearance.
- Pe = 1/10, Ph = 1, thus Pe x Ph = 1/10 was considered as equivalent to “Very Possible”.

SWAL3 states the level of satisfaction of Safety Requirements: SR-ACL-4, SR-ACL-8 and SR-ACL-18 will have to be in accordance with SWAL3.

3.6.2 SWAL Allocation Process

3.6.2.0 Process description

To allocate a SWAL to an ANS software function, the following five steps **shall** ^{3.6.2.0.1} be performed:

- Using the definitions in 3.6.2.1, identify the likelihood (Pe x Ph) that, once software malfunctions or fails, this software malfunction/failure can generate an end effect which has a certain severity (*for each effect of a hazard*¹);
 - i. Justify the likelihood (see guidance in 3.6.2.2);
 - ii. Identify the SWAL for that combination (of severity and likelihood) using the matrix below;
 - iii. Repeat the two previous steps for all SW malfunction/failure.

¹ This has to be done for all effects of a SW malfunction. If only the worst credible effect is taken into consideration, then only the SW malfunction leading to that worst credible effect of the hazard will have been accounted for.

Likelihood of generating such an effect (Pe x Ph)	Effect Severity Class			
	1	2	3	4
Very Possible	SWAL1	SWAL2	SWAL3	SWAL4
Possible	SWAL2	SWAL3	SWAL3	SWAL4
Very Unlikely	SWAL3	SWAL3	SWAL4	SWAL4
Extremely Unlikely	SWAL4	SWAL4	SWAL4	SWAL4

Allocate the most stringent SWAL out of all contributions as the final SWAL of the software.

SWAL identification **shall** ^{3.6.2.0.2} assure that the number of SW malfunction/failure leading to such SWAL has been considered.

NOTE: *This requirement aims at ensuring that the cumulative aspect of risks due to SW malfunction/failure has been considered while taking into account the ability of a SWAL to provide adequate assurance for not only one, but a number of malfunction/failures having similar severity/likelihood effects. However, due to the intrinsic range of assurance provided by a given SWAL, the SWAL allocation can generally be performed by considering only the most severe software malfunction/failure.*

NOTE: *The qualitative definition of likelihood ("Very Possible", "Possible", "Very Unlikely", "Extremely Unlikely") can be interpreted with quantitative values.*

NOTE: *More than 4 SWALs can be defined according to ESARR 6 and the EC Regulation 482/2008, although 4 SWALs are used in ED-153.*

If the "Software" scope, as initially specified by the System Design Team and to which a SWAL is allocated during the risk assessment and mitigation process eg PSSA, leads to further detailed "Software" design that requires splitting the initial Software into parts, then each of those parts must also achieve the initially assigned SWAL level. An exception to this is where isolation can be demonstrated and a SWAL is allocated to each isolated software by applying the SWAL allocation process again. This may lead to different SWALs being allocated to the isolated software.

Reminder (Objective 3.0.14): ANS software components that cannot be shown to be isolated from one another **shall** ^{3.6.2.0.3} be allocated the software assurance level of the most critical of the isolated components.

Any introduction of software into operation **shall** ^{3.6.2.0.4} demonstrate that an acceptable level of safety is satisfied. If the risk assessment and mitigation process leads to a situation where the mitigation of the end-effect Severity Class 1, due to software malfunction or failure, is solely based on SWAL1 then the risk assessment and mitigation demonstration should justify that relying solely on SWAL 1 meets the acceptable level of safety.

The following considerations should be taken into account in relation to SWAL 1

- SWAL1 objectives and associated evidence are extremely stringent leading to a highly elevated; cost, development effort and difficulty in generating evidence.
- Wherever possible alternative system designs should be proposed including other mitigation means rather than relying on software being allocated a SWAL1.

If the retained strategy to mitigate the end-effect Severity Class 1 is solely based on SWAL1, the ANSP may be required to:

- Justify why an alternative design including other additional independent mitigation means is not used.

3.6.2.1 Likelihood Definition

The following qualitative definitions apply when assessing the likelihood ($P_e \times P_h$) during the SWAL allocation process for a system/SW having numerous years of operations. For SW/system having a short lifetime, either quantitative definitions may be defined (e.g. derived from the ANSP Risk Classification Scheme) or other qualitative appropriate statements may be proposed.

Very Possible: This effect, due to software malfunction or failure, will almost certainly occur throughout the system life.

Possible: This effect, due to software malfunction or failure, may occur throughout the system life.

Very Unlikely: This effect, due to software malfunction or failure, is expected to occur only in some exceptional circumstances throughout the system life.

Extremely Unlikely: This effect, due to software malfunction or failure, is not expected to occur throughout the system life.

These definitions are complemented by the guidance in section 3.6.2.2:

3.6.2.2 Likelihood Guidance

The following, non-exhaustive list of factors should be considered when justifying the likelihood ($P_e \times P_h$) during the SWAL allocation process:

- Number of mitigation means;
- Efficiency of mitigation means (reliability, availability, suitability, and how relevant and protective it is for the malfunction or failure being assessed);
- Completeness of the mitigation means failure mode analysis;
- Level of isolation of mitigation means (dependency between internal and external mitigation means, common modes between internal mitigation means, common modes between external mitigation means, physical location between mitigation means, particular risks);
- Nature of mitigation means (and diversity of nature with the element being assessed);
- Novelty of mitigation means;
- Complexity of mitigation means;

Difference in the context of collection of mitigation means efficiency (operational environment, usage/role of mitigation means) with the future context of element being assessed in case of reuse of an existing mitigation means. This criteria can be used to contribute in assessing the efficiency of the mitigation means.

The following guidance should be considered when justifying the likelihood ($P_e \times P_h$) during the SWAL allocation process:

Very Possible: No effective and independent significant mitigation means exist.

For example:

- The number and efficiency of the mitigation means cannot be relied upon to downgrade the SWAL.
- The number and efficiency of mitigation means from an individual point of view may seem to be significant enough, however the isolation criteria is seriously impaired (common mode of failure).

- The mitigation means are not yet proven or not yet proven in the future context.
- The complexity or novelty of the mitigation means prevents relying significantly on them (assessment complexity and completeness challengeable).
- Though mitigation means may exist, they are not accounted for.

Possible: Effective, isolated mitigation means exist.

For example:

- At least one effective, isolated mitigation means is proven.
- The number of mitigation means is reduced but the efficiency is significant.
- A number of isolated mitigation means each having moderate efficiency.

Very Unlikely: Very effective, isolated mitigation means exist.

For example:

- The number of mitigation means is limited but the efficiency is very significant.
- A significant number of isolated mitigation means each having significant efficiency.

Extremely Unlikely: Extremely effective, isolated mitigation means exist.

For example:

- The number of mitigation means is limited but the efficiency is extremely significant.
- A very significant number of isolated mitigation means each having significant efficiency.
- A significant number of isolated mitigation means each having very significant efficiency.
- The mitigation means are proven very significant in an identical context.

3.6.3

Examples of SWAL Allocation

3.6.3.0

Allocation of SWAL: looking at all effects

1st CASE: Safety Objectives were allocated by assessing all effects (eg SAM-FHA Chapter 3 Guidance Material G “Methods for Setting Safety Objectives” Methods 1 & 3 or SAM-FHA Guidance Material E “ED-125 Quantitative Model” [2]). So all effects, due to software malfunction or failure, are taken into consideration.

This Software will be allocated a SWAL = SWAL3 as it is the most stringent SWAL (for both hazards).

Effect Severity	1	2	3	4
Likelihood of generating such an effect (Pe x Ph)				
Very Possible	SWAL1	SWAL2	SWAL3	SWAL4
Possible	SWAL2	SWAL3	SWAL3	SWAL4
Very Unlikely	SWAL3	SWAL3	SWAL4	SWAL4
Extremely Unlikely	SWAL4	SWAL4	SWAL4	SWAL4

SW malfunction or failure leading to:

Hazard1: 

Hazard2: 

The way to read the table is the following:

For Hazard 1:

- If it is “Extremely Unlikely” that a SW malfunction or failure generates Hazard1 and an effect having a severity 1, then this SW should be allocated a SWAL4;
- If it is “Possible” that a SW malfunction or failure generates Hazard1 and an effect having a severity 2, then this SW should be allocated a SWAL3;
- If it is “Very Possible” that a SW malfunction or failure generates Hazard1 and an effect having a severity 3, then this SW should be allocated a SWAL3;
- If it is “Very Possible” that a SW malfunction or failure generates Hazard1 and an effect having a severity 4, then this SW should be allocated a SWAL4;

For Hazard 2:

- If it is “Extremely Unlikely” that a SW malfunction or failure generates Hazard2 and an effect having a severity 1, then this SW should be allocated a SWAL4;
- If it is “Extremely Unlikely” that a SW malfunction or failure generates Hazard2 and an effect having a severity 2, then this SW should be allocated a SWAL4;
- If it is “Very Unlikely” that a SW malfunction or failure generates Hazard2 and an effect having a severity 3, then this SW should be allocated a SWAL4;

If it is “Possible” that a SW malfunction or failure generates Hazard2 and an effect having a severity 4, then this SW should be allocated a SWAL4.

3.6.3.1

Allocation of SWAL: looking at Worst Credible case

2nd CASE: Safety Objectives were allocated using SAM-FHA Chapter 3 Guidance Material G “Methods for Setting Safety Objectives” Method 2 or 4 or SAM-FHA Guidance Material E “ED-125 Semi-Quantitative, Semi-Prescriptive or Fixed-Prescriptive Models”.

Only the worst credible scenario which has been used to set safety objectives is taken into consideration.

This Software will be allocated a SWAL = SWAL3 as it is the most stringent SWAL (for both hazards which have a worst credible hazard effect having a severity 3).

Effect Severity	1	2	3	4
Likelihood of generating such an effect (Pe x Ph)				
Very Possible	SWAL1	SWAL2	SWAL3	SWAL4
Possible	SWAL2	SWAL3	SWAL3	SWAL4
Very Unlikely	SWAL3	SWAL3	SWAL4	SWAL4
Extremely Unlikely	SWAL4	SWAL4	SWAL4	SWAL4

SW malfunction or failure leading to:

Hazard1:

Hazard2:

The way to read the table is the following:

For Hazard 1: As the Worst Credible effect of Hazard 1 has a Severity 3 (FHA result), then

- If it is “Very Possible” that a SW malfunction or failure generates Hazard1 and an effect having a severity 3, then this SW should be allocated a SWAL3;

For Hazard 2: As the Worst Credible effect of Hazard 1 has a Severity 3 (FHA result), then

If it is “Very Unlikely” that a SW malfunction or failure generates Hazard2 and an effect having a severity 3, then this SW should be allocated a SWAL4.

3.6.3.2 Misuse of a SWAL

Whilst quantitative failure rate objectives for the combination of hardware and software may be set by the Risk Assessment and Mitigation process, and achievement of these rates may be claimed quantitatively or qualitatively, their achievement may not be claimed through the application of a process.

Compliance to ED-153 at a given SWAL **shall** ^{3.6.3.2.1} not be used to claim a failure rate/reliability figure for the software.

3.6.4 Grading Policy

3.6.4.0 Criteria

This document provides guidance on objectives to be achieved using defined criteria but does not restrict the acceptability of evidence to demonstrate achievement of those objectives to some specific evidence.

The following criteria have been considered in order to establish the objectives applicable to each SWAL:

- Level of achievement necessary ie:

Legend (*used throughout this document*):

- | | |
|-------|---|
| ● | The objective to be achieved with independence: ie the resulting outputs or activities generated by the objective, have been verified or reviewed by peers, different departments of the organisation, different organisation etc |
| ○ | The objective to be achieved. |
| Blank | Achievement of objective is at organisation's discretion. |

- Nature of evidence.
 - Direct evidence: that which is produced by an activity taking place or software behaviour occurring, which is directly related to the claim being made;
 - Indirect (backing) evidence: that which shows that the Direct Evidence is both credible and soundly based.
- Depth of investigation according to the table below.

	Depth of investigation down to	Comments
SWAL4	Software Requirements	Software can be considered as a Black Box.
SWAL3	Architectural Design	Access to the First level of Software Components. A first level of architecture (by refining the software into software components) is specified and analysed, but these (first level) software components can be considered as black boxes
SWAL2	Code (Software Component)	Access to the source code;
SWAL1	Executable	Compiler, linker translation and editing outputs are examined;

NOTE: *The comments in this table are not necessarily applicable to COTS (see section 7.2)*

NOTE: *This depth of investigation is not applicable to every objective.*

3.6.4.1 Grading Policy Rationale

The following table provides rationale to help in understanding the aim of a SWAL including what kind of overall objective is intended and the kind of errors which are supposed to be prevented or tolerated.

Each SWAL adopts, in addition to its own criteria, the criteria of the less stringent SWALs.

NOTE: *This grading policy rationale is not applicable for every objective.*

SWAL	Errors to be prevented	Methods for demonstrating error prevention	Independence in performing the prevention	Rationale
4	Software Functional error	Validity of the software requirements, Testing	No	Black box testing to ensure that software performs its intended functional specifications
3	Design/low level functional error	Validity and testing of the architectural design requirements : Interface and functionality tests of primary components	No	Confidence augmentation by 'grey box testing' to provide better evidence of intended functioning.
2	Credible corruption until implementation (source code level)	Validity and testing of the detailed design requirements with structural coverage statement verification: state compatibility, hardware compatibility, coding standards (strong typing...)...	Yes for some objectives	Confidence augmentation by examination of each line of code
1	Credible corruption until implementation (executable code level)	Testing of the implementation (executable code) with deeper structural coverage, conditional/decisional activation(s) and pre- and post-conditions.	Yes for most of the objectives	Confidence augmentation by examination of the conditional/decisional activation(s) of the lines of code

TABLE 11: GRADING CRITERIA

3.6.5 Similarity of Levels throughout Various Standards

Multiple standards exist today that propose activities whose evidence addresses some of the objectives stated in this document. Where any such standard is used it **shall** ^{3.6.5.1} be complemented, where necessary, to satisfy the objectives/activities set for a given SWAL.

This section provides a reference against which standards may be assessed for the equivalence of assurance that they provide.

An analysis of the processes and their associated evidence (See Annex A) was used to provide the comparison of assurances. Annex A provides the detailed traceability between ED-153 objectives and the various standards (eg IEC 61508, ED-109) therefore the user can deduce detailed equivalence and compliance from this annex. It is the responsibility of the user to identify complementary activities for those objectives for which partial compliance is provided.

The comparison did not address assurances of quantified reliability or quantified integrity.

It is not intended that this document be used in isolation to demonstrate achievement of a SWAL but instead that it be used in conjunction with other standards.

The following figure provides a comparison of assurance levels when considering the process-oriented activities of ED153:

- ED-109;
- IEC61508.

NOTE: No existing standard fully covers all objectives of ED-153.

SW Standards

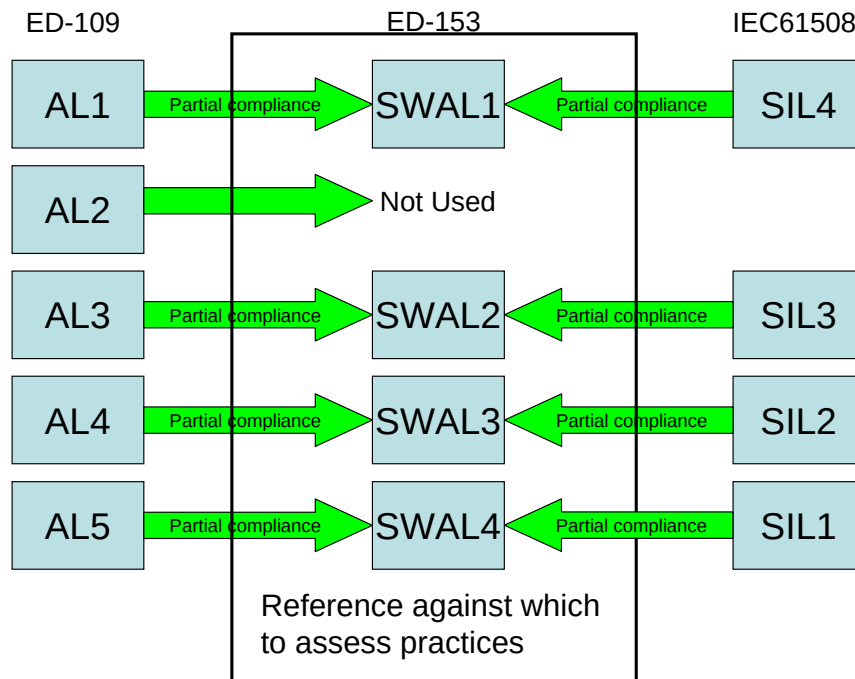


FIGURE 4: SOFTWARE STANDARDS

However, caution is recommended when claiming similar assurances, in particular with IEC61508, firstly due to the absence of a customised version of this generic standard to ANS, secondly due to the various interpretations that can be made of the Tables in Annexes A & B of IEC61508 Part 3 where some activities are “R”: recommended or “HR” Highly recommended.

Example: Table B.2 “Structure-based testing” which is “R” for SIL2.

This recommendation is similar to statement coverage which is recommended for SWAL2 and required for ED-109/DO-278 Level 3, consequently a SIL2 “similar” to SWAL3 should not include this aspect. Therefore if a SIL2 is required and includes this aspect, then the level of assurance becomes a “very strong” SWAL3 and gets close to SWAL2 (though SWAL2 includes additional objectives).

CHAPTER 4

PRIMARY LIFECYCLE PROCESSES

4.0 INTRODUCTION

This chapter lists the objectives, per SWAL, that belong to primary lifecycle processes.

Primary lifecycle processes consist of:

1. Acquisition process:
 - a. Initiation;
 - b. System Safety Assessment;
 - c. Request-for-Tender [Proposal];
 - d. Contract preparation and update;
 - e. Supplier monitoring;
 - f. Acceptance and completion.
2. Supply process:
 - a. Initiation;
 - b. Preparation of response;
 - c. Contract;
 - d. Planning;
 - e. Execution and control;
 - f. Review and evaluation;
 - g. Delivery and completion.
3. Development process:
 - a. System requirements analysis;
 - b. System architectural design;
 - c. Process implementation;
 - d. Software requirements analysis;
 - e. Software architectural design;
 - f. Software detailed design;
 - g. Software integration;
 - h. Software installation;
 - i. Standards/Rules Definition;
 - j. Standards/Rules;
 - k. Requirement Development Management;
 - l. Use of Requirement Management Tool;
 - m. Resource Management;
 - n. Rationale for Design choices;
 - o. Traceability;
 - p. Transition criteria;
 - q. Design tool;

- r. Use of a design tool;
 - s. Code generation tool;
 - t. Complexity constraints.
4. Operation process:
 - a. Process implementation;
 - b. Intended Operational Environment;
 - c. User support;
 - d. Software operation;
 - e. Performance monitoring.
 5. Maintenance process:
 - a. Process implementation;
 - b. SWAL allocation confirmation;
 - c. SWAL achievement;
 - d. Software Migration;
 - e. Software Decommissioning.

The objectives in a primary process are the responsibility of the organisation initiating and performing that process. This organisation ensures that the process is in existence and functional.

4.1 ACQUISITION PROCESS

This chapter also provides guidance for SW supplier/acquirer relationships. However, in some cases such formal relationships may not exist (e.g. in-house development) and this chapter may either be considered as not applicable or can be used to clarify the roles and responsibilities of the various units along the SW lifecycle.

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.

Objectives 4.1.1 to 4.1.3 (and 4.1.4 if the 'project' is at system or equipment level) are not specific to software but related to the system (eg FHA & PSSA).

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

Obj N°	ALs Objectives Title/Topic	OBJECTIVES	SWAL				Output
			1	2	3	4	
4.1.1	Initiation	Annex A section A.3.1 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. The acquirer shall 4.1.1.1 analyse and approve the system requirements eg against the user requirements. The system requirements shall 4.1.1.2 include business, organisational and user as well as safety, security, and other criticality requirements along with related design, testing, and compliance standards/rules and procedures. The acquirer shall 4.1.1.3 prepare, document and execute an acquisition plan. <i>Note: The scope of applicability of this objective is restricted to the contractual tasks to be performed by the acquirer in the framework of the contract with the supplier. Therefore this objective should be reconciled with objective 4.3.3.</i>	○	○	○	○	SSF1 - Part I SSF2 – Part XX
4.1.2	Risk Assessment and Mitigation Process – safety objectives	Annex A section A.3.1 The acquirer shall 4.1.2.1 determine how safe the system needs to be. The acquirer shall 4.1.2.2 perform a hazard analysis (eg FHA) and identify Safety Objectives for hazards.	●	●	●	●	SSF1 - Part II SSF2 – Part I
4.1.3	Risk Assessment and Mitigation Process – safety requirements	Annex A section A.3.1 The acquirer shall 4.1.3.1 determine (during the System Design phase) whether or not the proposed architecture is expected to achieve the Safety Objectives. The Acquirer shall 4.1.3.2 specify Safety Requirements (including allocation of a SWAL) for system elements.	●	●	●	●	SSF1 - Part II SSF2 – Part I