EUROPEAN ORGANISATION FOR THE SAFETY OF AIR NAVIGATION

EUROCONTROL

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

RECOMMENDATIONS FOR A.N.S SOFTWARE

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

Edition	:	1.0
Edition Date	:	21/12/2005
Status	:	Released Issue
Class	:	EATMP

EUROPEAN AIR TRAFFIC MANAGEMENT PROGRAMME

DOCUMENT IDENTIFICATION SHEET

DOCUMENT DESCRIPTION		
Document Title RECOMMENDATIONS FOR A.N.S SOFTWARE		
EWP DELIVERABLE REFERENCE NUMBER		
PROGRAMME REFERENCE INDEX SAF.ET1.ST03.1000.GUI-01-00	**EDITION :** 1.0 **EDITION DATE :** 21/12/2005	

Abstract

This document provides recommendations for Air Navigation System (ANS) Software.
This document is based upon the ANS Software Lifecycle defined in “ANS Software Lifecycle”

Keywords

Software	Safety Assurance	Software Assurance Level
Standards	Quality Assurance	SWAL
Safety Assessment	Software Life Cycle	

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

DOCUMENT STATUS AND TYPE

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

ELECTRONIC BACKUP

INTERNAL REFERENCE NAME :

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

DOCUMENT APPROVAL

The following table identifies all management authorities who have successively approved the present issue of this document.

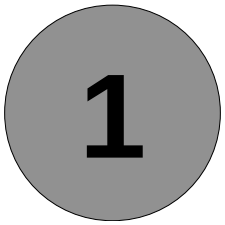
AUTHORITY	NAME AND SIGNATURE	DATE
Chairman of the EATMP Software Task Force	P.MANA	21/12/2005
Chairman of the Safety Assessment Methodology Task Force	P.MANA	21/12/2005
Chairman of the Safety Team	E.MERCKX	21/12/2005
DAP Director	G.PAULSON	21/12/2005

DOCUMENT CHANGE RECORD

The following table records the complete history of the successive editions of the present document.

EDITION	DATE	REASON FOR CHANGE	SECTIONS PAGES AFFECTED
0.1	17/06/99	First Issue	All
0.2	10/09/99	Second Working Draft Issue, after review by the Task Force	All
0.3	10/01/02	Third Working Draft Issue, after review by the Software Task Force. Title changes into “Recommendations for ANS Software”	All
0.4	10/06/02	Fourth Working Draft Issue, after review by the Software Task Force.	All
0.5	03/05/04	First Proposed Issue, after review by the Software Task Force and publication of ESARR6 V1.0.	All
0.6	14/06/2004	Second Proposed Issue after review by the Software Task Force	Intro ,Chapter 1 & 2
1.0	21/12/2005	First Released Issue after review by the Software Task Force	Intro ,Chapter 1, 2 & 3

TABLE OF CONTENTS



GENERAL INTRODUCTION

1 INTRODUCTION

1.1 PURPOSE

An increasing proportion of ANS 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 Systems 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.

Using as a basis the “ANS Software lifecycle” document, this document intends to provide:

- A reference against which stakeholders can assess their own practices for software specification, design, development, operation, maintenance and decommissioning.

- Recommendations on the major processes required to provide assurance for software in Air Navigation Systems, including:
    - An allocation process for Software Assurance Levels (SWAL);
    - A SWAL grading policy, ie the identification of a policy and its rationale to justify and substantiate increasing stringency of the objectives to be met per SWAL;
    - A list of objectives to be satisfied per SWAL;
    - The identification of some appropriate activities (techniques or methods) to achieve these objectives mainly by referencing existing standards (when existing) in order also to provide guidance on how to give confidence that these objectives are achieved, so that a SWAL is satisfied.

For reminder, the “ANS Software Lifecycle” document provides:

- a recommended ANS Software lifecycle and its associated activities (how) to achieve the objectives (as identified in this document);
- The reference to other standards (focusing on ED109/DO278, ISO/IEC 12207, IEC 61508, ED12B/DO 178B and CMMi);
- An assessment of their coverage of the recommended lifecycle and its associated activities for the development, operation and maintenance of Air Navigation System software.

1.2 SCOPE

The scope of this document is software that is part of an Air Navigation System. However, such as ESARR6 V1.0, this document does not aim at applying to aircraft software (as ED12B/DO178B is the domain practice and acceptable means of compliance), this document mainly focuses on “ground” segment of ANS.

This document is part of Air Navigation System Safety Assessment Methodology as SAM-Part IV Annex F (SAM [2]) and consequently requires an a priori safety assessment (done in accordance with SAM). Namely a FHA and PSSA should be done and their results are input to this document as Software is addressed mainly within SSA. PSSA provides recommendations to allocate SWAL which are exactly the same as those described in Chapter 2 of this document. Chapter 2 of this document is included to make “Recommendations for ANS Software” covering overall aspects of Software within an Air navigation system.

The scope of this document covers the overall lifecycle of software within an Air Navigation System.

1.3 STRUCTURE OF THE CURRENT ISSUE

This current issue of the report includes the following chapters:

- The purposes of **Chapter 2** are to provide :
    - The SWAL allocation process (illustrated by examples);
    - The SWAL grading policy principles.
- The purpose of **Chapter 3** is to list objectives to set-up a Software Safety Assurance System.
- The purpose of **Chapter 4** is to list objectives per SWAL that belong to primary life cycle processes:
    - Acquisition process
    - Supply process
    - Development process
    - Operation process
    - Maintenance process
- The purpose of **Chapter 5** is to list objectives per SWAL that belong to supporting life cycle processes:
    - Documentation process
    - Configuration management process
    - Quality assurance process
    - Verification process
    - Validation process
    - Joint review process
    - Audit process
    - Problem resolution process
- The purpose of **Chapter 6** is to list objectives per SWAL that belong to organisational life cycle processes:
    - Management process
    - Infrastructure process
    - Improvement process
    - Training process
- The purpose of **Chapter 7** is to list additional objectives per SWAL that do not belong to ISO/IEC 12207, but have been added due to:
    - The analysis of other standards more safety oriented (DO 178B/ED 12B and IEC 61508),
    - ATM particularities (Field service experience),
    - Omissions by existing standards (COTS).
- The purpose of **Chapter 8** is to propose a structure of a Software Safety Folder (SSF) and to identify which objective output should contribute to fill in this SSF.

1.4 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 apply those “Recommendations for ANS SW” for the relevant software.

For example, Software Team is responsible for the implementation of objectives of the correct SWAL and for the verification of their satisfaction.

Project/Programme Manager or Safety Manager.

1.5 READERSHIP

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

	Software Team	Safety Practitioner	Other roles (Programme/project Manager, Safety Manager, ..)	System Designers
Chapter 1 – Introduction	✓	🕮	✓	✓
Chapter 2 – SWAL (Part of PSSA)	✓	🕮	✓	🕮
Chapter 3 – Software Safety Assurance System	✓	🕮	🕮	✓
Chapter 4 – Primary lifecycle	🕮	🕮	✓	✓
Chapter 5 – Supporting Lifecycle	🕮	🕮	🕮	N/A
Chapter 6 – Organisational Lifecycle	✓	🕮	🕮	N/A

	Software Team	Safety Practitioner	Other roles (Programme/project Manager, Safety Manager, ..)	System Designers
Chapter 7 – Additional Lifecycle	✓	🕮	✓	N/A
Chapter 8 – Software Safety Folder	🕮	🕮	N/A	N/A

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

2 REFERENCES

[1] ANS Software Lifecycle
by SAM-Software Task Force
SAF.ET1.STO1.1000-REP-01-00, edition 3.0 (TBD)

[2] Air Navigation System Safety Assessment Methodology
by Safety Assessment Methodology Task Force
SAF.ET1.STO1.1000-MAN-01-00, edition 2.0 (04/2004)

3 GLOSSARY

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 and 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)
HMI	Human Machine Interface
Independence *	For software verification process activities, independence is achieved when the verification process activities are performed by a person(s) other than the developer of the item being verified; a tool(s) may be used to achieve an equivalence to the human verification activity.
Overload Tolerance*	The behaviour of the system in the event of, and in particular its tolerance to, inputs occurring at a greater rate than expected during normal operation of the system.
Resource Usage*	The amount of resources within the computer system that can be used by the application software. N*ote: 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.*
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 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.
Software Failure *	The inability of software to perform a required function correctly.

Software Lifecycle data*	Data that is produced during the software life cycle to plan, direct, explain, define, record, or provide evidence of activities. This data enables the software life cycle processes, system or equipment approval and post-approval modification of the software product.
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	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.

*: same as ESARR6 V1.0

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.

Note 2: additional information on “**Software Component**” definition

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.

Note 3: additional information on “**Software Unit**” definition

In the framework of this document, it was found necessary to further developed the definition of “Software Unit” by providing the following information:

A software 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.

SOFTWARE ASSURANCE LEVEL

Part of PSSA – Safety Requirements Specification

(See PSSA Chapter 3 Guidance Material A §2.4.2)

WARNING:
This Chapter has been added in this document in order to deliver a package that covers all the safety aspects of Software in ANS system/service. However, the SWAL allocation process remains within the remit of the ANS system design and specifically the PSSA (Preliminary System Assessment). SWAL is allocated by the System team (Designers and safety assessors), but NOT by the Software team.

If the “Software” scope as initially specified by the System Design Team and to which a SWAL is allocated during the PSSA leads to further detailed “Software” design ending up in splitting the initial Software into many “sub”-Software, then each of those “sub”-Software is allocated a SWAL which has to be “consistent” with the initial software SWAL (See Chapter 3 Obj 3.0.14). “Consistent” means that a demonstration of the initial Software SWAL satisfaction should be made. This demonstration may rely on argument such as isolation, partitioning of “sub-“software which should be proven to justify any reduction of SWAL for some specific “sub-“software.

This chapter aims at increasing software team awareness on the process to allocate SWAL, so that software team can support and contribute to enhance the confidence that the allocated

SWAL is the appropriate one by further understanding Software role, contribution, interference, interactions with the overall ANS system architecture.

1. INTRODUCTION

The Software Assurance Level definition is part of PSSA (Preliminary System Safety Assessment), however there is an obvious need to state them in Software related guidelines. Besides, this definition is part of the Software Safety Assurance System.

A Software Assurance Level (SWAL) relies upon planned and systematic actions necessary to provide confidence and assurance (through arguments, evidences or other means) that a software product or process satisfies given requirements.

SWAL is based upon the contribution of software to potential consequences of its anomalous behaviour as determined by the system safety assessment process. The Software Assurance Level implies that the level of effort recommended to showing compliance with Safety Requirements varies with the severity of the end effect of the software failure and the probability/likelihood of occurrence of the end effect.

SWAL is based upon criteria to evaluate a software product and/or a process to provide assurance that the product and/or process satisfies given requirements and can be relied upon to work correctly in its intended environment. The criteria are a set of items dependent upon the software assurance level and risk classification scheme, as determined by the system safety assessment process. The selected set of items is to be applied to the software lifecycle processes and data to demonstrate compliance to the documented process and correctness of the product.

The Software Assurance Level (SWAL) is a uniform measure of how the software was developed, transferred into operation, maintained and decommissioned (the process) and a measure of the ability of the product to function as intended (the product).

ANS software components with different software assurance levels are independent from each other (as designed and required during PSSA and demonstrated during SSA). In case independence is not achieved, assurances for the ANS software should be provided to the more rigorous software assurance level.

The assignment of a Software Assurance Level does not imply calculating a failure rate for that software. Software assurance levels or software reliability rates based on software assurance levels cannot be used by the system safety assessment process as can hardware failure rates.

SWAL does not replace Safety Requirements, but sets the level of at which Safety Requirements have to be satisfied.

Ex: Air-Ground Datalink

- Hazard: Undetected Corruption of a CPDLC message used for separation.
- 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).
- Safety Requirements:
    - SR-ACL-4: response message shall indicate to which message it refers.
    - SR-ACL-8: Any processing (data entry/encoding/ transmitting/ decoding/ displaying) shall not affect the intent of the CPDLC message.
    - SR-ACL-18: The aircraft/ATSU shall be capable of detecting a corrupted CPDLC message.
    - SWAL: SWAL3; so the level of satisfaction of Safety Requirements: SR-ACL-4, SR-ACL-8 and SR-ACL-18 will have to be iaw SWAL3.

2. SWAL DEFINITION

2.1 Basics of mitigation means influence

Basics of Mitigation Means Influence

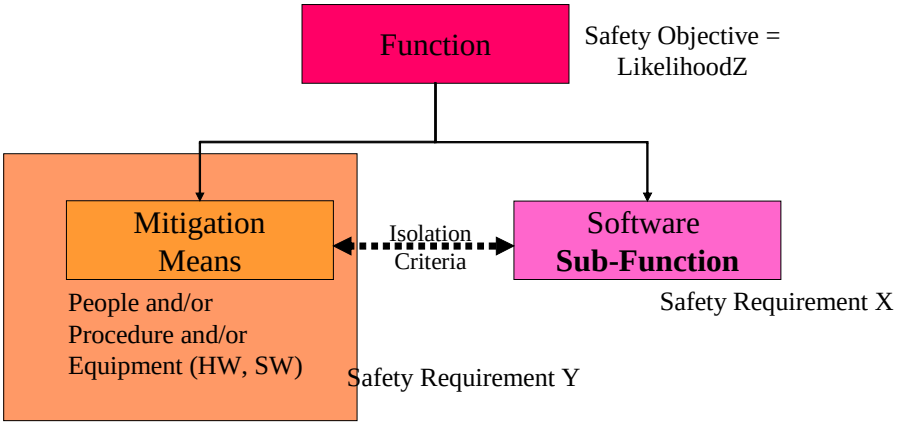


Figure 2.1.1: Basics of Mitigation Means Influence

As shown in Figure 2.1.1, “Mitigation means” are any kind of internal means (people and/or procedures and/or equipment) designed to control or prevent failures from causing harm and to reduce the expected effects of failures and hazards to an acceptable level. In Figure 2.1.1, “Mitigation Means” encompass all the other sub-functions that are part of the function (that has a safety Objective “LikelihoodZ”) and complement the “SW sub-function” to which a SWAL is being allocated.

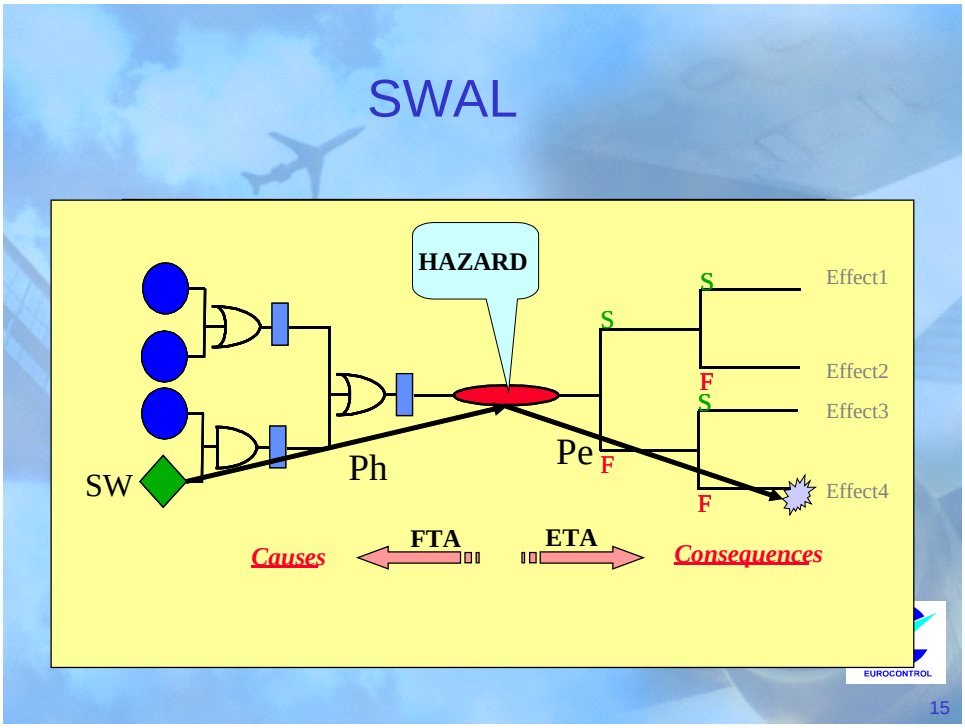


Figure 2.1.2: Relationship between SW failure, hazard and effects.

The likelihood (Ph x Pe) that, once software fails, this software failure could generate a certain effect is illustrated in the above figure 2.1.2:

- Ph (identified during the PSSA) is the probability that the software generates a hazard. Ph is commensurate with the ability (probability) of the remaining part of the architecture to mitigate the software failure;

- Pe (identified during the FHA) is the probability that the hazard generates an effect having a certain severity.

Depending on the method used to set Safety Objectives (See SAM-FHA Chapter 3 Guidance material G) there can be:

- up to four probabilities Pe (one Pe per effect of the hazard), to be assessed for each individual effect (when using the quantitative or the criticality methods for setting Safety Objectives) or;

- only one probability Pe (one for the Worst Credible effect when using the prescriptive or qualitative methods for setting Safety Objectives).

The SWAL allocation process has been designed such that importance of quantification is reduced to an acceptable level due to:

- The lack of accuracy on Pe as Pe includes not only quantifiable “barriers” or models such as Collision Risk Models (CRM) but mainly some human, procedural and equipment aspects. Thus Pe can not always be quantified precisely but remains at the level of an order of magnitude;

- The lack of accuracy on Ph as Ph includes not only quantifiable “barriers” but mainly some human, procedural and equipment aspects. Thus Ph can not always be quantified precisely but remains at the level of an order of magnitude;

- Strictly forbidding allocating a failure rate to the Software. Consequently, it is assumed that Software fails.

Therefore, it was assessed as key to keep the link to the end effect and the probability of generating such an effect (i.e. the criticality of the Software), so that the lack of accuracy in assessing Pe and Ph is mitigated by looking at the overall system design in its operational environment.

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

For example some qualitative rules may be added to support the application of such Pe x Ph evaluation e.g. number, level of independence, nature, novelty, complexity of mitigation means ….

Of course as part of the SAM-SSA, appropriate monitoring has to be put in place to ensure that these values are satisfied as Pe and Ph should be transposed into:

- Safety Requirements on the Operational Environment (Pe) during FHA and;

- Safety Requirements on the elements of the ANS System itself (Ph) during PSSA.

2.2 SWAL ALLOCATION PROCESS

To allocate a SWAL to an ATM software function, the following steps should be performed:

1. Identify the likelihood (Pe x Ph) that, once software fails, this software failure can generate an end effect which has a certain severity *(do that for each effect of a hazard*);*
2. Identify the SWAL for that couple (severity, likelihood) using the matrix here after;
3. This has to be done for all the hazards due to the software.

The final SWAL of the software is the most stringent SWAL.

***: This has to be done for all effects only if Methods 1 & 3 of setting Safety Objectives is used. If Methods 2 & 4 are used then only the worst credible effect will be taken into consideration. (See SAM-FHA Chapter 3 Guidance Material G "Methods for setting Safety Objectives).**

Effect Severity Likelihood of generating such an effect (Pe x Ph)	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

Very Possible: This effect will certainly occur due to software failure.

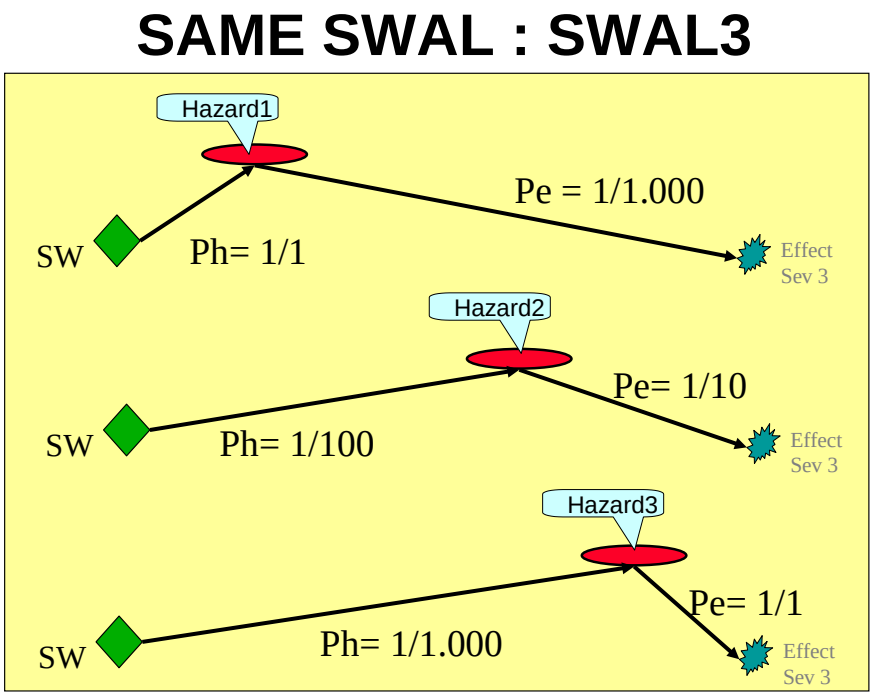
Possible: This effect may happen due to software failure.

Very Unlikely: it is not expected to have such an effect due to software failure more than exceptionally and in some specific circumstances throughout the system lifetime.

Extremely Unlikely: Such an effect is not expected to happen due to software failure throughout the system lifetime.
Note: It should be noted that SWAL1 is so stringent that it should nearly never be allocated for the following reasons:
1. SWAL1 means somehow that software "can directly kill once it fails and this failure happens in its usual mode of operation" as having a Severity1 effect is "Very Possible" (very limited means to mitigate SW failure(s)). This can only be tolerated in

extremely exceptional circumstances. Besides, this is not how ATM architecture is designed as of today;

2. SWAL1 is so demanding to be satisfied. As the objectives and associated evidences are so stringent, the cost and development duration and effort are very high. Consequently, another design should be proposed including other mitigation means to satisfy Safety Objectives without relying on software being allocated a SWAL1.



2.3 EXAMPLES OF SWAL ALLOCATION

2.3.1 Allocation of SWAL: looking at all effects

1st CASE: Safety Objectives were allocated using Method 1 or 3 (See FHA Chapter 3 Guidance Material G "Methods for Setting Safety Objectives). So all effects, due to Software 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 Likelihood of generating such an effect (Pe x Ph)	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

Hazard1:

Hazard2:

The way to read the table is the following:
For Hazard 1:

- If it is "Extremely Unlikely" that once SW fails, it generates Hazard1 and an effect having a severity 1, then this SW should be allocated a SWAL4;
- If it is "Possible" that once SW fails, it generates Hazard1 and an effect having a severity 2, then this SW should be allocated a SWAL3;
- If it is "Very Possible" that once SW fails, it generates Hazard1 and an effect having a severity 3, then this SW should be allocated a SWAL3;
- If it is "Very Possible" that once SW fails, it 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 once SW fails, it generates Hazard2 and an effect having a severity 1, then this SW should be allocated a SWAL4;
- If it is "Extremely Unlikely" that once SW fails, it generates Hazard2 and an effect having a severity 2, then this SW should be allocated a SWAL4;
- If it is "Very Unlikely" that once SW fails, it generates Hazard2 and an effect having a severity 3, then this SW should be allocated a SWAL4;
- If it is "Possible" that once SW fails, it generates Hazard2 and an effect having a severity 4, then this SW should be allocated a SWAL4.

2.3.2 Allocation of SWAL: looking at Worst Credible case

2nd CASE: Safety Objectives were allocated using Method 2 or 4 (See FHA Chapter 3 Guidance Material G "Methods for Setting Safety Objectives). So 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 Likelihood of generating such an effect (Pe x Ph)	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

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 once SW fails, it 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 once SW fails, it generates Hazard2 and an effect having a severity 3, then this SW should be allocated a SWAL4.

Recommendation: As far as software domain practices should always be as a minimum equivalent to SWAL4. That is where quality assurance meets safety assurance. This recommendation matches the ATM domain practices, where most industry companies are involved into some maturity model assessment (CMM or CMMi or SPICE) which states that practices (set by level) should be applied whatever software development. For these recommendations, the word “whatever” can be understood as “whatever the safety impact”.

Note: This recommendation matches current ATM domain practices for many ATM industry companies which are already CMM or CMMi or SPICE level 2 and many of them intend to go beyond. However, this statement does not intend to claim that being CMM or CMMi or SPICE level 2 directly provides an equivalent SWAL4 assurance.

The recommendation to have the domain practices equivalent, as a minimum, to SWAL4 can also be an argument when claiming that all Severity Class 4 end effect hazards do not need to be substantiated/developed in the safety argumentation (or so-called “safety case”).

2.3.3 Non-ATM example of allocation of SWAL

System: Navigation system (Hardware and software) in a car using GPS signal:

Assuming that the Severity Classification Scheme defines severity classes as following:

Severity Class 1: Accident

- Death (drivers and occupants and maybe other vehicle occupants or pedestrians);
- Vehicle(s) destroyed.

Severity Class 2: Serious Incident

- Serious injuries (maybe one death);
- Car destroyed.

Severity Class 3: Major Incident

- Major injuries;
- Car damaged.

Severity Class 4: Significant Incident

- Stress, increase of workload to recover the situation;
- Possibly minor car damages.

1°) Navigation system used for indication (as it is today)

OED (Operational Environment Definition): The following operational environment is assumed:

- Drivers have a driving license;
- Drivers have a good vision;
- Drivers have a situational awareness: other traffic, road signals (continuous line, one-way indication, priority signs, …), direction indication;
- Drivers know their final destination and the navigation system is used only for indication (as described in the User's Manual);
- Road regulations exist and are known by drivers.

Assuming that operational environment, let's assess the following hazard:

- Hazard1: Undetected credible corruption of direction indication (provided by navigation system).

When looking at all effects to allocate a SWAL:

Effect Severity Likelihood of generating such an effect	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

- It is "Extremely Unlikely" that once SW fails, it generates Hazard1 and an effect having a severity 1, then this SW should be allocated a SWAL4 as:
  - The driver controls his/her car and has to assess the credibility of the indication before applying it and so will not apply it (See OED). Thus the probability of applying a credibly corrupted indication is "Extremely Unlikely";
- It is "Extremely Unlikely" that once SW fails, it generates Hazard1 and an effect having a severity 2, then this SW should be allocated a SWAL4 as;
  - The driver controls his/her car and has to assess the credibility of the indication before applying it (See OED). Thus the probability of applying a credibly corrupted indication is "Extremely Unlikely";
- It is "Extremely Unlikely" that once SW fails, it generates Hazard1 and an effect having a severity 3, then this SW should be allocated a SWAL4 as:

    - The driver controls his/her car and has to assess the credibility of the indication before applying it (See OED). Thus the probability of applying a credibly corrupted indication is “Extremely Unlikely”;
- It is “Very Possible” that once SW fails, it generates Hazard1 and an effect having a severity 4, then this SW should be allocated a SWAL4 as:
    - The driver spends some time assessing the indication applicability, so it increases driver workload, may stress him/her. Maybe the physical location of the car is not the expected one, but this is impacting performance not safety.

As a conclusion, as far as the hazard “credible corruption of navigation system indication” is concerned, the SWAL allocated to the Navigation system in the OED as described is:

- **SWAL4.**

2°) Navigation system in command (futuristic use)

OED (Operational Environment Definition): The following operational environment is assumed:

- Drivers have to apply navigation system command;
- Drivers are only monitoring the system;
- Drivers do not need a situational awareness: other traffic, road signals (continuous line, one-way indication, priority signs, …), direction indication. Cars may not have windows!;
- Drivers have only to enter their final destination into the navigation system (as described in the User’s Manual);
- Road regulations exist and are known by navigation system.

Assuming that operational environment, let’s assess the following hazard:

- Hazard1: Undetected credible corruption of direction command (provided by navigation system).

When looking at all effects to allocate a SWAL:

Effect Severity Likelihood of generating such an effect	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

- It is “Very Possible” that once SW fails, it generates Hazard1 and an effect having a severity 1, then this SW should be allocated a SWAL1 as:
  - The driver applies the Navigation system command (See OED). Thus the probability of applying a credibly corrupted indication that can kill the driver (and other occupants and maybe other vehicle occupants) is “Very Possible”;
- It is “Very Possible” that once SW fails, it generates Hazard1 and an effect having a severity 2, then this SW should be allocated a SWAL2 as;
  - The driver applies the Navigation system command (See OED). Thus the probability of applying a credibly corrupted indication that can seriously injure the driver (and other occupants and maybe other vehicle occupants) and destroys the car is “Very Possible”;
- It is “Very Possible” that once SW fails, it generates Hazard1 and an effect having a severity 3, then this SW should be allocated a SWAL3 as:
  - The driver applies the Navigation system command (See OED). Thus the probability of applying a credibly corrupted indication that can seriously injure the driver (and other occupants and maybe other vehicle occupants) and destroys the car is “Very Possible”;
- It is “Very Possible” that once SW fails, it generates Hazard1 and an effect having a severity 4, then this SW should be allocated a SWAL4 as:
  - The driver applies the Navigation system command (See OED). Thus the probability of applying a credibly corrupted indication that can stress the driver (and other occupants and maybe other vehicle occupants) and damages the car is “Very Possible”.

As a conclusion, as far as the hazard “credible corruption of navigation system indication” is concerned, the SWAL allocated to the Navigation system in the OED as described is:

- **SWAL1.**

2.4 How not to allocate a SWAL

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 §2.2 as far as the SWAL allocation process is concerned, then equivalence between the allocated SWAL and its demonstration via a SIL can be achieved.

Generally speaking, a SWAL can not be used to claim that a piece of software cannot/will not fail more than a certain quantitative failure rate.

3. GRADING POLICY

3.1 Criteria

The following criteria have been identified to allocate objectives per Software Assurance Level:

- To Be Satisfied;
- Not To Be Satisfied;
- Independence: verification or review by peers, by different departments of the organisation, by a different organisation etc.

Legend:	
●	The objective should be satisfied with independence.
❍	The objective should be satisfied.
Blank	Satisfaction of objective is at organisation’s discretion.

- Nature of evidence.
  - Direct evidence = on the product (i.e. output of the process). It can cover product metrics, testing, verification, validation, service history…
  - Indirect evidence = on the process. It covers process metrics, inspection, …

    - Direct and indirect evidences will have a certain weight: e.g. metrics will be less convincing than tests…
- Depth of investigation.

This document does not intend to provide recommendations neither on acceptable combinations of evidences nor on the number of evidences.

3.2 Grading policy principles

The following approach has been agreed to help defining a grading policy. However this approach does not intend to be comprehensive and to cover all processes.

Amongst other criteria, the SWAL grading policy is articulated around the **level of depth of analysis, verification and evidence requested**.

This following depth of analysis, verification and evidence is requested:

- SW AL1: Executable
  It recommends looking at compilers/linkers output;
- SW AL2: Code (Software Component)
  It recommends having access to the source code;
- SW AL3: Architectural Design

  It recommends having 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;
- SW AL4: Software Requirements
  Software can be considered as a Black Box.

It is recommended that any ATM software should be allocated and should satisfy at least a SWAL4.

This means that this could be used as an argument within safety case to ease and reduce management of ESARR4 severity 4 end effects that software could cause or contribute to.

3.3 Grading Policy Rationale

The following table provides rationale, which sustain the allocation of objectives per SWAL.

The purpose of such rationale is to help understanding what is the aim of a SWAL, what kind of overall objective is intended, the kind of errors which is supposed to be avoided or still tolerated.

Note: the “hazard effects to be mitigated” column provides examples of software failure direct effect (and only direct).

SW AL	Example of Very Possible Hazard effects to be mitigated	Errors to be mitigated	Means of mitigation	Independence	Rationale
4	**Significant incidents (SC4)**	Functional behavior	Testing	No	to meet the requirements
3	**Major Incidents (SC3)**	Software integration or functionality	Interface and functionality tests of primary components	No	Black box testing alone cannot give sufficient evidence. Partial White box testing will complement it. Robustness tests are included.
		Functional behavior	Testing	Separate team	Graceful degradation.
2	**Serious Incidents (SC2)**	Reducing Credible corruption	statement verification: pre- and post-conditions, state compatibility, hardware compatibility, coding standards (strong typing…)…		reducing credible corruption White box testing
		Software integration	Interface and functionality tests of primary components	Separate team	
		Functional behavior	Testing	Separate team	
1	**Accidents (SC1)**	Reducing even more credible corruption and loss of functionality no greater than improbable	graceful degradation leading to a fail-safe state		
		reducing credible corruption	statement verification: pre- and post-conditions, state compatibility, hardware compatibility, coding standards (strong typing…)…		
		Software integration	Interface and functionality tests of primary components	Separate team	
		Functional behavior	Testing	Separate team	
				independent verification	

4. EQUIVALENCE OF LEVELS THROUGHOUT VARIOUS STANDARDS

The purpose of this section is to provide a reference against which to assess various standards in order to provide equivalence of assurance.

This equivalence of assurance was assessed performing an analysis of the processes and their associated evidences (See EATM - ANS Software Lifecycle).

This equivalence of assurance is not at the level of equivalence of quantified reliability or quantified integrity claimed by standards.

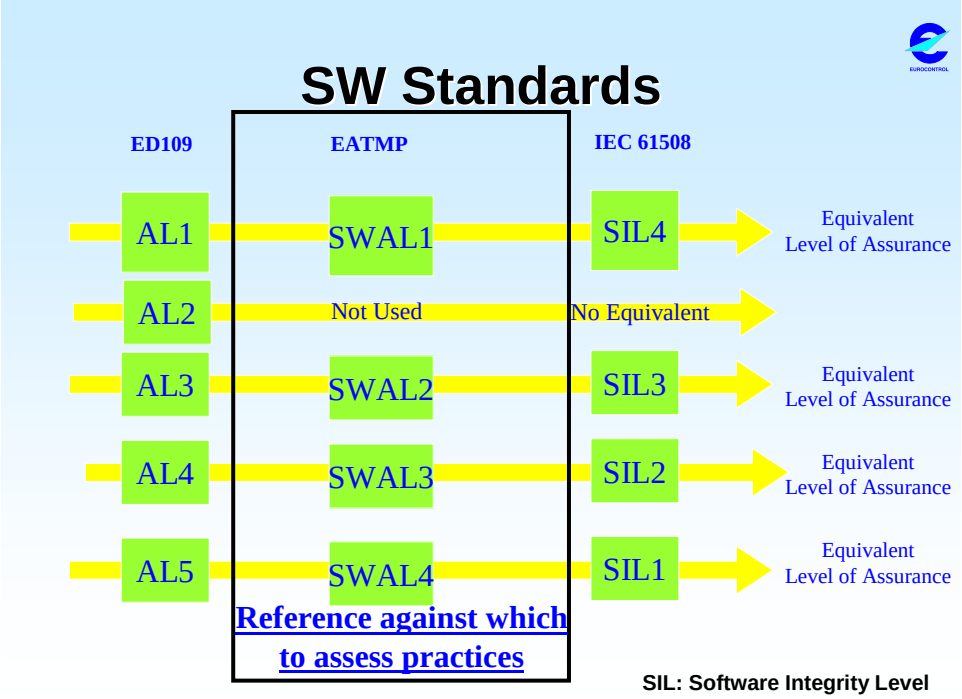
SWAL as defined in this document is the reference. Therefore it is recommended to allocate only SWAL.

However, it does not suggest that the demonstration of a SWAL satisfaction can be done using only this document as such, but that the use of any standard to support the SWAL satisfaction demonstration is aligned with this document.

In particular, parts missing to conform to EATM SWAL can be easily identified in any standard (see ANS Software Lifecycle). It does not mean that the standard can not be used, but that it has to be complemented.

The following figure provides an equivalence of levels of assurance for when considering the process-oriented activities:

- ED109;
- IEC61508.



However, caution should be recommended when stating equivalent assurance in particular with IEC61508 due first to the absence of customised version of this generic standard to ANS or ATM, second to the various interpretation 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.

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

This recommendation is equivalent to statement coverage which is recommended for SWAL2 and required for ED109/DO278 Level 3, consequently a SIL2 “equivalent” 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).

SOFTWARE SAFETY ASSURANCE SYSTEM

0 INTRODUCTION

Software Safety Assurance System encompasses the following tasks:

1) Software Safety Assurance System overall 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 OVERALL OBJECTIVES

The following objectives have to be satisfied whatever the SWAL (SoftWare Assurance Level) as they are dedicated to set-up a Software Safety Assurance System, which aims at (but is not limited to) allocating SWAL to software. This Software Safety Assurance System is a part of the Safety Management System of an organisation and consequently does not only apply to one software but to any software under the responsibility of that organisation.

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	ANS SW Lifecycle V2.0 Part I-Chap 1 §1 A Software Safety Assurance System should be defined, implemented and documented (as part of the overall System Safety Assessment Documentation).	Software Manual (Part of the Safety Management System)
3.0.2	Requirements Correctness and Completeness	ANS SW Lifecycle V2.0 Part I-Chap 1 §1 The software requirements should correctly state what is required by the software, in order to meet safety objectives and requirements, as identified by the risk assessment and mitigation process.	SSF Part VII
3.0.3	Requirements Traceability Assurance	ANS SW Lifecycle V2.0 Part I-Chap 1 §1 All software requirements should be traced to the level required by the SWAL (so to the level which is to be demonstrated)	SSF Part VII
3.0.4	Unintended Functions	ANS SW Lifecycle V2.0 Part I-Chap 1 §1 The software implementation should contain no functions, which adversely affect safety or whose effect is not consistent with the safety analysis.	SSF Part VII
3.0.5	SWAL Allocation	ANS SW Lifecycle V2.0 Part I-Chap 1 §1 and §2 Any ANS software intended for operational use should be allocated a SWAL. SWAL definitions are provided in Chapter 2 §2 of this document.	SSF Part II
3.0.6	Requirements Satisfaction Assurance	ANS SW Lifecycle V2.0 Part I-Chap 1 §1 The ANS software should satisfy its requirements with a level of confidence which is consistent with the SWAL allocated during PSSA.	SSF Part VII
3.0.7	Configuration Management Assurance	ANS SW Lifecycle V2.0 Part I-Chap 1 §1 Any Assurance 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 (including specifications) that have been used in the production of that version.	SSF Part VII
3.0.8	Assurance Rigour Objective	ANS SW Lifecycle V2.0 Part I-Chap 1 §1 The assurance and the levelling of assurance should give sufficient confidence that the ANS software can be operated, as a minimum, acceptably safely.	SSF Part VII

Obj N°	ALs Objectives Title/Topic	OBJECTIVES	Output
3.0.9	Assurance Rigour Criteria	ANS SW Lifecycle V2.0 Part I-Chap 1 §1 The variation in rigour of the assurance per software assurance level should be specified with the following criteria: – required to be achieved with independence, – required to be achieved, – not required.	Software Manual (Part of the Safety Management System)
3.0.10	SWAL Assurance	ANS SW Lifecycle V2.0 Part I-Chap 1 §1 Assurance should provide confidence that SWAL is satisfied.	SSF Part VII
3.0.11	SWAL Monitoring	ANS SW Lifecycle V2.0 Part I-Chap 1 §1 Assurance should be given that once in operation the software meets its SWAL 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)	SSF Part VII
3.0.12	Software Modifications	ANS SW Lifecycle V2.0 Part I-Chap 1 §1 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.	SSF Part V
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).	SSF Part VI
3.0.14	Independence	ATM software components that cannot be shown to be independent of one another should be allocated the software assurance level of the most critical of the dependent components.	SSF Part II
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).	SSF Part II

Note (Objective 3.0.4): This objective does not mean that there should not be any unintended function in the software, but that these functions are not activated or that the consequences of them being activated is not more critical than what the allocated SWAL ensures.

Note (Objective 3.0.10): Assurance may be based on direct or back-up arguments and evidences.

Note (Objective 3.0.11): This objective will also help in the future to assess whether the set of objectives recommended to satisfy a SWAL is the appropriate one (too demanding or not enough).

1.1 SOFWARE SAFETY ASSESSMENT INITIATION

Obj N°	ALs Objectives Title/Topic	OBJECTIVES	SWAL				Output
			1	2	3	4	
3.1.1	System Description	ANS SW Lifecycle V2.0 Part I-Chap 1 §3.1 The Software purpose should be defined. Operational scenarios should be defined (especially HMI: Operator Handbook should define the mode of operation and the human-machine interface). The Software/System functions and their relationships should be defined. Software boundaries should be defined (operational, time, ..) Software external interfaces should be described.	❍	❍	❍	❍	SSF Part I
3.1.2	Operational Environment	ANS SW Lifecycle V2.0 Part I-Chap 1 §3.1 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.	❍	❍	❍	❍	SSF Part I
3.1.3	Regulatory Framework	ANS SW Lifecycle V2.0 Part I-Chap 1 §3.1 Safety regulatory objectives and requirements should be defined.	❍	❍	❍	❍	SSF Part II
3.1.4	Applicable Standards	ANS SW Lifecycle V2.0 Part I-Chap 1 §3.1 Safety standards applicable to the Software should be defined.	❍	❍	❍	❍	SSF Part II
3.1.5	System FHA & PSSA Output	ANS SW Lifecycle V2.0 Part I-Chap 1 §3.1 The result of the system FHA (Functional Hazard Assessment) or PSSA (Preliminary System Safety Assessment) should be made available. Results of similar system safety assessment should be used as a reference.	❍	❍	❍	❍	SSF Part II

Note: A system (People, procedure, equipment) safety assessment has to be performed first, then the output of it are re-assessed during the equipment safety assessment and then software safety assessment.

1.2. SOFTWARE SAFETY ASSESSMENT PLANNING

Obj N°	ALs Objectives Title/Topic	OBJECTIVES	SWAL				Output
			1	2	3	4	
3.2.1	Software Safety Assessment Approach	ANS SW Lifecycle V2.0 Part I-Chap 1 §3.2 The overall approach for the Software Safety Assessment across Software Lifecycle should be defined and documented.	❍	❍	❍	❍	SSF Part III
3.2.2	Software Safety Assessment Plan	ANS SW Lifecycle V2.0 Part I-Chap 1 §3.2 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)	❍	❍	❍	❍	SSF Part III
3.2.3	Software Safety Assessment Plan Review	ANS SW Lifecycle V2.0 Part I-Chap 1 §3.2 The Software Safety Assessment plan should be reviewed and commented for suitability and approval.	❍	❍	❍	❍	SSF Part III
3.2.4	Software Safety Assessment Plan Dissemination	ANS SW Lifecycle V2.0 Part I-Chap 1 §3.2 The Software Safety Assessment plan should be made available to the interested parties.	❍	❍	❍	❍	SSF Part III

Note: these objectives do not 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).

1.3 SOFTWARE REQUIREMENTS SPECIFICATION

Obj N°	ALs Objectives Title/Topic	OBJECTIVES	SWAL				Output
			1	2	3	4	
3.3.1	Failure Identification	ANS SW Lifecycle V2.0 Part I-Chap 1 §3.3 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.	❍	❍	❍	❍	SSF Part II
3.3.2	Failure Effects	ANS SW Lifecycle V2.0 Part I-Chap 1 §3.3 The effects of failure occurrence should be evaluated. The hazards associated with failure occurrences should be identified in order to further complete the list of hazards initiated during FHA and further completed during PSSA.	❍	❍	❍	❍	SSF Part II
3.3.3	Assessment of Risk	ANS SW Lifecycle V2.0 Part I-Chap 1 §3.3 The purpose of this objective is to classify hazards according to the severity of their effects in order to further complete the list of hazards and Safety Objectives initiated during FHA and further completed during PSSA.	❍	❍	❍	❍	SSF Part II

Obj N°	ALs Objectives Title/Topic	OBJECTIVES	SWAL 1	SWAL 2	SWAL 3	SWAL 4	Output
3.3.4	Software Requirements Setting	ANS SW Lifecycle V2.0 Part I-Chap 1 §3.3 For each function and combination of functions to which software participates, – Refine the functional breakdown; – Evaluate system architecture(s); – Apply risk mitigation strategies; – Apportion Safety Objectives into Safety Requirements; – Balance Safety Requirements. The purpose of this objective is to re-assess and to further complete the output of the PSSA when performing software-related activities. Software Requirements should be compliant with the System Safety Objectives. (System Safety Objectives specify the maximum acceptable frequency of occurrence of a hazard).	❍	❍	❍	❍	SSF Part II
3.3.5	SWAL Allocation	ANS SW Lifecycle V2.0 Part I-Chap 1 §3.3 A SWAL should be allocated to the software.	❍	❍	❍	❍	SSF Part II

Note: Software “safety” requirements are not limited to the SWAL identification. See Chapter 4 §3 Objective 4.3.4. Consequently, in this document, it is always mentioned **software requirements** and not software safety requirements as the same requirement can be classified in many aspects: performance, security, interoperability, safety .. 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 be specified to ensure an acceptably safe operation have been identified and verified.

1.4. SOFTWARE SAFETY ASSESSMENT VALIDATION, VERIFICATION AND PROCESS ASSURANCE

Obj N°	ALs Objectives Title/Topic	OBJECTIVES	SWAL 1	SWAL 2	SWAL 3	SWAL 4	Output
3.4.1	Software Safety Assessment Validation	ANS SW Lifecycle V2.0 Part I-Chap 1 §3.4 Ensure that Software Requirements are complete and correct. Traceability, review and ***tracking*** of software safety requirements should be performed.	❍	❍	❍	❍	SSF Part III
3.4.2	Software Safety Assessment Verification	ANS SW Lifecycle V2.0 Part I-Chap 1 §3.4 Software Requirements should be consistent with the outcomes of the hazard effects and hazards description and classification and with Safety Objectives.	❍	❍	❍	❍	SSF Part III
3.4.3	Software Safety Assessment Process	ANS SW Lifecycle V2.0 Part I-Chap 1 §3.4 The performing of every step of the Software Safety Assessment should be ensured.	❍	❍	❍	❍	SSF Part III

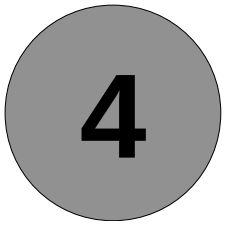
Obj N°	ALs Objectives Title/Topic	OBJECTIVES	SWAL				Output
			1	2	3	4	
	Assurance						

Note (Objective 3.4.1): tracking means: follow-up

1.5. 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	ANS SW Lifecycle V2.0 Part I-Chap 1 §3.5 The Software Safety Assessment process results should be documented.	❍	❍	❍	❍	SSF
3.5.2	Software Safety Assessment Documentation Configuration Management	ANS SW Lifecycle V2.0 Part I-Chap 1 §3.5 Software Safety Assessment documentation should be put under configuration management.	❍	❍	❍	❍	SSF
3.5.3	Software Safety Assessment Documentation Dissemination	ANS SW Lifecycle V2.0 Part I-Chap 1 §3.5 Software Safety Assessment documentation should be disseminated to interested parties.	❍	❍	❍	❍	SSF

Note: As the purpose of this task is to contribute to populate the Software Safety Folder, no particular part but the overall Software Safety Folder is concerned.



PRIMARY LIFE CYCLE PROCESSES

0 INTRODUCTION

The purpose of this chapter is to list objectives, per SWAL, that belong to primary life cycle processes.

Primary life cycle processes consist of:

1) Acquisition process:
    1) Initiation;
    2) FHA (Functional Hazard Assessment);
    3) PSSA (Preliminary System Safety Assessment);
    4) Request-for-Tender [Proposal];
    5) Contract preparation and update;
    6) Supplier monitoring;
    7) Acceptance and completion.

2) Supply process:
    1) Initiation;
    2) Preparation of response;

3) Contract;
4) Planning;
5) Execution and control;
6) Review and evaluation;
7) Delivery and completion.

3) Development process:
   1) System requirements analysis;
   2) System architectural design;
   3) Process implementation;
   4) Software requirements analysis;
   5) Software architectural design;
   6) Software detailed design;
   7) Software integration;
   8) Software installation;
   9) Standards/Rules Definition;
   10) Standards/Rules;
   11) Requirement Development Management;
   12) Use of Requirement Management Tool;
   13) Resource Management;
   14) Rationale for Design choices;
   15) Traceability;
   16) Transition criteria;
   17) Design tool;
   18) Use of a design tool;
   19) Code generation tool;
   20) Complexity constraints.

4) Operation process:
   1) Process implementation;
   2) Intended Operational Environment;
   3) User support;
   4) Software operation;
   5) Performance monitoring.

5) Maintenance process:
   1) Process implementation;
   2) SWAL allocation confirmation;
   3) SWAL satisfaction;;
   4) Software Migration;
   5) 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.

Note: as explained in Chapter 2, the organisation of the following lists into processes and objectives is coming from “ANS Software lifecycle” mainly based upon ISO/IEC 12207.

1 ACQUISITION PROCESS

Obj N°	ALs Objectives Title/Topic	OBJECTIVES	SWAL				Output
			1	2	3	4	
4.1.1	Initiation	ANS SW Lifecycle Part I-Chapter 2 § 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 will define and analyse the system requirements. 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/rules and procedures. The acquirer should prepare, document and execute an acquisition plan.	❍	❍	❍	❍	SSF Part I
4.1.2	Functional Hazard Assessment	ANS SW Lifecycle Part I-Chapter 2 §1 The acquirer should determine how safe the system needs to be. The acquirer should perform a FHA and thus should identify Safety Objectives for system hazards.	❍	❍	❍	❍	SSF Part II
4.1.3	Preliminary System Safety Assessment	ANS SW Lifecycle Part I-Chapter 2 §1 The acquirer should determine (during the System Design phase) whether the proposed architecture is expected to achieve the Safety Objectives defined by the FHA. Thus the Acquirer should specify Safety Requirements (including allocation of SWAL Software Assurance Level) for system elements.	❍	❍	❍	❍	SSF Part II
4.1.4	Request For Tender	ANS SW Lifecycle Part I-Chapter 2 §1 The acquirer should determine which processes, activities, and tasks of these recommendations are appropriate for the project and should tailor them accordingly.	❍	❍	❍	❍	
4.1.5	Contract preparation and update	ANS SW Lifecycle Part I-Chapter 2 §1 The acquirer should establish a procedure for supplier selection including proposal evaluation criteria and requirements compliance weighting.	❍	❍	❍	❍	
4.1.6	Supplier monitoring	ANS SW Lifecycle Part I-Chapter 2 §1 The acquirer will monitor the supplier's activities.	❍	❍	❍	❍	
4.1.7	Acceptance and completion	ANS SW Lifecycle Part I-Chapter 2 §1 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.	❍	❍	❍	❍	

2 SUPPLY PROCESS

Obj N°	ALs Objectives Title/Topic	OBJECTIVES	SWAL				Output
			1	2	3	4	
4.2.1	Initiation	ANS SW Lifecycle Part I-Chapter 2 § 2 The supplier conducts a review of requirements in the request for proposal taking into account organisational policies and other regulations.	❍	❍	❍	❍	
4.2.2	Preparation of response	ANS SW Lifecycle Part I-Chapter 2 § 2 The supplier should define and prepare a proposal in response to the request for proposal, including its recommended tailoring of any applied International Standard/rules.	❍	❍	❍	❍	
4.2.3	Contract	ANS SW Lifecycle Part I-Chapter 2 § 2 The supplier should negotiate and enter into a contract with the acquirer organisation to provide the software product or service.	❍	❍	❍	❍	
4.2.4	Planning	ANS SW Lifecycle Part I-Chapter 2 § 2 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 any applied International Standard/rules should be selected and mapped onto the lifecycle model. The supplier should develop and document project management plan(s).	❍	❍	❍	❍	
4.2.5	Execution & control	ANS SW Lifecycle Part I-Chapter 2 § 2 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.	❍	❍	❍	❍	
4.2.6	Review & evaluation	ANS SW Lifecycle Part I-Chapter 2 § 2 The supplier should co-ordinate contract review activities, interfaces, and communication with the acquirer's organisation. The supplier should perform quality assurance activities.	❍	❍	❍	❍	
4.2.7	Delivery & completion	ANS SW Lifecycle Part I-Chapter 2 § 2 The supplier should deliver and provide assistance to the acquirer in support of the delivered software product or service as specified in the contract.	❍	❍	❍	❍	

3 DEVELOPMENT PROCESS

System development process is not impacted by SWAL allocation, only Software development process is. Consequently, satisfying system-related objectives is a pre-requisite to software objectives satisfaction.

Obj N°	ALs Objectives Title/Topic	OBJECTIVES	SWAL				Output
			1	2	3	4	
4.3.1	System Requirements Analysis	ANS SW Lifecycle Part I Chapter 2 § 3.2 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 validation requirements.	❍	❍	❍	❍	
4.3.2	System Architectural Design	ANS SW Lifecycle Part I Chapter 2 §3.3 It should be ensured that all the system requirements are allocated among hardware, software, and manual-operations.	❍	❍	❍	❍	

Obj N°	ALs Objectives Title/Topic	OBJECTIVES	SWAL				Output
			1	2	3	4	

Obj N°	ALs Objectives Title/Topic	OBJECTIVES	SWAL 1	SWAL 2	SWAL 3	SWAL 4	Output
4.3.3	Process Implementation	ANS SW Lifecycle Part I Chapter 2 §3.1 Including: - end of activity/phase criteria for each activity/phase - joint technical review for each activity/phase The developer should define or select a software life cycle model appropriate to the scope, magnitude, and complexity of the project. The developer should select, tailor, and use those rules, methods, tools, and computer programming languages. The developer should develop plans for conducting the activities of the development process. Scope: life cycle definition, output documentation, output configuration management, SW products problems, environment definition, development plan, COTS	●	❍	❍	❍	SSF Part VII
4.3.4	SW requirements analysis	ANS SW Lifecycle Part I Chapter 2 §3.4 The developer should establish and document software requirements, using software requirements rules. The Software Requirements should as a minimum: • specify the functional behaviour of the ATM software, capacity, accuracy, timing performances, software resource usage on the target hardware, robustness to abnormal operating conditions, overload tolerance. • be complete and correct • comply with the System Requirements. Algorithms should be specified.	●	❍	❍	❍	SSF Part VII
4.3.5	SW architectural design	ANS SW Lifecycle Part I Chapter 2 §3.5 The developer should transform the requirements for the software item into an architecture that describes its top-level structure and identifies the software components. Scope: top level SW architecture definition, top level interfaces design, SW integration definition, SW architecture definition criteria	●	❍	❍		SSF Part VII
4.3.6	SW detailed design	ANS SW Lifecycle Part I Chapter 2 §3.6 The developer should develop a detailed design for each software component of the software item using software design rules. Scope: SW detailed design definition, interfaces design, SW Units tests definition.	●	❍			SSF Part VII
4.3.7	SW integration	ANS SW Lifecycle Part I Chapter 2 §3.8 The developer should develop an integration plan to integrate the software units and software components into the software item. The plan should include verification/test requirements, procedures, data responsibilities, and schedule. The plan should be documented. Scope: SW integration plan, SW integration definition, user documentation, SW validation preparation, SW integration evaluation (partially)	●	❍	❍		SSF Part VII
4.3.8	SW installation	ANS SW Lifecycle Part I Chapter 2 §3.11 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.	●	❍	❍	❍	SSF Part VII

Obj N°	ALs Objectives Title/Topic	OBJECTIVES	SWAL 1	SWAL 2	SWAL 3	SWAL 4	Output
4.3.9	Standards/rules definition	ANS SW Lifecycle Part I Chapter 2 §3. ***Development Plan*** The developer should develop plans for conducting the activities of the development process. The plans should include as a minimum: specific standards/rules, methods, tools, actions and responsibility associated with the development and validation of all requirements including safety (configuration management and justification of use of tools is SWAL dependent, see item 3, 8 to 10 of this table). If necessary, separate plans may be developed. These plans should be documented and executed.	●	❍	❍	❍	SSF Part VII SW Requirements rules, SW design rules, SW coding rules, Integration rules
4.3.10	Standards/rules	ANS SW Lifecycle Part I Chapter 2 §3. ***SW development plan (standards/rules)*** The developer should identify: - SW Requirements rules (See objective 4.3.4 for minimum content), - SW Design Rules, - SW Code Rules, Also, references to the standards/rules for previously developed software, including COTS software, if those standards/rules are different.	● ● ●	❍ ❍ ❍	❍ ❍	❍	SSF Part VII
4.3.11	Requirement development management	ANS SW Lifecycle Part I Chapter 2 §3.1.1 ***Software Development Environment*** The developer should identify the selected software development environment in terms of: (1) The chosen requirements development method(s), procedure(s) and tools (if any) to be used. (2) The hardware platforms for the tools (if any) to be used Ex: Method(s) are for example: SADT, SART, OOD…, though procedures are organisational ways of performing requirement management.	●	❍	❍	❍	SSF Part VII - identification of the SW development environment in the SW devpt plan and/or in the SW test plan
4.3.12	Use of a Requirement specification tool	ANS SW Lifecycle Part I Chapter 2 §3.4 A Requirement specification tool should be used. ***Remark:*** Requirement specification tool quality control should be performed	●	❍			SSF Part I

Obj N°	ALs Objectives Title/Topic	OBJECTIVES	SWAL 1	SWAL 2	SWAL 3	SWAL 4	Output
4.3.13	Resource management (load, memory, time response….)	ANS SW Lifecycle Part I Chapter 2 §3.5, 3.6 The developer should define a necessary margin for safety purpose, measure or verify that this margin is obtained. ***Remark:*** - Estimate is only requested - Measure or verification is requested. If many software share the same resources, then the margin should be evaluated at system level.	● ●	❍ ❍	❍	❍	SSF Part VII Estimate, measures or verification data
4.3.14	Rationale for design choices especially real time oriented one	ANS SW Lifecycle Part I Chapter 2 §3.6 The developer should ensure safety integrity of software components at design level. A set of principles should be identified concerning: - tasks and run- time aspects (priority, events, communications, ….) - interruptions (priorities, delay management, SW watchdog…) - treatment & propagation of errors (detection & recovering mechanisms, ….) - data management (protection & deadlock mechanisms, ….) - initialisation/ stop (exchange of data during these phases)		❍	❍		SSF Part VII *Included in safety SW design rules*
4.3.15	Traceability	ANS SW Lifecycle Part I Chapter 2 §3 The developer should perform traceability: a) Between System and Software requirements (ANS SW Lifecycle Part I Chapter 2 §3.4) b) Between Software requirements and Software design (Software component level, architectural design) (ANS SW Lifecycle Part I Chapter 2 §3.6) c) Between Software Architectural Design and Code (ANS SW Lifecycle Part I Chapter 2 §3.7) d) Between Code and Executable	● ● ● ●	❍ ❍ ❍	❍ ❍	❍	SSF Part VII Two matrices per link (both ways)
4.3.16	Transition criteria between life cycle phases	ANS SW Lifecycle Part I Chapter 2 §3.1.1. ***Devpt/SW life cycle*** ***Verification/ transition criteria*** The developer should describe the software life cycle processes to be used to form the specific software life cycle(s) to be used on the project, including the transition criteria for the software development processes. All essential information from a phase of the software lifecycle needed for the correct execution of the next phase should be available and verified. *See also evaluation criteria for Specification, design, code, test, integration* ***Remark*** These transition criteria must be defined according to the SWALs: - Transition criteria for all phases - Transition criteria only for Requirements Analysis and Verification phases.	● ●	❍ ❍	 ❍		SSF Part VII SW devpt plan and/or SW verification plan And/or SCM plan

Obj N°	ALs Objectives Title/Topic	OBJECTIVES	SWAL 1	SWAL 2	SWAL 3	SWAL 4	Output
4.3.17	Design tool	ANS SW Lifecycle Part I Chapter 2 §3.1.1 ***Software Development Environment*** If a design tool is used, then the developer should identify the selected software development environment in terms of: (1) The chosen design method(s), procedure(s) and tools (if any) to be used. (2) The hardware platforms for the tools (if any) to be used	●	❍	❍		SSF Part VII - identification of the SW development environment in the SW devpt plan and/or in the SW test plan, configuration management plan
4.3.18	Use of Design tool	ANS SW Lifecycle Part I Chapter 2 §3.1.1 A design tool should be used. ***Remark:*** Design tool quality control should be performed.	●				SSF Part I - identification of the SW development environment in the SW devpt plan and/or in the SW test plan, configuration management plan
4.3.19	Code generation Environment	ANS SW Lifecycle Part I Chapter 2 §3.7 ***Software Development Environment*** The developer should identify the selected software development environment in terms of: (1) The programming language(s), coding tools, compilers, linkage editors and loaders to be used, (2) The hardware platforms for the tools to be used ***Programming Languages*** ANS SW Lifecycle Part I Chapter 2 §3.7 Suitable programming languages should be selected for the required Assurance Level ***Compilers considerations*** ANS SW Lifecycle Part I Chapter 2 §3.7 Compilers mode of use (optimisations, limitations,….) should be defined and documented ***SW development tool validation*** ANS SW Lifecycle Part I Chapter 2 §3. The context for such a validation should be defined and documented *(Validation/certification of compilers/linkers/code generation tools)*	● ● ● ●	❍ ❍			SSF Part I - identification of the SW development environment in the SW devpt plan and/or in the SW test plan

Obj N°	ALs Objectives Title/Topic	OBJECTIVES	SWAL 1	SWAL 2	SWAL 3	SWAL 4	Output
4.3.20	Complexity constraints	ANS SW Lifecycle Part I Chapter 2 §3.1.1 A level of complexity (as well as selected criteria defining this complexity) must be defined and measured. If value exceeds thresholds (to be defined), a justification should be provided.	●	❍			SSF Part VII Data results in Safety & quality folder

Note: Qualification is not considered as to be developed in this document due to new institutional framework (involving Single Sky Committee, EASA and EUROCONTROL) that will address this topic. Consequently, qualification is considered as beyond the scope of this document and generally speaking beyond the scope of the EATM Safety Assessment Methodology (SAM).

4 OPERATION PROCESS

Operation process starts once transfer into operation starts, so when at least a first software release has been validated. Operation includes transfer into operation, commissioning and operation (decommissioning is covered by Problem Resolution and Maintenance Processes).

The means (activities and evidence) to satisfy Operation process objectives 4.4.1 and 4.4.5 will vary per SWAL due to:

- Operation of the software and the consequences of the operation (the rigour will increase as the consequences of software failure or malfunction increase);
- The need of performance monitoring data.

Operational process intends to address the system (people, procedure and equipment). Consequently, software is included in the equipment being used by operational staff according to operational procedures.

As described in this document (chapter 5 §6 VALIDATION), equipment validation is not covered in this §.

Some ATM procedures exist which rely on the software. The purpose of this chapter is not to define these ATM procedures, but to define how the software should be operated (HMI user's manual, mode of operation, …) when using these ATM procedures. These ATM procedures needs to be verified, however the ATM procedures verification and validation (Procedure Assurance Level is satisfied) is neither part of this chapter.

Procedure for raising problem reports and modification requests are covered by the Problem Resolution Process (Chapter 5 §8 of this document).

Obj N°	ALs Objectives Title/Topic	OBJECTIVES	SWAL				Output
			1	2	3	4	
4.4.1	Process implementation	ANS SW Lifecycle Part I Chapter 2 §4 An operation process should be developed, documented and executed. This process should at least include: • Procedure to operate the software in a specified environment and support users • Means to monitor the software performance especially vis-à-vis the SWAL.	●	❍	❍	❍	SSF Part VII
4.4.2	Intended Operational Environment	ANS SW Lifecycle Part I Chapter 2 §4 The software should be operated in its system intended environment according to the user documentation.	●	❍	❍	❍	
4.4.3	User support	ANS SW Lifecycle Part I Chapter 2 §4 The operator should provide assistance and consultation to the users as requested.	●	❍	❍	❍	
4.4.4	Software Operation	ANS SW Lifecycle Part I Chapter 2 §4 Procedures to operate the software should be defined, documented and executed.	●	❍	❍	❍	
4.4.5	Performance Monitoring	ANS SW Lifecycle Part I Chapter 2 §4 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

Maintenance process as defined in this document covers modification of software that has been commissioned so software that is into operation (not during software development). However, Maintenance process as described in this document does not cover the process which collects any kind of report or request for modification and agrees on the acceptance of the modification (See Chapter 5 §8 Problem Resolution Process).
The Problem Resolution Process triggers the Maintenance process.
Note: Maintenance does not cover modifications due to new requirements or change to existing requirements as this should lead to re-iterate the complete System Safety Assessment process.

Obj N°	ALs Objectives Title/Topic	OBJECTIVES	SWAL				Output
			1	2	3	4	
4.5.1	Process implementation	ANS SW Lifecycle Part I Chapter 2 §5 A maintenance process should be developed, documented and executed. This should include: - Procedure for receiving, recording and tracking modification requests (problem reports are managed in chapter 5 §8) - Providing feedback to the originators of modification requests. This software maintenance intervention should be performed in accordance with the System Safety Assessment (SSA) part of the EATM Safety Assessment Methodology (SAM) that provides guidelines on how to perform a maintenance intervention risk assessment. The maintainer should tailor this procedure to software maintenance intervention if anything is specific.	●	❍	❍	❍	SSF Part VII
4.5.2	SWAL allocation confirmation	ANS SW Lifecycle Part I Chapter 2 §5 First the impact on safety of the problem or modification as provided by the "Problem Resolution Process" should be confirmed throughout the maintenance process.	●	❍	❍	❍	SSF Part V
4.5.3	SWAL satisfaction	ANS SW Lifecycle Part I Chapter 2 §5 The maintainer should ensure that any maintenance activity does not impair the confidence that (new or old confirmed) SWAL is satisfied. This means that SWAL objectives allocated for this SWAL to all the other processes are still satisfied.	●	❍	❍	❍	SSF Part VII
4.5.4	Software migration	ANS SW Lifecycle Part I Chapter 2 §5 The maintainer should define a procedure to migrate the modified software and put it into operation. Some criteria could be listed to customise the procedure according to the SWAL.	●	❍	❍	❍	SSF Part V
4.5.5	Software Decommissioning	ANS SW Lifecycle Part I Chapter 2 §5 A decommissioning plan to remove active support by the operation and maintenance organisations should be developed and documented. An impact analysis should be performed.	●	❍	❍	❍	SSF Part V

This page is intentionally left blank.

SUPPORTING LIFE CYCLE PROCESSES

0 INTRODUCTION

The purpose of this chapter is to list objectives, per SWAL, that belong to supporting life cycle processes.

Supporting life cycle processes consist of:

1) Documentation process:
    1) Process implementation;
    2) Design and development;
    3) Production;
    4) Maintenance.
2) Configuration management process:
    1) Process implementation;
    2) Configuration identification;
    3) Configuration control;
    4) Configuration status accounting;

5) Configuration evaluation;
6) Retrieval & Release process
7) Use of tool;
8) Acquirer agreement for the use of a tool;
9) Configuration Management at the level of Software Component;
10) Configuration Management Traceability.

3) Quality assurance process:

1) Process implementation;
2) Product assurance;
3) Process assurance;
4) Quality audits.

4) Verification process:

1) System verification;
2) Verification plan;
3) Software requirements;
4) Integration;
5) Software Design;
6) Code;
7) Independent verification;
8) Executable;
9) Data;
10) Traceability;
11) Complexity measures;
12) Verification process results;
13) Retrieval & Release process.

5) Validation process: (Not Applicable to SW as validation is system-related)

1) Process implementation;
2) Validation planning;
3) Boundaries validation;
4) Pass/Fail criteria;
5) Validation test;
6) Record of validation activities;
7) Independent validation team.

6) Joint review process:

1) Process implementation;
2) Project management reviews;
3) Technical reviews.

7) Audit process:

1) Process implementation;
2) Audit.

8) Problem resolution process:

1) Process implementation;
2) Problem resolution;
3) Safety impact;
4) Problem Report Configuration Management.

Note: as explained in Chapter 2, the organisation of the following lists into processes and objectives is coming from “ANS Software lifecycle” mainly based upon ISO/IEC 12207.

1 DOCUMENTATION PROCESS

Obj N°	ALs Objectives Title/Topic	OBJECTIVES	SWAL 1	SWAL 2	SWAL 3	SWAL 4	Output
5.1.1	Process Implementation	ANS SW Lifecycle Part I Chapter 3 §1 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).	●	❍	❍	❍	SSF Part III
5.1.2	Design & Development	ANS SW Lifecycle Part I Chapter 3 §1 Each identified document should be designed in accordance with applicable documentation standards/rules for format, content description, page numbering, figure/table placement, proprietary/security marking, packaging, and other presentation items.	●	❍	❍		SSF Part III
5.1.3	Production	ANS SW Lifecycle Part I Chapter 3 §1 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.	●				SSF Part III
5.1.4	Maintenance	ANS SW Lifecycle Part I Chapter 3 §1 The tasks, that are required to be performed when documentation is to be modified, should be performed.	●	❍			SSF Part III

2 CONFIGURATION MANAGEMENT PROCESS

Obj N°	ALs Objectives Title/Topic	OBJECTIVES	SWAL 1	SWAL 2	SWAL 3	SWAL 4	Output
5.2.1	Configuration management process	***Configuration management process*** ANS SW Lifecycle Part I Chapter 3 §2 *Process implementation* A configuration management plan should be developed. The plan should describe: - the configuration management activities; - procedures and schedule for performing these activities; - the organisation(s) responsible for performing these activities; and their relationship with other organisations, such as software development or maintenance;	●	❍	❍	❍	SSF Part VII

Obj N°	ALs Objectives Title/Topic	OBJECTIVES	SWAL				Output
			1	2	3	4	
		- Software life cycle environment control management (tools used to develop or verify SW) - Definition of SW life cycle data (any output) control management (identify for each output which kind of Configuration Management to set-up). The plan should be documented and implemented.					
5.2.2	Configuration identification	ANS SW Lifecycle Part I Chapter 3 §2 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; • the problem reports list (those already fixed, those fixed in that particular version and those still open if any); • and other identification details. The items to be configuration-identified should be drawn with its associated configuration management level.	●	❍	❍	❍	SSF Part VII
5.2.3	Configuration control	ANS SW Lifecycle Part I Chapter 3 §2 The following should be performed: identification and recording of change requests; 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.	●	❍	❍	❍	SSF Part V
5.2.4	Configuration status accounting	ANS SW Lifecycle Part I Chapter 3 §2 Management records and status reports that show the status and history of controlled software items including baseline should be prepared. Status reports should include the number of changes for a project, latest software item versions, release identifiers, the number of releases, and comparisons of releases.	●	❍	❍	❍	SSF Part V & VII
5.2.5	Configuration evaluation	ANS SW Lifecycle Part I Chapter 3 §2 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).	●	❍	❍	❍	SSF Part VII
5.2.6	Retrieval & Release Process	ANS SW Lifecycle Part I Chapter 3 §2 A retrieval and release process should exist and should be documented. 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	●	❍	❍	❍	SSF Part VII

Obj N°	ALs Objectives Title/Topic	OBJECTIVES	SWAL				Output
			1	2	3	4	
		delivered in accordance with the policies of the organisations involved.					
5.2.7	Use of tool	A tool should be used to perform Software items configuration management.	●	❍	❍	❍	SSF Part I
5.2.8	Use of tool (acquirer agreement)	The acquirer should accept the selected software items configuration management tool.	●	❍			SSF Part VII
5.2.9	At level of SW component	The software items configuration management should be performed at the software component level.	●	❍	❍		SSF Part VII
5.2.10	Configuration Management Traceability	Software life cycle data (any output) should be traceable between versions. Besides, at the equipment level, configuration management should trace software and hardware versions to ensure that compatibility is achieved.					

3 QUALITY ASSURANCE PROCESS

Obj N°	ALs Objectives Title/Topic	OBJECTIVES	SWAL				Output
			1	2	3	4	
5.3.1	Process Implementation	***Quality assurance process*** ANS SW Lifecycle Part I Chapter 3 §3 *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.	●	❍	❍	❍	SSF Part VII
5.3.2	Product Assurance	*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 documentation comply with the contract and adhere to the plans.	●	❍	❍	❍	SSF Part VII

Obj N°	ALs Objectives Title/Topic	OBJECTIVES	SWAL 1	SWAL 2	SWAL 3	SWAL 4	Output
		A Software Conformity review should be performed.					
5.3.3	Process Assurance	*Process assurance* It should be assured that those software life cycle 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.	●	❍	❍	❍	SSF Part VII

4 VERIFICATION PROCESS

Note: The activity of this paragraph consists in providing assurance that such a level of verification has been done (and not only doing this level of verification).

4.1 SYSTEM VERIFICATION

Obj N°	ALs Objectives Title/Topic	OBJECTIVES	SWAL 1	SWAL 2	SWAL 3	SWAL 4	Output
5.4.1	verification of system requirements	***Requirements verification*** ANS SW Lifecycle Part I Chapter 3 §4 The requirements should be verified considering the criteria listed below: a) The system requirements are complete and correct. b) The system requirements are consistent, feasible, and verifiable.	●	❍	❍	❍	System verification results

4.2 SOFTWARE VERIFICATION

	ALs Objectives Title/Topic	OBJECTIVES	SWAL				Output
			1	2	3	4	
5.4.2	Verification plan	***Verification plan*** ANS SW Lifecycle Part I Chapter 3 §4.1 A verification plan should be developed and documented. The plan should address the life cycle verification activities and phase outputs subject to verification and related resources, responsibilities, and schedule. The plan should address procedures for forwarding verification reports to the acquirer and other involved organisations.	●	❍	❍	❍	SSF Part VII (set of) verification plan(s)
5.4.3	Verification of software requirements	***Requirements verification*** ANS SW Lifecycle Part I Chapter 3 §4.2 The software requirements should be verified considering the criteria listed below: a) The software requirements are complete and correct; b) The functional behaviour of the Software complies with the Software Requirements; c) The timing performances of the software complies with the Software Requirements; d) The software requirements are consistent, feasible, and verifiable; e) The software robustness to abnormal conditions complies the Software Requirements (Only SWAL1&2&3) f) External consistency with the system requirements; g) Internal consistency between software requirements; h) Verification coverage of the requirements of the software item; (Only for SWAL1 & 2) i) No conflict exist between software requirements and the HW/SW features of the target computer (system response time, Input/output HW, software resource usage on the target computer) (Only SWAL1 & 2) ; j) Software requirements conform to Software requirements rules (Only for SWAL 1 & 2 & 3) ; k) Algorithms are accurate and correct (Only SWAL 1 & 2 & 3); l) The capacity of the Software complies with the Software Requirements (Only SWAL 1 & 2 & 3); m) The overload tolerance of the Software complies with the Software Requirements (Only SWAL 1 & 2 & 3).	●	❍	❍	❍	SSF Part VII SW verification results
5.4.4	Integration Verification	***Integration verification*** ANS SW Lifecycle Part I Chapter 3 §4.2 The integration should be verified considering the criteria listed below: a) the software components have been completely and correctly integrated into the software item b) the software units have been completely and correctly integrated into the software component c) the hardware items, software items, and manual operations of the system have been completely and correctly integrated into the system. d) the integration tasks have been performed in accordance with an integration plan. ***Examples of verification criteria are (especially as far as isolation between software is concerned)*** - Linking and loading data and memory map - Data control and coupling - Incorrect HW addresses - Memory overlaps	● ● ● ●	❍ ❍ ❍ ❍	❍ ❍ ❍	 ❍ ❍	SSF Part VII Integration verification results (tests results, reviews records….)

	ALs Objectives Title/Topic	OBJECTIVES	SWAL				Output
			1	2	3	4	
		- Missing SW components. ***Remark:*** Global verification should be performed either through tests or other methods like reviews…..					
5.4.5	Verification of software design	***Design Verification*** ANS SW Lifecycle Part I Chapter 3 §4.2 The developer should evaluate the design tests, test results, and user documentation considering the criteria listed below. a) External consistency with the software requirements; b) Internal consistency (data flow and control flow); c) Verification coverage of the software architectural design; d) Design conforms to Design rules e) Appropriateness of test rules and methods used; f) Conformance to expected results; g) Feasibility of software design testing; h) Feasibility of maintenance; i) Verification criteria on which verification completion will be judged. The results of the evaluations should be documented. ***Remark:*** The compliance should be verified according to the definition of the transition criteria between life cycle phases (cf AL allocation for Development process)	●	❍	❍		SSF Part VII Design verification results + Integration test description verification results
5.4.6	Verification of code	***Software Units Code & Verification Results Evaluation Criteria*** ANS SW Lifecycle Part I Chapter 3 §4.2 The developer should evaluate software code and verification results considering the criteria listed below. a) External consistency with the requirements and design of the software item; b) Internal consistency between unit requirements; c) Verification coverage of units; d) Code conforms to Code rules; e) Appropriateness of coding methods and rules used; f) Feasibility of software code verification; g) Feasibility of maintenance. The results of the evaluations should be documented. ***Remark:*** Global verification should be performed either through tests or other methods like reviews or other means…..	●	❍			SSF Part VII SW unit code & test description verification results (tests or reviews…)
5.4.7	Independent verification	"Independence" means : - independent team for system verification - independence at people level for SW	●	❍			SSF Part VII
5.4.8	Verification of executable	***Executable verification*** ANS SW Lifecycle Part I Chapter 3 §4.2 The developer should evaluate executable and verification results considering the criteria listed below. a) External consistency with the code of the software item (e.g. is the compiler generating an appropriate exe?); b) Internal consistency between exe requirements (e.g.: is the compiler always generating the same exe for the same source?);	●				SSF Part VII SW source code

	ALs Objectives Title/Topic	OBJECTIVES	SWAL				Output
			1	2	3	4	
		c) Verification coverage of executable (e.g. is the compiler generating additional and unnecessary executable such as dead executable code?); d) Feasibility of executable verification; The results of the evaluations should be documented.					verification results (inspection & test)
5.4.9	Data verification	The data structures specified during design should be verified for: ANS SW Lifecycle Part I Chapter 3 §4.2 - completeness - self-consistency - protection against alteration or corruption	●	❍			SSF Part VII Data verification results (inspection & test)
5.4.10	Traceability	Traceability should be verified: a) Between System and Software requirements b) Between Software requirements and Software design (Software component level, architectural design) c) Between Software Architectural Design and Code d) Between Software Code and Executable	● ● ● ●	❍ ❍ ❍	❍ ❍	❍	SSF Part VII
5.4.11	Complexity measures	Verification of the folder related to complexity measures: - measures analysis, - performed actions .	●	❍			SSF Part VII
5.4.12	Verification of Verification process results	ANS SW Lifecycle Part I Chapter 3 & 4.2 Verification cases, procedures and results should be verified, so that: – Verification procedures are correct and complete – Verification results are correct and complete and discrepancies are explained – Verification of the software requirements verification cases , procedures and results is correct and complete – Verification of the software design verification cases, procedures and results is correct and complete – Verification of the software code verification cases, procedures and results is correct and complete – Verification of the software executable verification cases, procedures and results is correct and complete (Only for SWAL 1) – Verification of the software integration verification cases, procedures and results is correct and complete – Verification of the software data verification cases, procedures and results is correct and complete – Verification of the traceability verification procedures and results is correct and complete	●	❍			SSF Part VII

Note: 3 types of verification exist:
a) developing software;
b) verifying that software is “correct” (satisfactorily developed);
c) verifying that verification is “correct”.

So in that §4.2, verification of verification objectives are of type c). All the other verification objectives are of type b).

4.3 OTHER VERIFICATIONS

Obj N°	ALs Objectives Title/Topic	OBJECTIVES	SWAL 1	SWAL 2	SWAL 3	SWAL 4	Output
5.4.13	verification of Retrieval and Release process	***Retrieval & Release process verification*** ANS SW Lifecycle Part I Chapter 3 §2 The Software Retrieval and release process should be verified.	●	❍	❍		SSF Part VII

5 VALIDATION PROCESS

5.1 EQUIPMENT VALIDATION

The ANSP is responsible for conducting validation (which intends to show that the system meets its safety objectives in its operational environment). This section identifies how SWAL contribute to give confidence that safety objectives are met. This validation concerns the equipment part of the system, as some procedures are defined to perform this equipment validation. However, this equipment validation does not intend to validate the procedures. (Cf: Chapter 1 § 3).

Outputs of Validation process are not part of the Software Safety folder and validation objectives are to be satisfied whatever the SWAL as validation is at system level and not at Software assurance level.
N/A: Not Applicable

Obj N°	ALs Objectives	OBJECTIVES	SWAL	Output

	Title/Topic		1	2	3	4	
5.5.1	Validation	***Process implementation*** ANS SW Lifecycle Part I Chapter 3 §5 A validation process should be established to validate the system or software product in its operational environment. Validation tasks, including associated methods, techniques, and tools for performing the tasks**,** should be selected. A validation plan should be developed, documented and implemented.	N/A	N/A	N/A	N/A	Validation plan
5.5.2	Validation planning	***System Description*** ANS SW Lifecycle Part I Chapter 1 §1 The Software purpose should be defined. Operational scenarios should be defined. The Software/System functions and their relationships should be defined. Software external interfaces should be described. ***Validation*** ANS SW Lifecycle Part I Chapter 3 §5 *Task list includes:* ANSP prepares 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.	N/A	N/A	N/A	N/A	Operational scenario description; system design document; interface description Validation test description and results (including appropriate ness of testing environment for “intended use”
5.5.3	Boundaries Validation	***System Description*** ANS SW Lifecycle Part I Chapter 3 §5 Software boundaries should be validated (performance, operational, time, ..)	N/A	N/A	N/A	N/A	Validation test description and results
5.5.4	Pass/ fail criteria	Validation tests pass/fail criteria should be defined.	N/A	N/A	N/A	N/A	

Obj N°	ALs Objectives Title/Topic	OBJECTIVES	SWAL				Output
			1	2	3	4	
5.5.5	Validation test	***System Integration*** The software configuration items should be integrated, with hardware configuration items, manual operations, and other system elements 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 validation requirement of the system**,** a set of tests, test cases (inputs, outputs, test criteria) and test procedures for conducting System Validation Testing should be developed and documented. ***System Validation Evaluation Criteria*** The system should be validated considering the criteria listed below. The results of validation should be documented. a) Test (or equivalent means) Coverage of system requirements; b) Conformance to expected performance; c) Feasibility of operation (are system requirements correct ?) and maintenance. Validation results should comply with pass criteria to be accepted.	N/A	N/A	N/A	N/A	
5.5.6	Record of validation activities/ results	***Closure (cf management process)*** When all system elements, 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 system elements, activities, and tasks employed for completeness. These results and records should be archived in a suitable environment as specified in the contract.	N/A	N/A	N/A	N/A	Acceptance test results
5.5.7	Independent validation team	Validation process *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*	N/A	N/A	N/A	N/A	Development plans

6 JOINT REVIEW PROCESS

Obj N°	ALs Objectives Title/Topic	OBJECTIVES	SWAL				Output
			1	2	3	4	
5.6.1	Process implementation	ANS SW Lifecycle Part I Chapter 3 §6 Periodic reviews should be held at predetermined milestones as specified in the project plan(s). The review results should be documented and distributed.	●	❍	❍	❍	SSF Part VII
5.6.2	Project management reviews	ANS SW Lifecycle Part I Chapter 3 §6 Project status should be evaluated relative to the applicable project plans, schedules, standards/rules, transition criteria and guidelines.	●	❍	❍	❍	SSF Part VII
5.6.3	Technical reviews	ANS SW Lifecycle Part I Chapter 3 §6 Technical reviews should be held to evaluate the software products or services under consideration.	●	❍	❍	❍	SSF Part VII

7 AUDIT PROCESS

Obj N°	ALs Objectives Title/Topic	OBJECTIVES	SWAL				Output
			1	2	3	4	
5.7.1	Process implementation	ANS SW Lifecycle Part I Chapter 3 §7 Audits should be held at predetermined milestones as specified in the project plan(s) or upon specific request. After completing an audit, the audit results should be documented and provided to the audited party.	●	❍	❍	❍	SSF Part VII
5.7.2	audits at SW requirement level	ANS SW Lifecycle Part I Chapter 3 §7 Audits may be conducted at predetermined milestones to ensure that: - The acceptance review and verification requirements prescribed by the documentation are adequate for the acceptance of the software. - Verification data comply with the specification. - Software was successfully verified and meets its SW requirements. - Verification reports are correct and discrepancies between actual and expected results have been resolved. - Product (SW requirement) and User documentation complies with rules as specified. - Activities have been conducted according to applicable requirements, plans, and contract. - The costs and schedules adhere to the established plans. For SWAL4 it can decided that audits are not necessary.	●	❍	❍	❍	SSF Part VII

Obj N°	ALs Objectives Title/Topic	OBJECTIVES	SWAL				Output
			1	2	3	4	
5.7.3	audits down to SW design level	ANS SW Lifecycle Part I Chapter 3 §7 Audits should be conducted at predetermined milestones or upon specific request to ensure that: - The acceptance review and verification requirements prescribed by the documentation are adequate for the acceptance of the software. - Verification data comply with the specification. - Software was successfully verified and meets its SW requirements and SW architecture requirements. - Verification reports are correct and discrepancies between actual and expected results have been resolved. - Product (SW requirement and SW architecture) and User documentation complies with rules as specified. - Activities have been conducted according to applicable requirements, plans, and contract. - The costs and schedules adhere to the established plans.	●	❍	❍		SSF Part VII
5.7.4	quality audits down to source code level	ANS SW Lifecycle Part I Chapter 3 §7 Audits should be conducted at predetermined milestones and upon specific request to ensure that: - The acceptance review and verification requirements prescribed by the documentation are adequate for the acceptance of the software. - Verification data comply with the specification. - Software was successfully verified and meets its SW requirements, SW architectural requirement and SW detailed design requirements. - Verification reports are correct and discrepancies between actual and expected results have been resolved. - Product (SW requirement, SW architecture, SW detailed design and Source code) and User documentation complies with rules as specified. - Activities have been conducted according to applicable requirements, plans, and contract. - The costs and schedules adhere to the established plans.	●	❍			SSF Part VII
5.7.5	quality audits down to executable level	ANS SW Lifecycle Part I Chapter 3 §7 Audits should be conducted at predetermined milestones or upon specific request to ensure that: - The acceptance review and verification requirements prescribed by the documentation are adequate for the acceptance of the software. - Verification data comply with the specification. - Software was successfully verified and meets its SW requirements, SW architectural requirement and SW detailed design requirements. - Verification reports are correct and discrepancies between actual and expected results have been resolved. - Product (SW requirement, SW architecture, SW detailed design, Source code and executable) and User documentation complies with rules as specified. - SW development tools (e.g. Compilers) are qualified. - Activities have been conducted according to applicable requirements, plans, and contract. - The costs and schedules adhere to the established plans.	●				SSF Part VII

8 PROBLEM/CHANGE RESOLUTION PROCESS

The Problem/Change 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.

So Problem/Change Resolution phase includes:

- Non-conformance reports (during all phases) which could or not affect safety
- Correction: a modification has to be performed in order to fix a reported problem
- Prevention: a modification has to be performed because an analysis has concluded that the software behaviour could contribute to a safety-related event:
  - either because the system safety assessment (at the system or at the software level) process review/update identified it and did not do so till then.
  - Or because an operational report identified it though no safety occurrence happened.
- Evolution: a modification has to be performed because software has to be updated for technological reasons (change of hardware platform, software development tool version change, software development tool obsolescence, ..)
- Improvement: a modification has to be performed because the software performance, though compliant with requirements has to be improved.

Obj N°	ALs Objectives Title/Topic	OBJECTIVES	SWAL				Output
			1	2	3	4	
5.8.1	Process implementation	***Problem resolution process*** ANS SW Lifecycle Part I Chapter 3 §8 *Process implementation* A problem resolution process should be established and documented for handling all problems (including non-conformances) detected in the software products and processes/activities.	●	❍	❍	❍	SSF Part V
5.8.2	Problem resolution	*Problem resolution* ANS SW Lifecycle Part I Chapter 3 §8 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.	●	❍	❍	❍	SSF Part V
5.8.3	Safety Impact	An analysis should be performed to: • assess if a problem report has a safety impact (FHA and/or PSSA and/or SSA) • corrective actions exist such that safety-related problems can be shown to have been tolerably mitigated.	●	❍	❍	❍	SSF Part V
5.8.4	Problem Report Configuration Management	Part of Objective 5.2.1 which defines the lifecycle data to be put under configuration management and which level of configuration management to be set for this kind of data.	●	❍	❍	❍	SSF Part V

Page intentionally left blank.

ORGANISATIONAL LIFE CYCLE PROCESSES

0 INTRODUCTION

The purpose of this chapter is to list objectives, per SWAL, that belong to organisational life cycle processes.

Organisational life cycle processes consist of:

1) Management process:
    1) Initiation and scope definition;
    2) Planning;
    3) Execution and control;
    4) Review and evaluation;
    5) Closure.
2) Infrastructure process:
    1) Process implementation;
    2) Establishment of the infrastructure;
    3) Maintenance of the infrastructure.

3) Improvement process:

    1) Process establishment;
    2) Process assessment;
    3) Process improvement.

4) Training process:

    1) Process implementation;
    2) Training material development;
    3) Training plan implementation.

Note: as explained in Chapter 2, the organisation of the following lists into processes and objectives is coming from “ANS Software lifecycle” mainly based upon ISO/IEC 12207.

1 MANAGEMENT PROCESS

Obj N°	ALs Objectives Title/Topic	OBJECTIVES	SWAL 1	SWAL 2	SWAL 3	SWAL 4	Output
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.	❍	❍	❍	❍	SSF Part III
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; - 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.	❍	❍	❍	❍	SSF Part III
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.	❍	❍	❍	❍	SSF Part III
6.1.4	Review & evaluation	The manager should ensure that the software products and plans are evaluated for satisfaction of requirements. The manager should assess the evaluation results of the software products, activities, and tasks completed during the execution of the process vis-à-vis the achievement of the objectives and completion of the plans.	❍	❍	❍	❍	SSF Part III
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.	❍	❍	❍	❍	SSF Part III

2 INFRASTRUCTURE PROCESS

Obj N°	ALs Objectives Title/Topic	OBJECTIVES	SWAL 1	SWAL 2	SWAL 3	SWAL 4	Output
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 this process, considering the applicable procedures, standards/rules, tools, and techniques. The establishment of the infrastructure should be planned and documented.	❍	❍	❍	❍	SSF Part I
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.	❍	❍	❍	❍	SSF Part I
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 this process. As part of maintaining the infrastructure, the extent to which the infrastructure is under configuration management should be defined.	❍	❍	❍	❍	SSF Part VII

3 IMPROVEMENT PROCESS

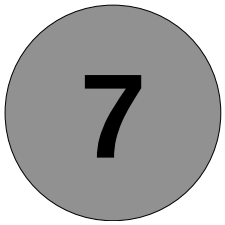
Obj N°	ALs Objectives Title/Topic	OBJECTIVES	SWAL 1	SWAL 2	SWAL 3	SWAL 4	Output
6.3.1	Process implementation	The organisation should establish a suite of organisational processes for all software lifecycle processes as they apply to its business activities. The processes and their application to specific 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).	❍	❍	❍	❍	SSF Part III or VII
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.	❍	❍	❍	❍	SSF Part III or VII
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.	❍	❍	❍	❍	SSF Part III or VII

4 TRAINING PROCESS

In this paragraph, Training does not address training of operational staff in charge of operating software, but training of staff in charge of developing software.

Obj N°	ALs Objectives Title/Topic	OBJECTIVES	SWAL				Output
			1	2	3	4	
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.	❍	❍	❍	❍	SSF Part VII
6.4.2	Training material development	Training manuals, including presentation materials used in providing training, should be developed.	❍	❍	❍	❍	SSF Part VII
6.4.3	Training plan implementation	The training plan should be implemented to provide training to personnel. Training records should be maintained.	❍	❍	❍	❍	SSF Part VII

Page intentionally left blank.



ADDITIONAL ANS SOFTWARE LIFECYCLE OBJECTIVES

0 INTRODUCTION

The purpose of this chapter is to list objectives, per SWAL, that do not belong to ISO/IEC 12207, but have been added due to:

- The analysis of other standards more safety oriented (ED109/DO278, DO 178B/ED 12B and IEC 61508),
- ATM particularities (some are included in ED109/DO278),
- Omissions by existing standards.

These additional life cycle processes consist of:

1) Software development Environment
    1) Definition

2) Programming language
3) Compiler considerations

2) COTS
1) COTS plans;
2) COTS Transition criteria;
3) COT Plan consistency;
4) COTS requirement coverage;
5) COTS Lifecycle data;
6) COTS Derived requirements;
7) COTS HW compatibility;
8) COTS Configuration Management;
9) COTS Problem Reporting;
10) COTS Incorporation;
11) COTS Configuration Management Archiving;

12) Tool qualification (Out of scope of these recommendations)
1) Qualification criteria for software development tools
2) Qualification criteria for software verification tools

4) Service Experience

1 SOFTWARE DEVELOPMENT ENVIRONMENT

Obj N°	ALs Objectives Title/Topic	OBJECTIVES	SWAL				Output
			1	2	3	4	
7.1.1	Definition	A suitable set of development tools should be selected for the allocated Assurance Level.	❍	❍	❍	❍	SSF Part I
7.1.2	Programming Languages	Suitable programming languages should be selected for the allocated Assurance Level.	❍	❍	❍	❍	SSF Part VII
7.1.3	Compiler Considerations	Compilers features (optimisations, limitations, ..) should be defined.	❍	❍	❍	❍	SSF Part I

2 COTS

COTS definition (ANS SW Lifecycle Part I chapter 5 § 3.1)

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.

Obj N°	ALs Objectives Title/Topic	OBJECTIVES	SWAL				Output
			1	2	3	4	
7.2.1	COTS Plans	Acquisition, verification, configuration management, quality assurance plans are defined	N/A	❍	❍	❍	SSF Part VII
7.2.2	COTS Transition Criteria	Transition criteria are defined (according to the relationships between COTS processes and appropriate CNS/ ATM lifecycle processes): only for AL2 & AL3	N/A	❍	❍		SSF Part VII
7.2.3	COTS Plans Consistency	COTS plans are consistent with ANS SW plans (plans for acquisition, evaluation, integration …processes are consistent with ANS SW plans): only for AL2 & AL3	N/A	❍	❍		SSF Part VII
7.2.4	COTS Requirements Coverage	ANS SW requirements coverage achieved by the COTS is determined	N/A	❍	❍	❍	SSF Part VII
7.2.5	COTS Lifecycle data	Life cycle data availability is determined in accordance with SWAL (extent of life cycle data that are available for assurance purposes)	N/A	❍	❍	❍	SSF Part VII

Obj N°	ALs Objectives Title/Topic	OBJECTIVES	SWAL				Output
			1	2	3	4	
7.2.6	COTS Derived Requirements	Derived requirements are defined (requirements imposed on the ANS system due to the usage of COTS or requirements to prevent the unneeded functions of the COTS from affecting the ANS system)	N/A	❍	❍	❍	SSF Part VII
7.2.7	COTS HW Compatibility	Compatibility of COTS with target computers is determined	N/A	❍	❍	❍	SSF Part VII
7.2.8	COTS Configuration Management: Identification	COTS configuration and data items are identified.	N/A	❍	❍	❍	SSF Part VII
7.2.9	COTS Problem Reporting	COTS problem reporting is established.	N/A	❍	❍	❍	SSF Part VII
7.2.10	COTS Incorporation	Incorporation of COTS release is controlled.	N/A	❍	❍	❍	SSF Part VII
7.2.11	COTS Configuration Management: Archiving	COTS configuration and data items are archived.	N/A	❍	❍	❍	SSF Part VII

Note: COTS (as defined here above and more extensively in ANS SW Lifecycle Part I chapter 5 § 3.1) usage is not accepted for software having to satisfy a SWAL1.

3 TOOL QUALIFICATION

Qualification of a tool is needed when processes of these recommended guidelines are eliminated, reduced or automated by the use of a software tool without its output being verified as specified in Verification Process. The use of software tools to automate activities of the software life cycle processes can help satisfy system safety objectives insofar as they can enforce conformance with software development standards and use automatic checks.

The objective of the tool qualification process is to ensure that the tool provides confidence at least equivalent to that of the process(es) eliminated, reduced or automated.

If partitioning of tool functions can be demonstrated, only those functions that are used to eliminate, reduce, or automate software life cycle process activities, and whose outputs are not verified, need be qualified.

Only deterministic tools may be qualified, that is, tools which produce the same output for the same input data when operating in the same environment. The tool qualification process may be applied either to a single tool or to a collection of tools.

Software tools can be classified as one of two types:

- Software development tools: Tools whose output is part of product software and thus can introduce errors. For example, a tool, which generates Source Code directly from requirements, would have to be qualified if the generated Source Code is not verified as specified in Verification Process.

- Software verification tools: Tools that cannot introduce errors, but may fail to detect them. For example, a static analyser, that automates a software verification process activity, should be qualified if the function that it performs is not verified by another activity. Type checkers, analysis tools and test tools are other examples.

However, tool qualification is no more considered as to be developed in this document due to new framework (involving EASA and EUROCONTROL Regulatory Committee) that will address this topic, which has institutional aspects. Consequently, tool qualification is considered as beyond the scope of this document and generally speaking beyond the scope of the EATMP Safety Assessment Methodology.

4. SERVICE EXPERIENCE

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, which make it informative and not normative. However, putting this text as informative in ED109 was the result of a consensus with airworthiness experts. EATMP Software Task Force 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:

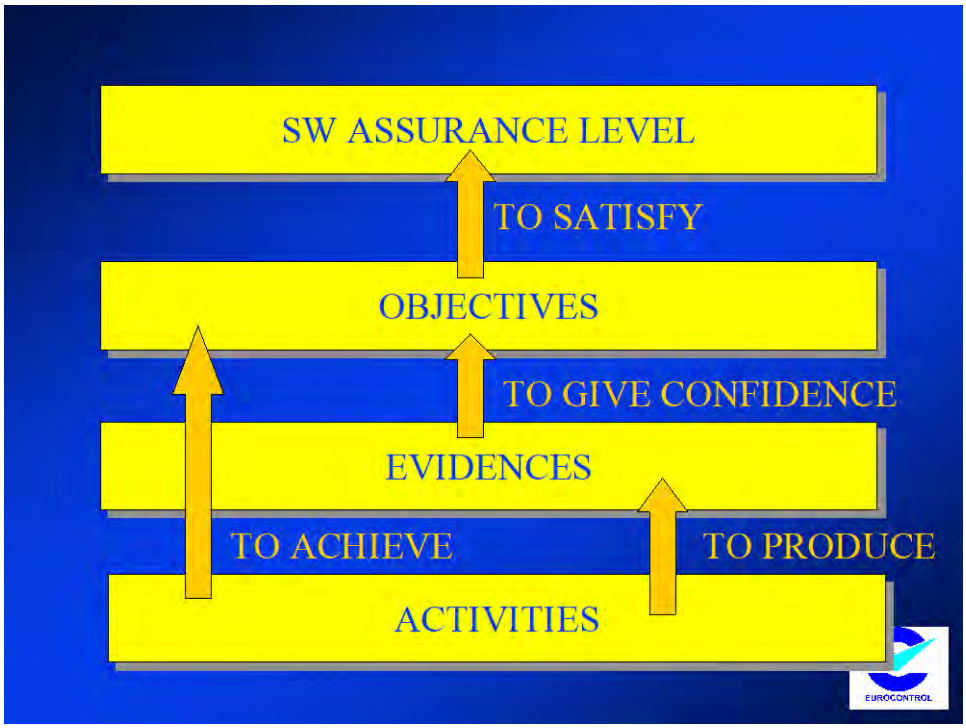
(1) Cannot be applied for SWAL1.
(2) A minimum of one year (8,760 hours) of service experience with no failure for SWAL2.
(3) A minimum of six months (4,380 hours) of service experience with no failure for SWAL3.
(4) SWAL4 objectives are typically satisfied without a need for alternate methods.

8

SOFTWARE SAFETY FOLDER

0 INTRODUCTION

The purpose of this chapter is to propose a structure for the documents and evidences that intend to provide assurance that the objectives to satisfy a SWAL are achieved.



1 SOFTWARE SAFETY FOLDER STRUCTURE

Note: The “Software Manual” (which is part of the Safety Management System) is not part of the Software Safety folder as it is not software dependent.

A Software Safety Folder is dedicated to one and only one software.

However as many software could be part of a system and share some common items (development tools, system description …), the content of the software safety folder can be restricted to a reference.

As the purpose of the Software Safety Folder is to structure the documents and evidences that constitute it, objectives 3.5.X aim at recommending the structure proposed here after.

Part N°	Item title	Reference (Objective N°)
PART I: ENVIRONMENT		
I	System description	3.1.1; 4.1.1;
I	Operational environment	3.1.2;
I	Operating environment	
I	List of environment tools (CM, Development tools)	5.2.7; 6.2.1; 6.2.2; 7.1.1, 7.1.3;
PART II: SYSTEM SAFETY ASSESSMENT CONTEXT		
II	Regulatory framework	3.1.3;
II	Applicable standards	3.1.4;
II	FHA & PSSA output	3.0.6; 4.1.2; 4.1.3;
II	SW “FHA” & “PSSA”	3.1.5; 3.3.1; 3.3.2; 3.3.3; 3.3.4; 3.3.5
PART III: SOFTWARE SAFETY ASSESSMENT PROCESS		
III	Plan for Software Safety Assessment	3.2.1; 3.2.2; 6.1.1; 6.1.2; 6.1.3; 6.1.5;
III	Review of Plan for Software Safety Assessment	3.2.3; 6.1.4;
III	List of Plan for Software Safety Assessment recipients	3.2.4;

Part N°	Item title	Reference (Objective N°)
III	SW safety assessment process V&V	3.3.1; 3.4.2; 3.4.3; 6.3.1; 6.3.2; 6.3.3;
III	List of documents and the documentation process	5.1.1; 5.1.2; 5.1.3; 5.1.4;
PART IV: SAFETY REQUIREMENTS		
IV	(SW) Safety Requirements	3.0.6;
PART V: SW Modifications		
V	Change	3.0.12; 4.5.2; 4.5.4; 5.2.3; 5.2.4;
V	Problem Resolution	4.5.2; 4.5.4; 5.2.3; 5.2.4; 5.8.1; 5.8.2; 5.8.3;
V	Decommissioning	4.5.5;
PART VI: COTS		
VI	Lifecycle: Acquisition and integral process plans	7.2.1;
VI	Lifecycle: Transition criteria	7.2.2;
VI	Assurance: COTS plans consistency assurance	7.2.3;
VI	Assurance of ANS requirements satisfaction by COTS	7.2.4;
VI	Assurance: Lifecycle data adequacy assurance	7.2.5;
VI	Requirements: Derived requirements	7.2.6;
VI	Assurance of COTS compatibility with target computers	7.2.7;
VI	Assurance of COTS configuration and data items identification	7.2.8;
VI	Modifications: problem reporting	7.2.9;
VI	Assurance of COTS release incorporation control	7.2.10;
VI	Assurance of COTS configuration and data items archive	7.2.11;
PART VII: ASSURANCES		
VII	Tools assurance (development, CM, maintenance, ….)	7.1.2, 7.1.4; 4.3.12; 4.3.17; 4.3.18;

Part N°	Item title	Reference (Objective N°)
		4.3.19; 5.2.8; 6.2.3;
VII	Operation assurance	4.4.1; 4.4.2;
VII	SWAL: SWAL Rigour variation, assurance, monitoring	3.0.8; 3.0.10; 3.0.11; 4.4.3;
VII	Audit, Joint review reports	5.6.1; 5.6.2; 5.6.3; 5.7.1; 5.7.2; 5.7.3; 5.7.4; 5.7.5
VII	Training assurance	6.4.1; 6.4.2; 6.4.3;
VII	Requirements completeness and correctness assurance	3.0.2; 5.4.3; 5.4.4; 5.4.5; 5.4.6; 5.4.8; 5.4.9; 5.4.11;
VII	Requirements Traceability assurance	3.0.3; 4.3.15; 5.4.10;
VII	Unintended functions assurance	3.0.4;
VII	Requirements satisfaction assurance	3.0.6;
VII	Configuration Management Assurance	3.0.7; 5.2.1; 5.2.2; 5.2.4; 5.2.5; 5.2.6; 5.2.9; 5.4.13
VII	Development assurance (includes SDP: SW Development Plan)	4.3.3; 4.3.4; 4.3.5; 4.3.6; 4.3.7; 4.3.8; 4.3.9; 4.3.10; 4.3.11; 4.3.13; 4.3.14; 4.3.16; 4.3.20;
VII	Maintenance assurance	4.5.1; 4.5.3;
VII	Verification assurance (verification of the verification process results)	5.4.2; 5.4.7; 5.4.12;
VII	Quality Assurance	5.3.1; 5.3.2; 5.3.3; 6.3.1; 6.3.2; 6.3.3;
VII	Assurance that SW is acceptably safe	The SSF.

Page intentionally left blank.