EUROPEAN ORGANISATION
FOR THE SAFETY OF AIR NAVIGATION

EUROCONTROL

ESARR ADVISORY MATERIAL/GUIDANCE MATERIAL
(EAM/GUI)

EAM 6/GUI 1

ESARR 6 GUIDANCE TO ATM SAFETY REGULATORS

Explanatory Material on ESARR 6 Requirements

Edition	**:**	**0.05**
Edition Date	**:**	**22 December 2006**
Status	**:**	**Working Draft**
Intended for	**:**	**Restricted SRU**
Category	**:**	**Guidance Document**

SAFETY REGULATION COMMISSION

F.2 DOCUMENT CHARACTERISTICS

TITLE		
EAM 6/GUI 1 **ESSAR 6 Guidance to ATM Safety Regulators – Explanatory Material on ESARR 6 Requirements**		
Document Identifier :	**Reference :**	EAM 6/GUI 1
filename	**Edition Number :**	0.05
	Edition Date :	22-12-2006

Abstract :

This guidance material has been prepared by the Safety Regulation Commission to provide guidance for ATM safety regulators and support the implementation of ESARR 6 – Software in ATM Systems.

The main purpose of this document is to provide guidance about the provisions established in ESARR 6, Obligatory Provisions. Each requirement is illustrated by giving explanatory material that includes a rationale, the most significant implications for both Regulator and Provider, and information about further development.

This is the first deliverable of a series of guidance documents to be developed by SRC relevant for ESARR 6.

Keywords :

ESARR 6	ATM Software	Software requirements
Safety Assurance	Configuration Management	Verification

Contact Person(s) :	**Tel :**	**Unit :**
Antonio Licu	+32 2 729 34 80	DGOF/SRU

DOCUMENT STATUS AND TYPE					
Status :		**Intended for :**		**Category :**	
Working Draft	☑	General Public	☐	Safety Regulatory Requirement	☐
Draft	☐	Restricted EUROCONTROL	☐	ESARR Advisory Material	☑
Proposed Issue	☐	Restricted SRC	☐	Comment/Response Document	☐
Released Issue	☐	Restricted SRU	☑	Policy Document	☐
				Document	☐

SOFTCOPIES OF SRC DELIVERABLES CAN BE DOWNLOADED FROM :

www.eurocontrol.int/src

F.3 DOCUMENT APPROVAL

The following table identifies all management authorities who have approved this document.

AUTHORITY	NAME AND SIGNATURE	DATE
Quality Control (SRU)	(Daniel HARTIN)	
Head Safety Regulation Unit (SRU)	(Peter STASTNY)	
Chairman Safety Regulation Commission (SRC)	(Philip S. GRIFFITH)	

F.4 DOCUMENT CHANGE RECORD

The following table records the complete history of this document.

EDITION NUMBER	EDITION DATE	REASON FOR CHANGE	PAGES AFFECTED
0.01	04-Jun-02	Creation – First working draft 0.01 from SRU. Using available material following ASW 4 meeting (May 2002).	All
0.02	05-Jul-02	Revisions after ASW 5 meeting to capture the changes in ESARR 6 (ref. ESARR 6 ed. 0.10).	All
0.03	02-Sep-02	Revisions following Norway comments on edition 0.02.	5.1 & 6
0.04	25-Oct-02	Revisions following ASW 6 meeting and consultation thereafter. Main changes due to new edition ESARR 6 WD 0.12 and ESARR 6 Draft Issue 0.1. Document format also updated.	All
0.05	21-Dec-05	Revisions to all sections as part of a project to extend the level of guidance provided for ESARR6.	All

F.5 CONTENTS

F.6 EXECUTIVE SUMMARY

This guidance material has been prepared by the Safety Regulation Commission to provide guidance for ATM Safety Regulators and support the implementation of ESARR 6.

Within the overall management of their ATM services, ATM service-providers shall have in place safety management systems (SMS) in accordance to ESARR 3. In order to deal with deployment of software, additional safety assurances are required to ensure that risks associated with operating ATM software have been reduced to a tolerable level.

ESARR 6 requires the Designated Authority to ensure adequate and appropriate safety regulatory oversight to verify that services as part of its safety oversight. This guidance material will give an insight of what specific steps, ATM safety regulators may take when dealing with approval of service provider operations supported by software functions.

The main purpose of this document is to provide guidance about the provisions established in ESARR 6 mainly to obligatory provisions. Each requirement is illustrated by giving explanatory material that includes a rationale, the most significant implications for both Regulator and Provider, and information about further development.

1. INTRODUCTION

1.1 Scope of the Document

The main purpose of this document is to illustrate the provisions of Section *Obligatory Provisions* laid down in ESARR 6 and facilitate its interpretation. To enlarge the explanations, the non obligatory provisions have been also included to better explain the rationale and the Safety Objective of this safety regulatory requirement

1.2 Interpreting the Document

A standardised approach to the formatting of EUROCONTROL Safety Regulatory Requirements is used to reference, and to clarify, the status of information contained in the documents.

The document includes a Section 6 to provide guidance considered necessary to achieve the stated safety objectives. This section includes all applicable mandatory requirements (expressed using the word “shall”), including those relating to implementation.

[[CWJ: I have created a placeholder for this section 6 but will need to consult with everyone before writing it just to make sure I am clear on the general outline of the content]]

To ease the reading of the document the following editorial decoding needs to be used:

- whenever a text is highlighted in boxes as in the below example it represents a copy of text as was agreed in ESARR 6

Example:

i) ESARR 6 concerns the use of software in safety related ground-based ATM (Air Traffic Management) systems.

- The rest of text and pictures are used to interpret the requirements of ESARR 6 and to give additional guidance material to the ATM Safety Regulators in respect of usage and applicability of Safety Regulatory Requirements “Software in ATM systems”.

- The text in *[square brackets and italics]* represents editorial notes indicating in the working draft and draft editions places where additional text or pictures are needed to be added.

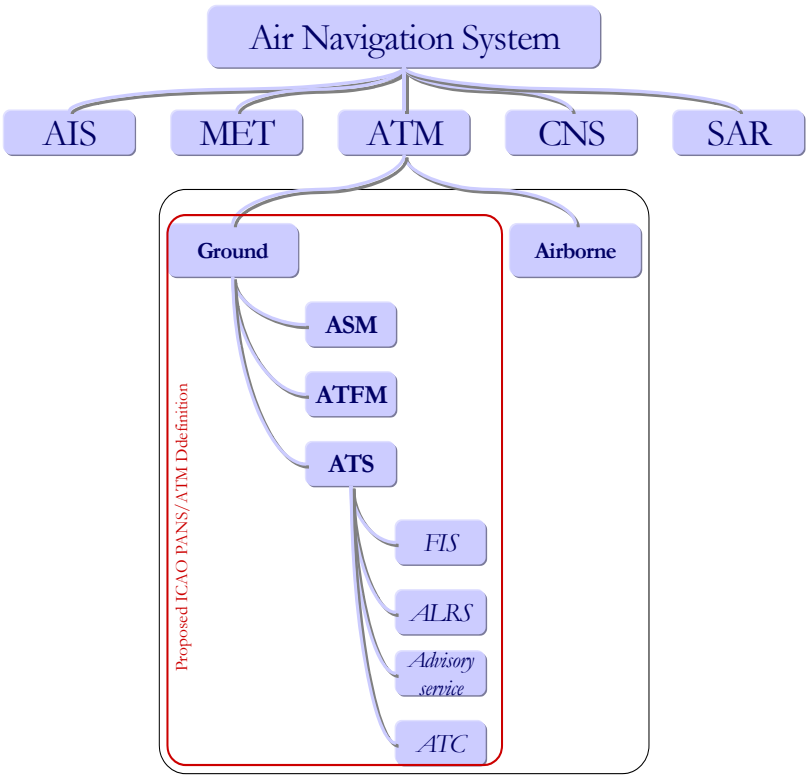
2. SECTION A – SCOPE

(*Introductory Material – The provisions of this section in ESARR 6 are not obligatory)*

i) ESARR 6 concerns the use of software in safety related ground-based ATM (Air Traffic Management) systems used for the provisions of ATM services to civil air traffic, including the periods of cutover (hot swapping).

ii) The scope of ESARR 6 is confined to the ground component of ATM and as such, its applicability cannot be claimed, unless modified and adequately assessed, for the airborne or spatial component of ATM systems. Nevertheless, ESARR 6 applies to the supporting services, including CNS systems, under the managerial control of the ATM service-provider.

For the current development of ESARR 6, the scope has been restricted to the ground component of ATM and as such, its applicability cannot be claim, unless modified and adequately assess, for the airborne or spatial component of ATM systems. ESARR 6 applies to the supporting C, N, S systems similarly like ESARR 3.



The CNS/ATM concept as defined by ICAO is a large scale concept which aims at achieving improvements in the global aviation system, that is increasing safety of flights, capacity and flexibility of air traffic and, as a consequence, decreasing delays and operating costs. As a result, it encompasses many sectors and domains. Thus, a CNS/ATM application is generally implemented with multiple components, in satellites, aircraft systems, telecommunication networks and air traffic control systems. ESARR 6 requirement is not intended to embrace the whole ICAO CNS/ATM concept for software aspects. It is only a contribution to this achievement limited to software operated in safety related ground-based ATM systems supported by ground C, N and S functions.

However, it is important to stress the interdependences that exist between the systems that support ATM infrastructures extending to ground, airborne and space based systems. Many of the provisions within ESARR6 can be usefully applied to this wider class of systems. However, this is not the focus of the regulatory requirement. The rationale for this emphasis on ground based is to address a perceived lack of provision in this particular area. In contrast, there is a host of existing regulatory provision covering these other more diverse systems. For example, ED-12B/DO-178B provides part of the regulatory background for airborne systems. Documents such as the EUROCAE ED-109 Guidelines on Software Integrity Assurance can also be used to support the development of space based applications. Both of these documents have been used to support the development of guidance material for ESARR6. This is a deliberate decision intended to ensure that the material for ground based software systems integrates well with guidance and regulatory support for these wider aspects of air traffic management infrastructure.

[[See chapter 1, page 1, of ED-109]]

iii) ESARR 6 assumes that an a priori risk assessment and mitigation process is conducted to an appropriate level to ensure that due consideration is given to all aspects of ATM including ATM functions to be performed by software. Additionally ESARR 6 assumes that the effectiveness of risk assessment and mitigation associated with software malfunctions or failures is already in place.

Existence of a risk assessment and mitigation process necessary to assess the criticality of ATM functions supported by software is a pre-requisite of application of ESARR 6. This actually is required by:

- ESARR 3 section “5.2.4 Risk Assessment and Mitigation

Within the operation of the SMS, the ATM service-provider;

a) shall ensure that risk assessment and mitigation is conducted to an appropriate level to ensure that due consideration is given to all aspects of ATM;

b) shall ensure that changes to the ATM system are assessed for their safety significance, and ATM system functions are classified according to their safety severity;

c) *shall ensure appropriate mitigation of risks where assessment has shown this to be necessary due to the safety significance of the change;*

- ESARR 4 – Risk Assessment and Mitigation in ATM section 5.1, 5.2 and 5.3 which are not reproduced here for the sake of brevity. Additionally in ESARR 4 the link with ESARR 6 is further made through section "*8.2.2 Link with ATM software qualification;*

  8.2.2.1 - The safety objectives allocated to each hazard drive the determination of specific means to attain the proper level of confidence in the success of implementing the mitigation strategies and related safety requirements.

  8.2.2.2 - These means may include a set of different levels of constraints being set on specific software elements of the ATM System".

  Software cannot kill or injure anyone unless it has some influence on the operation of equipment including aircraft, ground vehicles, construction equipment etc. Hence risks must be considered in terms of the adverse events that are associated with 'Equipment Under Control'. Hence risk assessments proceed by considering the hazards that stem from the wider systems that are being controlled. This helps to shape what are termed Functional Hazard Assessments and Preliminary System Safety Assessments. Software frequently plays a role in the mitigation or reduction of the risks identified in these assessments. For example, Short Term Conflict Alert Systems (STCA) can provide a last resort or safety net against the general hazard created by AIRPROX incidents. It follows that the criticality or important of the function provided by the software is measured in terms of the risk reduction that is intended to be provided by that software. If a piece of code reduces an unacceptable risk to one that is now acceptable then it can be argued that the safe operation of the system now relies on that software and, in consequence, additional development resources should be allocated to ensure that the code will function in a reliable and timely manner.

  The EUROCONTROL Recommendations for Air Navigation Systems Software provide a strong rationale for the approach advocated above and embodied within ESARR6. This establishes the lifecycle requirements for Air Navigation Systems software within the context of a wider risk assessment process structured around techniques such as those embodied within the EUROCONTROL Safety Assessment Methodology (SAM). [[See Chapter 1, page 2 of SAF.ET1.ST03.1000]

Therefore;

- It is assumed that the risk assessment and mitigation process derives system-level safety requirements from a hazard and risk analysis of the ATS environment in which the system is required to operate.

- It is assumed that a necessary and sufficient set of system-level safety requirements exist, which describe the functionality and performance required of the system in order to support a tolerably safe ATS.

- It is assumed that the failure modes which the software must detect and mitigate in order to meet the system safety requirements have been identified e.g. those failure modes associated with: other systems, system-system interactions, equipments, pre-existing software and all user-system interactions.

- It is assumed that the failure modes identified include generic failures relevant to the safety related ATS application, e.g. security threats, loss of communications, and loss of power.

- It is assumed that the failure modes identified (including human errors) are representative of the operational environment for the system and workload on the system operators.

The previous paragraphs raise a number of key points that require additional guidance and some supporting rationale. In particular the emphasis on the interaction between Air Navigation Systems and their environment is a critical aspect of risk assessment. Changes in the systems being used can alter the risk profile of operational practices; for example the loss of the SWI communications system arguably added to the burdens on ATCOs during the Überlingen accident[1]. Similarly, changes in the operating environment can also affect the risks associated with air traffic service provision. For example, changes in the mix between general and commercial aviation formed part of the background leading to the Linate[2] runway incursion. Hence in order to assess the degree to which software may reduce the risks associated with service provision it is necessary to consider the current state as well as potential changes both to Air Traffic systems and to their operating environment.

The second point, mentioned above, is that there must be both a necessary and a sufficient set of system level safety requirements before any risk assessment can be completed. Informally, a necessary requirement is one that if it were violated then the system as a whole would have failed. If we forget to include a necessary functional requirement then some key aspect of the infrastructure will have been omitted. For example, a necessary requirement of air traffic service provision is to ensure adequate separation. Sufficient requirements collectively describe conditions that if they all hold then the system is successful. If we do not have a sufficient set of requirements then some aspect of the system will also be perceived to have failed. For instance,

[1] http://www.bfu-web.de

[2] http://www.ansv.it

although separation is a necessary requirement it is not sufficient on its own. In particular, it is important to ensure that aircraft arrive at their intended destination in a timely manner. Hence a sufficient set of requirements must also take these constraints into account. The importance of the previous paragraph is that if any of these requirements are omitted then it can be difficult to accurately conduct the system level risk assessments that are a prerequisite for the assessment of software criticality. For example, if an initial risk analysis did not consider the need to support on-time departures in poor visibility then many aspects of the subsequent development might be compromised because the hazards that relate to these operations would not have been considered. Hence, it would not have been possible to identify the importance of software components that might be necessary to reduce the risks associated with poor visibility operations.

The third point in the previous list is strongly related to the identification of failure modes. Once the functional requirements can be identified for Air Traffic Systems, it is important to consider the different ways in which they may fail. For example, a failure is total if it prevents the system from providing a particular function from the moment at which it occurs. A partial failure may degrade the provision of a function but will not totally eliminate it. An intermittent failure removes some or al provision of a system function but only during particular intervals of time at other times full functionality is resumed. Within each of these high-level categories there are more complex modes that must be considered during a risk assessment. The key insight here is that unless we consider a broad range of failure modes then it is unlikely that we will be able to adequately address the broad range of hazards that might have to be mitigated by the introduction of safety-critical software.

The fourth bullet point builds on this by identifying several broad classes of failures that must be considered during any analysis of potential failure modes. The final item in particular focuses on the importance of human intervention when considering the environment during any risk assessment. This is critical because operator involvement can significantly increase the complexity of any risk assessment given the many different ways in which ATCOs, managers and technical staff could inadvertently undermine key system functionality. This was a key finding of both the BFU report into Ueberlingen and the ANSV investigation of Linate. However, if human intervention is not considered within a preliminary risk assessment then it is unlikely to adequately reflect the true operational environment of Air Navigation Systems. In consequence, it would be difficult both to anticipate the need for software risk mitigation and to adequately assess the criticality of any existing software provision.

iv) ESARR 6 does not prescribe any type of supporting means of compliance for software. This is the role of software assurance standards. It is outside the scope of this requirement to invoke specific national or international software assurance standards.

A key issue here is that software development technique are likely to change rapidly over time as new hardware and software platforms emerge. Any regulatory instrument that embodies or advocates particular development techniques is, therefore, likely to have an extremely short shelf-life. There are also strong national and international differences over the suitability of particular development methodologies within the context of their national systems in terms of cultural, commercial and technical concerns. Hence, not only would the validity of any regulatory instrument be undermined by the

inclusion of such recommendations, it might also impose inappropriate and unnecessary constraints on those who must apply their provisions.

3. SECTION B – RATIONALE

(Introductory Material – The provisions of this section in ESARR 6 are not obligatory)

i) The SRC decision number 6/8/5 approved the inclusion of the development of a EUROCONTROL Safety Regulatory Requirement for software-based ATM systems in the SRC work programme. It is recognised that there is no precedent in this area neither by ICAO nor by any other international regulatory body responsible for ATM system safety.

The concern to develop regulatory material specifically to support software development in ATM systems reflects the growing importance of programmable systems within aviation safety. At the time when the ESARR was created, there was little or no specific guidance on appropriate techniques for software development within this domain. More general standards, such as IEC61508, provided some guidance but lacked the specific focus of the EUROCONTROL requirements. There development of ESARR6 can also be justified in terms of the need to integrate the requirements for software development within the suite of other regulatory instruments in European Air Traffic Management. The following paragraphs will explain the importance of creating specific provisions governing software development that support and are supported by the provisions within ESARR3 on Safety Management Systems and ESARR4 on risk assessment.

In addition, it is important to consider the justification for developing a separate ESARR dealing with software. Programmable systems introduce considerable opportunities for innovation. They support the integration of many diverse applications and hence can be used in safety related systems to mitigate against many different hazards. This increases their importance for the overall system. However, software also fails in novel ways that are quite different from hardware systems. Software does not age in the way that mechanical devices will wear out. A logical fault may remain hidden for weeks, months even decades without causing any problems until the relevant section of code is called upon. This property is compounded by the impossibility of testing every possible execution path through many complex software applications given that they rely on many million sets of instructions that can be contingent on multiple combinations of operator input and environmental observations. One consequence is that conventional testing techniques can only be used to identify the presence of bugs and not their absence; because we cannot be sure that we have covered all possible sequences of instructions. The difficulty of testing software has a knock-on effect in terms of project management. It can be difficult to know when enough resources have been devoted to software development and problems identified late in the lifecycle can be extremely expensive to correct. All of these reasons provide the rationale for a set of regulatory requirements that specifically address software in ATM systems.

ii) ESARR 3 (Use of Safety Management Systems by ATM Service Providers) requires that safety management systems include risk assessment and mitigation to ensure that changes to the ATM system are assessed for their significance and all ATM system functions are classified according with their severity. It also requires assurance of appropriate mitigation of risks where assessment has shown this to be necessary due to the significance of the change.

The previous paragraphs of guidance material referred to the importance of ESARR6 in helping create a consistent and comprehensive approach to regulation in Air Traffic Management. In particular, the introduction of specific provisions for software development helps to reinforce particular sections within ESARR3. This more general guidance on Safety Management Systems provides the context for ESARR6 by describing iterative approaches to the improvement of system safety where risk assessment, design innovation and operational experience help to form a 'virtuous circle' by which appropriate lessons are learned from the small number of adverse events that do occur.

There are multiple links and dependencies between ESARR3 and ESARR6. For example, the safety management systems within ESARR3 help to ensure that operational staff and safety managers cooperate to monitor adverse events and their precursors. This helps to both validate and extend existing risk assessments in the light of operational experience. It follows that if a risk assessment does not mirror the actual incidents that are being observed then there is a risk that it will not adequately anticipate potential problems. In consequence, it is unlikely that the software mitigation described within ESSAR6 will adequately address key safety concerns.

The provisions of ESARR3 are also important in other ways. For example, software failures must be fed back into the operational experience that informs the risk and criticality assessments proposed in ESARR6. The following sections of this guidance document will return to this issue in further detail, describing the integration of information about software behaviour within the wider safety management systems of ESARR3.

iii) ESARR 4 (Risk Assessment and Mitigation in ATM) expands ESARR 3 requirements on Risk Assessment and Mitigation, and provides for a comprehensive process to address people, procedures and equipment (software and hardware), their interactions and their interactions with other parts of the ATM system when introducing and/or planning changes to the ATM System.

As mentioned, ESARR6 provides an important component in the landscape of regulatory requirements that help to shape practice in European Air Traffic Management. In provides a specific focus in a key area for the more general ESARRs. The previous paragraphs have introduced the safety management systems perspective embodied within ESARR3. ESARR4 provides a more precise focus on the requirements for risk assessment and mitigation. It distinguishes between three broad areas of concern: people; procedures and equipment. Hazards stem both from within these areas and in the interactions between them. Software and hardware are explicitly distinguished with the equipment component, although as mentioned previously, software cannot by itself lead to significant adverse effects unless it affects hardware systems. The provisions dealing with software systems in ESARR4 can be summarised

by the following excerpt from the regulatory requirements from section 8.2.2 entitled 'Link with ATM Software Qualification':

8.2.2.1 The safety objectives allocated to each hazard drive the determination of specific means to attain the proper level of confidence in the success of implementing the mitigation strategies and related safety requirements.

8.2.2.2 These means may include a set of different levels of constraints being set on specific software elements of the ATM System.

(ESARR4, Page 11)

As can be seen, the provisions within ESARR4 are consistent with the broad scheme identified in ESARR6. Each hazard is associated with a safety objective. If this objective is achieved then the associated risk will be acceptable. This concept of an 'acceptable risk' is important because it is, typically, not possible to guarantee absolute safety given finite resources of money, time and expertise. In consequence, all that we can do is demonstrate that the risks which remain in an application are broadly acceptable or that it is impracticable to support any further risk reduction. This would be the case if, for example, additional safety investments were to completely undermine the viability of particular operations. In order to achieve these safety objects we must employ mitigation strategies and 'related safety requirements' that often involve software systems and these must be developed in such a way that we have sufficient 'confidence' they will satisfy the overall objectives.

Clause 8.2.2.2 in ESARR4 establishes the background for ESARR6 by recognizing that there may be different levels of confidence associated with different software components. For example, software mitigating low risk events will be associated with a lower level of criticality and hence may be subject to a more flexible set of constraints over its development and testing than software that is used to mitigate against high consequence of very likely failures. Hence the previous two clauses illustrate the close complementary relationship between ESARRs 4 and 6.

iv) ESARR 6 is the continuation of this safety regulatory build up process and expands ESARR 4 in regard with the software aspects of ATM systems. Complementary safety regulatory requirements for hardware aspects are under consideration.

[[CWJ Is it still the case that a separate ESARR is under consideration for hardware aspects?]]

The previous clause expands on the argument that has already been sketched in other areas of this guidance material. As mentioned, the unique characteristics of software, in terms of its failure modes and the difficulty of testing, as well as the increasing reliance on programmable systems in risk mitigation make it critically important that we expand and focus the regulatory framework that is provided within the risk assessment provisions of ESARR4.

v) Safety is an essential characteristic of ATM systems. It has a dominant impact upon operational effectiveness. ATM systems involving significant interactions in a continuously larger integrated environment, automation of operational functions formerly performed through manual procedures, increase in complexity. The massive and systematic use of software to challenge ATM system complexity, is now demanding a more formal approach to the achievement of safety.

The increasing pressures to improve performance, in terms of increased throughput and reduced mean delays, have had a significant impact upon ANSPs. At the same time there are requirements both to maintain and improve safety performance against a wide range of benchmarks. All of these targets must often be achieved within stringent financial constraints. One consequence of all of these disparate pressures has been to significantly increase moves towards technological innovation through the development of advanced software systems in many operational areas. These innovations have increase the interconnections and dependencies between subsystems, for example between flight planning and radar systems or between multiple sectors and flight levels. These interconnections mean that a fault in one area can have a massive impact on other aspects of ANSP operations. For example, the infrastructure work on the Geneva control room affected many of the systems that ATCOs interacted with and not simply the radar monitoring facilities that were at the heart of the upgrades prior to the Ueberlingen mid-air collision.

Complexity not only stems from the interconnections that software creates between specific subsystems, it also reduces the time margins that formerly existed in many aspects of operations. Digital flight strips can be instantaneously transferred between desks. Although this automation offers many benefits, it also reduces some of the opportunities for recall and reflection that characterised interaction with physical strip. These issues of integration and reduced margins are only two aspects of complexity amongst many others. However, they are sufficient to illustrate that software creates many advantages but also introduces many design issues and potential vulnerabilities that require a systematic approach to design if programmable systems are not to create as many risks as they help to mitigate.

The purpose of this requirement is to provide ATM safety regulatory bodies and ATM service providers with a uniform and harmonised set of safety regulatory requirements for software in ATM systems.

This aspect of ESARR6 is self-evident. A key concern is to establish minimum applicable standards that can be shared across different countries while at the same time allowing a diversity of approach in the implementation practices that is appropriate for the varying needs of different ANSPs. By having common requirements, it is also possible to exchange best practice in meeting the constraints of ESARR6 within a wider community.

4. SECTION C – SAFETY OBJECTIVE

(Introductory Material – The provisions of this section in ESARR 6 are not obligatory)

i) The prime software safety objective to be met for ATM systems that contain software, is to ensure that the risks associated with operating ATM software have been reduced to a tolerable level.

To achieve the above safety objective a number of safety regulatory requirements are placed on the responsibility of;

- ATM service Provider as part of its responsibility to ensure provision of safe services,

- the Designated Authority as part of its responsibility to;

  - set minimum acceptable levels of safety (in the public interest), including by means of target levels of safety,
  - define applicable national safety regulatory requirements, including those necessary to meet international commitments,
  - define any relevant Standards and Practices that apply to support or complement the requirements,
  - ensure that minimum acceptable levels of safety are met by service-providers,
  - ensure ongoing compliance with national safety regulatory objectives and requirements.

The opening of Section C builds on the previous observation that software, typically, helps to mitigate risks associated with hazards that are 'realised' by equipment and staff. The software itself cannot directly cause any injury within an ATM system. Hence the focus here is on the risks associated with OPERATING the software and not the software itself. The objective of ESARR6 is to reduce any residual risk so that it is at a tolerable level. Previous sections have referred to this tolerance and it is important to emphasise that this is not an absolute judgement. In other words, it is neither appropriate nor is it technically feasible to define in quantitative terms what would be a 'tolerable' residual risk within an Air Traffic Management system. The definition of tolerability is determined by social, political and environmental factors. Hence, there are strong differences between different areas of the globe in terms of the level of acceptable risk within Air Traffic Management. In economies that are undergoing rapid economic development from a relatively low base, there is often a greater tolerance for risk than would be the case in more mature economies that already have relatively high standards of safety in other industries. Similarly, ANSPs that have a relatively poor safety record may also find that the public tolerance for risk from ATM related software would be considerably reduced by the negative reaction to previous fatalities. Having made these general remarks, it is clearly important to establish minimum standards across member states and so ESARR6 helps to identify common practices that together will tend to ensure the broad tolerability for software related systems in the mitigation of ANS risk between different states.

Subsequent clauses in Section C on Safety Objectives help to establish an organisation set of responsibilities for the provisions within the regulatory requirements of ESARR6. The requirements to support software safety are part of the wider responsibilities on ANSPs to ensure the provision of safe services.

This note also refers to the designated authorities that are established in each member state to regulate the activities of the national ANSP. The reference to 'public interest' reinforces the earlier comments about the subjective nature of 'tolerable safety' in which national public opinion can play a strong role. These designated authorities must take the public view into account when establishing the measurable targets for safety that provide a concrete representation of the more subjective bounds for tolerable levels of safety related performance. In other words, in the immediate aftermath of an accident the general public may have unrealistic expectations for safety targets and may be extremely intolerant of any risk however remote. The designated authority must carefully balance this strong public view against the reasonable technical objectives that might be achieved by an ANSP. Setting objectives that are technically or economically infeasible can lead to a culture of cynicism and tolerance that discredits the most fundamental components of a regulatory framework.

The second bullet point relating to the designated authority reiterates the obligations that they owe to other international organisations in establishing necessary national requirements for software related systems. This is important because EUROCONTROL is one of several bodies that support safety improvements across the aviation industry. Previous sections have cited companion documents, guidance material and standards from bodies such as the ICAO that apply in addition to the regulator structures in ESARR6.

The role of the designated authority goes well beyond simply drafting national regulations to implement and refine those provided in the suite of ESARRs. They must also monitor their effective implementation across a national aviation industry. They must determine whether or not organisations actually satisfy the process requirements that are typically outlined in national requirements. For example, they must reassure themselves that adequate risk assessments have been done to ensure that the criticality of software components is closely related to the hazards that they are intended to address. Designated authorities must also conduct a higher level monitoring function to determine whether these particular processes actually do help to achieve the overall safety targets that have been identified for national service providers.

[[CWJ: Note for discussion: there are some issues with this opening statement because as it is set up in ESARR6 – just as in IEC61508, the risks are associated with the Equipment Under Control and not directly with the software? I've tried to smooth over this interpretation issue.]]

5. OBLIGATORY PROVISIONS

5.1 ESARR 6 – Section 1 – General Safety Requirements

Guidance in this section elaborates the general Safety Requirements from ESARR 6 section 1 of Obligatory Provisions.

1.1 Within the framework of its Safety Management System, and as part of its risk assessment and mitigation activities, the ATM service-provider shall define and implement a Software Safety Assurance System to deal specifically with software related aspects.

The unique nature of software and its growing importance within the provision of air traffic services helps to justify the development of a specific Software Safety Assurance System. As mention, software does not age in the same way that hardware. Hence, we cannot simply reuse preventative maintenance techniques to help improve reliability and availability. In contrast, the introduction of software updates paradoxically usually increases the chances of an immediate failure in a way that goes well beyond the 'burn in' effects that characterise some hardware components. The establishment of a specific assurance system helps reflect the unique demands of software development. It can create the organisational credibility and funding streams that are necessary to adequately resource this function within large, complex and often distributed service providers.

1.2 The ATM service-provider shall ensure, as a minimum, within its Software Safety Assurance System that; a) The software requirements correctly state what is required of the software by the risk assessment and mitigation process, b) Traceability is addressed in respect of all software requirements, c) The software implementation contains no functions which adversely affect safety, d) The ATM software satisfies its requirements with a level of confidence which is consistent with ESARR 6, e) Assurances that the above requirements are satisfied, are at all times derived from a known executable version of the software, a known range of configuration data, and a known set of software products and descriptions (including specifications) that have been used in the production of that version.

The previous clauses help to establish high level objectives for the Software Safety Assurance System. Point a) establishes a duty to verify that the software requirements actually capture the constraints identified by the need to mitigate particular risks. This is important because there is a danger that the products of a risk assessment are not carried forward into the software acquisition process. In such circumstances, the ANSP would support each necessary stage within ESARR6 but the integrity of the transitions between stages would not be maintained.

The second point, labelled b), is a critical requirement within ESARR6. In some ways, this is a more general constraint that that encapsulated within the previous item. Traceability enables independent observers and analysts to reconstruct the path from an initial risk assessment through the design of mitigation strategies to software criticality assessments and on into implementation. The key idea is that it should be possible to look at any piece of code and identify its criticality level and then to justify or explain its importance in terms of mitigating key systems risks.

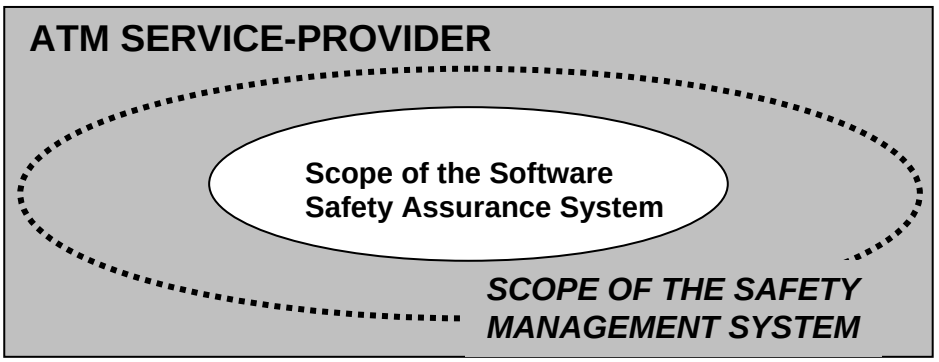
Item c) is more complex and difficult to satisfy. The requirement that software contains no functions that adversely affect safety can be extremely difficult to prove. Arguments on previous experience can be unreliable. Simply because a piece of code has functioned without bugs in the past provides no guarantee to future safety. Subtle changes in the environment or in operational practices can lead to input values that trigger the execution of instructions that have not been used in previous operations. Similarly, dynamic testing cannot easily be used to examine the many millions of instruction sequences that are encapsulated within even relatively commonplace systems in Air Traffic Management. Static inspections often fail to identify the environmental factors and operational behaviours that can trigger software failure. In spite of these technical and theoretical caveats, ESARR6 clearly charges designated authorities with responsibility for the provision of software that does not adversely affect safety. Hence it is up to the authority to determine whether or not an ANSP has discharged their obligation under ESARR6 to apply the appropriate blend of techniques that is required to increase confidence in safety related software even when it is impossible to establish 'safety' in an absolute sense.

The fourth item in the list reiterates the previous point. It charges the designated authority with ultimate responsibility for ensuring that ANSPs and other associated companies develop software that meets the requirements which are consistent with the required level of confidence. This level of confidence is linked back via the mitigation of risks, in the manner described in earlier sections of ESARR6 building upon ESARRs 3 and 4.

The item labelled e) again illustrates the need for a regulatory requirement that focuses directly on software systems. It includes the constraint that designated authorities must base their analysis on 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'. The importance of these requirements cannot be underemphasised. Software is based on a series of abstractions that can be modified, replicated, deleted with minimal effort. This creates considerable potential for confusion if small changes in the executable version of a program are not reflected by consequent changes in the support documentation. A key issue here is that traceability will not be possible unless ANSPs and their subcontractors have carefully developed policies for version control and modification tracking. Without this necessary infrastructure it will be possible to follow the development of mitigating factors from a risk assessment into code that is very different from that which is actually running on a given hardware system. Similarly, the configuration of the MSAW system

contributing to the Guam accident[3] has illustrated the importance of ensuring that ANSPs AND designated authorities actively consider the integrity of configuration data and not simply the sequences of instructions that form complex software systems.

Software Safety Assurance System (SSAS) is not a new sub-system required to the ATM service provider to be put in place, but is a constituent part of the Safety Management System as described in the figure below;



The SSAS is partly covering both the in Achievement and the Assurance layers in the SMS, when dealing with ATM software.

The previous diagram clearly illustrates that the software safety assurance system is a component of the overall safety management system. This is justified by variants of the arguments that have been presented in previous sections. It is difficult to adequately assess the overall safety of any proposed air traffic management system unless software related risks are explicitly considered. Conversely, the ubiquitous nature of software has created a situation where it is increasingly involved in adverse events and hence there must be a mechanism for feedback information about failures involving programmable systems so that we can improve both risk assessment practices and software development techniques.

1.3 The ATM service-providers shall provide assurances, that the requirements in 1.2 have been satisfied, to the designated Authority as required.

Although the designated authority has ultimate responsibility for the oversight of the requirements listed above, it is clear that ATM service providers are responsible for their implementation and there is a requirement on them in ESARR6 to provide the designated authority with the assurances that these objectives have been met. It would be relatively easy to skip over this point and miss important implications. However, it is critical that ANSPs and associated sub-contractors are in a position to document compliance. This can be difficult for a number of reasons unless the requirement to provide relevant assurances is considered during the initial stages of software acquisition. For example, problems can arise if sub-contractors must disclose implementation details of code that is commercially sensitive. Alternatively, it can be difficult to meet traceability requirements for system that integrate new

[3] http://www.ntsb.gov/Publictn/2000/AAR0001.htm

software with legacy applications even though parts of these systems will be exempt from the provisions of ESARR6. Subsequent sections of this guidance document will deal in detail with the problems that arise when ANSPs must provide designated authorities with assurances about systems that involve COTS (Commercial Off The Shelf Components), however, many of the same issues arise in this context as for legacy systems where the developers may have left the company or companies involved.

[[CWJ: note – check that legacy systems are still exempt and if not then modify previous statement – check what happens in this situation?]]

Former 1.4 has been moved into ESARR 1.

It is the national (State) responsibility to ensure that the services provided meet minimum levels of safety in the public interest. Safety regulation is concerned with the safety competence of the organisations, of systems and of those individuals conducting safety related tasks. Requirement 1.4 placed on the Designated Authority responsibility is derived and makes part from the core three fundamental processes of safety regulation:

- setting safety regulatory objectives and requirements;
- ensuring safety regulatory approval of organisations, operations and where required of the individuals undertaking safety related tasks ;
- ensuring ongoing safety oversight

The requirement in 1.4 represents the direct link between ESARR 1 (national ATM Safety Regulatory Framework) and ESARR 6.

This excerpt reinforces and develops previous comments about the public acceptability of risk by placing responsibility on each State to ensure minimum standards. In particular, the notion that there are minimum standards goes beyond any definition of acceptability in terms of any purely public assessment. As noted previously, this is important given that designated authorities have to consider whether it is technically feasible to achieve the levels of safety that are often demanded in the aftermath of accidents or incidents. Conversely, they may have to argue to maintain expenditure on safety related systems at times when the public may view such investments as unnecessary given a previously good safety record.

The previous paragraph goes on to expand on Requirement 1.4 by identifying competence as a key issue for both the individuals and organisations involved in ensuring the safety of air traffic management services. It is hard to underestimate the importance of this issue. Even if an organisation establishes exhaustive safety management systems and conducts rigorous risk assessments, there is a danger that safety will be undermined if staff are not competent to implement these processes. These observations reinforce further links between the requirements of ESARR6 on software development and those of ESARR5 that describe key requirements for the recruitment and training of ATM personnel.

The three items in the previous list identify core objectives or responsibilities for the national designated authority. These high level goals provide a direct link between ESARR1, which describes the main components of national ATM regulatory frameworks and the organisations that are responsible for directly monitoring the provisions within ESARR6. In other words, ESARR1 describes the manner in which

designated authorities must establish safety objectives and requirements through regulatory intervention. They must also be responsible for issuing the approvals to individuals and organisations who conduct safety critical operations within Air Traffic Management. Finally, ESARR1 establishes the framework by which national designated authorities ensure that their safety oversight and the safety processes of the organisations they support are monitored on a continual basis. However, ESARR6 develops these high level requirements within the context of software systems in air navigation service provision.

5.2 ESARR 6 - Section 2 - Requirements Applying to the Software Safety Assurance System

Guidance in this section elaborates the requirements applying to the Software Safety Assurance System from ESARR 6 Section 2 of Obligatory Provisions.

2.1 The ATM service-provider shall ensure, as a minimum, that the Software Safety Assurance System - Is documented specifically as part of the overall Risk Assessment and Mitigation Documentation;

This clause reinforces the links between ESARR6 on software development and ESARR3 on the use of Safety Management Systems by ANSPs. In particular, it builds on the following excerpt from this existing regulatory requirement:

5.3. Requirements for Safety Assurance

Within the operation of the SMS, the ATM service-provider:

5.3.4. Risk Assessment and Mitigation Documentation
Within the operation of the SMS,

…shall ensure that the results and conclusions of the risk assessment and mitigation process of a new or changed safety significant system are specifically documented, and that this documentation is maintained throughout the life of the system.
(ESARR3, page 12)

The key issue here is that the ATM service provide must create and maintain a system for documenting the products of a Software Safety Assurance System within the wider processes for documenting risk assessment and mitigation. This is an important requirement because of the specialist, technical nature of software safety assessments. There is a danger that the individuals and teams responsible for this work will fail to adequately communicate their results to their co-workers who must support the wider systems risk assessments in other areas of ANSP operations. If the results of a software safety assessment are not well integrated with these wider processes of risk assessment and mitigation then there is a danger that the traceability issues mention in previous requirements of ESARR6 will not be achieved. In other words, it will be hard if not impossible to trace the ways in which particular sections of code help to mitigate the risks that arise from equipment under control. In particular it is important to stress the requirement from ESARR3 that documentation must not simply be developed during the initial stages of a project and then forgotten. ESARR6

continues the requirement that the links between software safety assessment processes and wider risk assessment processes must be documented and maintained during the operational lifetime of these systems, including decommissioning.

2.2 The ATM service-provider shall ensure, as a minimum, that the Software Safety Assurance System - Allocates software assurance levels to all operational ATM software;

Software assurance levels have been introduced here to allow levels of rigour of assurance to be defined and related to tolerable levels of ATM risk.

This clause reinforces the links between ESARR6 and a variety of similar standards including IEC61508 and EUROCAE ED109, cited in previous sections. These documents together with the EUROCONTROL regulatory requirement establish a framework by which software assurance levels help determine the development, verification and validation resources that are allocated to pieces of code. The assurance levels in turn are related to the risk and hazard assessments that shape the functional and non-functional requirements for the software. The key contribution of this section 2.2 in ESARR6 is to clearly state that it must be possible to identify the software assurance level that is associated with every section of code in ATM systems.

This is an important regulatory requirement. There can be complex interconnections and dependencies between ATM software. In consequence, it is possible for some code that is associated with a relatively high assurance level to be compromised by bugs in other software components that were not assigned to any particular level within this classification system. It also creates a considerable challenge for ANSPs given the diversity and scope of the software systems that are currently integrated into many different operational areas. It should be noted that the previous paragraph does not explicitly focus on 'front line' operations such as control room software but has instead a more general application.

2.3 The ATM service-provider shall ensure, as a minimum, that the Software Safety Assurance System - Includes assurances of software; ❑ requirements validity, ❑ verification, ❑ configuration management, and, ❑ traceability.

Clause 2.3 further develops the regulatory requirements in ESARR6 by looking beyond the development practices that might be associated with particular levels of software assurance to look at some of the wider stages of development that are more exhaustively considered within Chapter 5: Supporting Lifecycle Processes of EUROCONTROL's Recommendations for ANS Software (SAF.ET1.ST03.1000.GUI-01-00). Rather than repeat the exhaustive guidance provided by this document, the following paragraphs focus in on a number of key issues within this section of ESARR6.

The first bullet point refers to 'requirements validity'. Validation provides an assessment of the value or worth of particular requirements. This is important because a system may fail even if software meets the requirements that have been specified for it. For example, if those requirements fail to consider a number of important hazards then there will continue to be significant vulnerabilities within the system.

In contrast, the second item in the previous list refers to verification. This is the process by which we establish whether or not the software actually does meet those requirements. This is an important distinction. Validation can only be seen in terms of application goals, as a means of determining the value of a set of requirements. Verification can be seen as a more technical process of proving whether or not software meets a set of requirements. Hence it is closely related to issues of traceability between requirements and particular sections of code within an implementation.

The second bullet point in the previous list focuses on configuration management. This is critical because many software systems now provide ANSPs with considerable flexibility. Programmable systems enable the configuration of systems to be changed and modified in response to changes in the operational environment in ways that could not have been considered with previous generations of hardware based systems. However, this creates considerable risks. In particular, it can be difficult to determine which of many versions of a program is actually running on a target platform. It can also be difficult to ensure that the software which controls infrastructure configuration does not accidentally disable key support functions. Hence, the management of configuration information and its associated documentation are an important concern during the development and operation of ATM software.

Final point refers again to the issue of traceability. This relates to the ability to identify the links between risk analysis and mitigation, software requirements, criticality assessments, design and implementation documentation and testing. In other words, it must be possible for assessors to trace the way in which risk mitigation is implemented within particular lines of code in ATM software applications. If this cannot easily be done then there is a danger that some hazards will be overlooked while, conversely, unnecessary complexity may be introduced by, for instance, legacy code that does not address particular functional or non-functional requirements.

2.4 The ATM service-provider shall ensure, as a minimum that the Software Safety Assurance System - Determines the rigour to which the assurances are established. The rigour shall be defined in terms of a software assurance level, and shall increase as the software increases in criticality. For this purpose:

a) the variation in rigour of the assurances per software assurance level shall include the following criteria;

- required to be achieved with independence,
- required to be achieved,
- not required.

b) the assurances corresponding to each software assurance level shall give sufficient confidence that the ATM software can be operated tolerably safely.

[[CWJ: I have tried to incorporate this into the note below: Assurance of configuration and traceability cannot be varied with software assurance level e.g. either there is complete traceability or there is incomplete traceability – incomplete traceability is unacceptable.]]

The previous section from ESARR6 provides a further component of the software safety assurance framework developed in previous sections of this guide. In particular, it requires that ANSPs must use software assurance levels to determine the level of rigour that is used in developing code. The intention is that greater resources of time, effort and expertise should be allocated to the design, development and testing of software that is associated with higher assurance levels. This ensures that resources are allocated in proportion to the criticality of the mitigation function that is implemented by each section of code. The 'minimum' reference is used to indicate that additional resources may be allocated to software over and above those normally associated with a particular level of criticality, for example if it implements a particularly complex function. However, the resource allocation should never fall below the minimum associated with each level.

The subsequent enumeration indicates three different issues that must be considered when determining the degree of rigour that is associated within each software assurance level. It distinguishes between requirements that are to be achieved 'with independence', those that are required to be 'achieved' and those that are important but are not requirements in themselves. The term 'independence' is clarified within the appendices of ESARR6 as follows:

> **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. (ESSAR6, page 17)**

Hence human auditors can be used with automated tools to increase the independence of any verification carried out during the software assurance process. The implementation of such a regulatory requirement raises a number of practical issues. For example, it seems clear that ANSPs must assess the degree of independence that is to be achieved. This can determine whether or not external agencies must be used or whether independence can be achieved through inspections by individuals and groups from other areas of an organisation. Similarly, supporting procedures must consider the level of confirmation and assurance that can be provided by automated tools. For example, theorem proving and model checking technologies rely on analysts being able to assert the properties that are to be checked against the model of the system. However, it can be difficult for members of a development team to consider the wide range of safety properties that must be considered during the application of these tools and techniques. Independent consultants can add a fresh perspective that is often missing from in-house assurance projects.

The final sentence in the previous excerpt requires that the rigour associated with each assurance level is sufficient to justify confidence that the 'software can be operated tolerably safely'. Although this is a relatively short section within the context of the ESARR as a whole, it is arguably one of the most important in the regulatory document. ANSPs must ensure that the techniques and processes that are recommended as minimum requirements

for software development at each assurance level will achieve the necessary confidence in the overall system. Clearly, if these techniques and processes are too onerous then the resulting application may be over-engineered and finite development resources may be diverted from other more critical aspects of a safety-critical system. Conversely, if the minimum requirements for each assurance level are too lax then it is likely that any resultant software will fail to achieve the intended mitigation that was identified in previous risk assessments.

2.5 The ATM service-provider shall ensure, as a minimum, that the Software Safety Assurance System - Uses feedback of ATM software experience to confirm that the Software Safety Assurance System and the assignment of assurance levels is appropriate. For this purpose, the effects resulting from any software malfunction or failure from the ATM operational experience reported according to ESARR 2, shall be assessed in respect of their mapping to ESARR 4.

ESARR 2 deals with the development of Safety Measurement and Improvement Programmes. In an appendix to this document, there is an explicit reference to the need for ANSPs to consider software within the causal classification of incidents and near misses:

> **A-3.3.1 Causes that combined to result in the occurrence shall be classified according to the following high level categories:**
>
> **…**
>
> **ATM service infrastructure/facilities/technical systems**
>
> **- Hardware issues**
> **- Software issues**
> **- Integration issues**
> **- Aerodrome layout and infrastructure**
>
> **…**
>
> **(ESARR2, page 16)**

Clause 2.5 from ESARR6, given above, makes this connection between the two EUROCONTROL regulatory documents. The analysis of adverse events can provide important feedback about whether or not the techniques associated with different software assurance levels are having their intended impact in guarding against software failures. Given that many software systems perform novel and innovative functions, it is critical that ANSPs make best use of the operational experience gained from their software systems. This is also important because resources often have to be specifically allocated to ensure that investigatory personnel have sufficient training to diagnose when software is involved in the causes of a minor accident or near-miss incident. Recall that adverse events are typically associated with equipment under control and the focus of any investigation can be dominated by the behavior of this equipment rather than by interactions with underlying software systems. It can be assumed that any major adverse events will automatically trigger the types of investigation where there will be adequate consideration of software in the potential causes.

It is also important to stress the implications of failure analysis relating to software systems. Given that ESARR6 developed a regulatory framework based around the processes that are used to develop code rather than advocating particular testing regimes, any software failures are likely to be symptomatic of problems with the underlying development processes and not just with individual sections of code. This considerably broadens the scope of

any investigation. For example, a failure to associate sufficient levels of rigor with a software criticality assessment level will affect not just the code that led to an incident but potentially will also affect every other program that was developed using this criticality assessment process.

The previous paragraphs also reiterate the need to integrate the information gleaned from an incident and accident reporting system within the wider tasks of risk assessment and of Safety Management. In other words, the key requirement is not just to gather data but also to make sure that it informs subsequent development and maintenance cycles. Hence as we have seen before there is a close integration between the provisions in ESARR6 and those of ESARR4.

2.6. The ATM service-provider shall ensure, as a minimum, that the Software Safety Assurance System - Provides the same level of confidence, through any means chosen and agreed with the Designated Authority, for developmental and non-developmental ATM software (e.g. Commercial Off The Shelf software, etc) with the same software assurance level.

The use of COTS (Commercial Off The Shelf) software has considerable attractions within many application areas of Air Traffic Management. There are strong justifications for using mass market applications and these are not simply related to the costs associated with acquiring these systems. The increased user-base for these systems can provide accurate data about potential failure rates. Any known problems are often reported and resolved over a relatively short timescale. The large volume of sales often implies higher levels of support and documentation than can be expected for more specialist, safety-related or bespoke software systems. However, there is an obvious risk that the development practices associated with COTS may not meet the requirements for assurance and traceability that we have already met in previous sections of ESARR6. In particular, the commercial sensitivity of these systems makes it unlikely that ANSPs will obtain the source code that can be necessary to perform 'white box' tests that deliberately expose potential weaknesses using a knowledge of the internal implementation.

The previous requirement reinforces the observation that there should be no 'special exemptions' for COTS software and that the same levels of assurance should be expected of code that was developed 'in house' and that which has been developed by other organisations. The integration of COTS has been extensively addressed within the guidance sections of Chapter 7, in Recommendations for ANS Software (SAF.ET1.ST03.1000.GUI-01-00). For example, this document helps explain the reference to alternate assurance methods in ESARR6:

> **"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".

(SAF.ET1.ST03.1000.GUI-01-00, page 78)

As before, interested readers should refer to this extended guidance material to obtain more details about the manner of handling COTS within particular software assurance levels.

5.3 ESARR 6 - Section 3 – Requirements Applying to the Software Assurance Level

Guidance in this section elaborates the requirements applying to the Software Assurance Level from ESARR 6 Section 3 of Obligatory Provisions.

3.1. The ATM service-provider, as a minimum within the Software Safety Assurance System, shall ensure that: - The software assurance level relates the rigour of the software assurances to the criticality of ATM software by using the ESARR 4 severity classification scheme. A minimum five software assurance levels shall be identified to map onto the five severity classes given in ESARR 4. Software assurance level 1 shall indicate the most critical software, to be associated with severity class 1. Software assurance level 5 shall indicate non-safety-related software, to be associated with severity class 5. Intermediate software assurance levels shall be mapped, as a minimum, onto the remaining severity classes in ESARR 4.

ESARR4 identifies a five level severity classification scheme, this is illustrated in the following table and can be summarised as follows:

1. ***Accidents.***

   Examples of the effects on operations include one or more catastrophic accidents, one or more mid-air collisions, one or more collisions on the ground between two aircraft, one or more Controlled Flight Into Terrain, total loss of flight control. In addition there exists no independent source of recovery mechanism, such as surveillance or ATC and/or flight crew procedures can reasonably be expected to prevent the accident(s).

2. ***Serious incidents,***

   Examples of the effects on operations include q large reduction in separation (e.g., a separation of less than half the separation minima), without crew or ATC fully controlling the situation or able to recover from the situation, one or more aircraft deviating from their intended clearance, so that abrupt manoeuvre is required to avoid collision with another aircraft or with terrain (or when an avoidance action would be appropriate).

3. ***Major incidents.***

   Examples of the effects on operations include large reduction (e.g., separation of less than half separation minima) in separation with crew or ATC controlling situation and able to recover the situation. minor reduction (e.g., separation of more than half separation minima) in separation without crew or ATC controlling the situation, hence jeopardising the ability to recover from the situation (without use of collision or terrain avoidance manoeuvres).

4. Significant incidents.

Examples of the effects on operations include q increasing workload of the air traffic controller or aircraft flight crew, or slightly degrading the functional capability of the enabling CNSsystem, minor reduction (e.g., a separation of more than half the separation minima) in separation with crew or ATC controlling the situation and fully able to recover from the situation.

5. No immediate effect on safety.

Examples of the effects on operations include no hazardous condition i.e. no immediate direct or indirect impact on the operations .

(ESARR4, page 15)

The software assurance levels should be mapped onto each of these severity levels in the manner described by clause 3.1.

3.2 The ATM service-provider, as a minimum within the Software Safety Assurance System, shall ensure that: - When allocating a software assurance level to ATM software the software assurance level shall be commensurate with the most adverse effect that software malfunctions or failures may cause, as per ESARR 4, taking into account the risks associated with software malfunctions or failures and the architectural and/or procedural defences.

Architectural and/or procedural defences may be implemented at the ATM systems level that mitigate the adverse effects originating from software malfunctions or failures. Consequently the software assurance level should take this mitigation into account. However, this clause also reiterates the importance of considering the 'most adverse effect'. Very often this involves some consideration of concurrent failures in other related systems and hence there may have to be some appeal to the 'worst plausible consequences'. Clearly determining the nature of the worst consequence can be a subjective process and should be subject to considerable peer review.

The previous regulatory requirement provides further links to previous sections in ESARR6 and the frameworks provided by ESARRs 3 and 4. The consequences of software failure are determined by the risks and the associated hazards that these components are intended to mitigate. Software failure, therefore, leads to the consequences that should already have been considered in the wider risk assessments.

3.3. The ATM service-provider, as a minimum within the Software Safety Assurance System, shall ensure that: - ATM software components that cannot be shown to be independent of one another shall be allocated the software assurance level of the most critical of the dependent components.

ATM software components that are independent from each other may be allocated different assurance levels.

ESSAR 6 describes independent software components in the following terms:

"Those software components which are not rendered inoperative by the same failure condition that causes the hazard".

(ESSAR 6, page 17)

It, therefore, follows that any two software components that can be affected by the same failure condition should not be considered independent. In some senses, the ESSAR definition is relatively weak. Dependencies often exist between software components where a common fault impairs the operation of those components but where the fault does not necessarily lead to a complete failure to operate. The previous excerpt from the regulatory requirement formalises the intuition that where any dependencies exist the different software components should inherit the highest software assurance level of any of the dependent components. If this heuristic were not to be followed then the assurance level might be diluted by the introduction of less critical code into high assurance software.

5.4 ESARR 6 - Section 4 – Requirements Applying to the Software Requirements Validity Assurances

Guidance in this section elaborates the Requirements applying for Software Requirements Validity Assurances from ESARR 6 section 4 of Obligatory Provisions.

4.1 The ATM service-provider, as a minimum, within the Software Safety Assurance System, shall ensure that software requirements - Specify the functional behaviour of the ATM software, timing performances, software resource usage on the target hardware, robustness to abnormal operating conditions, overload tolerance.

As mentioned previously, validation focuses on the value or relevance of a requirement while verification establishes the truth of whether or not a requirement has been satisfied.

It is clearly important that any software specification must consider an adequate range of constraints that collectively characterise the operational behaviour of any code. These characteristics include timing performance, software resource usage on the target hardware, robustness to abnormal operating conditions and overload tolerance. This list from the clause 4.1 is a minimum set of validation requirements, in other words, this information must be specified in order to have a 'valuable' or 'valid' specification.

Some of the concepts used in clause 4.1 deserve further explanation. Timing issues are relatively straightforward and consider a range of scheduling constraints, relative as well as hard real time deadlines. The term 'software resource usage on the target hardware' is more ambiguous than the timing requirements. This refers to a vast range of issues including processor requirements, primary and secondary memory issues, network bandwidth and so on. As with timing issues it is critical to consider these different aspects of resource usage at a level of detail that is likely to yield accurate results. The final reference to overload tolerance and to abnormal operating conditions provides regulatory guidance to consider what might happen if software applications exceeded the resources that are anticipated for ATM safety-related software systems. In addition, abnormal events should normally consider a range of adverse scenarios that can often be triggered by changes in the operational environment.

4.2 The ATM service-provider, as a minimum, within the Software Safety Assurance System, shall ensure that software requirements - Are complete and correct, and are also compliant with the system safety requirements.

As in previous sections, ESARR6 provides some initial guidance for the interpretation of this clause when it defines the completeness and correctness of software requirements in the following terms:

> **"All software requirements correctly state what is required of the software component by the risk assessment and mitigation process and their implementation is demonstrated to the level required by the Software assurance level. Therefore, the software component will remain tolerably safe as required by ESARR 4".**
>
> **(ESARR 6, page 16)**

Previous sections have already argued that completeness and correctness are important concepts when considering the relationships that stretch from system safety requirements, hazard analyses and risk assessments through various stages of software design and testing towards implementation. Hence traceability is critical if ANSPs are to ensure that every requirement derived from a risk assessment is carried through the successive stages of the lifecycle until it is realised in code. We have also discussed the importance that ESARR6 places on the integration of software safety requirements within wider concerns for SYSTEM SAFETY REQUIREMENTS and this is again reiterated in clause 4.2.

5.5 ESARR 6 - Section 5 – Requirements Applying to the Software Verification Assurances

Guidance in this section elaborates the Requirements applying for Software Verification Assurance from ESARR 6 section 5 of Obligatory Provisions.

5.1 The ATM service-provider, as a minimum, within the Software Safety Assurance System, shall ensure that :- The functional behaviour of the ATM software, timing performances, software resource usage on the target hardware, and robustness to abnormal operating conditions, comply with the software requirements.

As mentioned previously, validation focuses on the value or relevance of a requirement while verification establishes the truth of whether or not a requirement has been satisfied. This section of the ESARR6 regulatory requirements extend previous constraints from clause 4.1, which focused on the validation of functional behaviours, to now consider the verification of those behaviours.

Establishing that an implementation or design will meet particular behavioural requirements is non-trivial. For example, the calculation of performance timings creates a host of practical and technical problems that must be addressed during the more detailed development stages. For example, the impact of caching techniques might need to be addressed in order to accurately anticipate task performance on particular target platforms.

Similarly, establishing whether or not software will meet resource usage constraints can involve complex static analysis and a host of more dynamic techniques, including the monitoring of CPU and bus or network utilisation

under a broad range of conditions. The verification of these properties can lead on to further issues of validation, for example, to ensure not just that the software performs in the manner anticipated but also to ensure that any environmental factors used in performance simulation are valid approximations for a broad enough range of likely operational conditions.

5.2 The ATM service-provider, as a minimum, within the Software Safety Assurance System, shall ensure that :- The ATM software is adequately verified by analysis and/or testing and/or equivalent means as agreed with Designated Authority.

The Software Safety Assurance System constructs a framework for associating levels of rigour with different components that reflect the importance of those components in the mitigation of system risks. Hence the emphasis is on applying development processes that are appropriate for the degree of rigour demanded at each level of assurance. This contrasts with previous generations of standards that often focus on acceptance tests as a means of ensuring compliance. The present focus on risk based application of development processes is entirely appropriate for software engineering, given the previous observation that standard testing techniques can only establish the presence of bugs and can never demonstrate their absence. Recall the complexity involved in following every possible execution sequence through even a 20 or 30 line program.

Having reiterated the overall approach embodied in ESARR 6, it is important not to overlook the significance of appropriate testing and analysis techniques for increasing confidence in software quality and reliability. These different approaches form a key component within the various tools that help to demonstrate the appropriate level of rigour at various levels of assurance. Hence they are an important element of the software development process but they are not the central feature as they were in previous generations of standards.

5.3 The ATM service-provider, as a minimum, within the Software Safety Assurance System, shall ensure that :- The verification of the ATM software is correct and complete.

Clause 5.3 extends the correctness and completeness requirements that were previously applied to validation criteria but in this instance relates them to the verification of ATM software. This creates additional concerns for the traceability of key requirements. In previous sections we have considered the manner in which ANSPs must demonstrate that particular sections of code implement the mitigation requirements that are identified from risk assessments. However, there is also a traceability requirement between different levels of verification. In other words, establishing that a particular design will satisfy higher level safety requirements need not guarantee that any software implementation will also meet those requirements. Hence, it is important to show that those same tests can be fulfilled at each successive level of development.

5.6 ESARR 6 - Section 6 – Requirements Applying to the Software Configuration Management Assurances

Guidance in this section elaborates on the requirements applying to the Software Configuration Management Assurances from ESARR 6 section 6 of Obligatory Provisions.

6.1 The ATM service-provider, as a minimum, within the Software Safety Assurance System, shall ensure that - Configuration identification, traceability and status accounting exist such that the software life cycle data can be shown to be under configuration control throughout the ATM software life cycle.

It is important that ANSPs maintain good control over the configuration of the software that implements key services. Previous sections have stressed that the flexibility of programmable systems creates enormous opportunities to adapt safety-critical systems in response to environmental changes or revised operational practices. Similarly, software updates can be implemented, distributed and installed over a relatively short timescale. However, these very benefits create significant problems in terms of project management. It can be difficult to determine the precise version of a program that is running on particular platforms. It is important not to underestimate the importance of even the most basic accounting information. For example, many dozens of hours of staff time can be wasted in tracing incident and bug reports back through software listings if it is unclear which version of a program is actually installed on a system.

The ability to dynamically reconfigure hardware components using dynamic programming techniques also creates the opportunity for significant additional complexity. It is likely that the application of these approaches will grow from their present, rather limited levels. Hence status accounting is a key issue for the support and technical staff who must monitor and maintain safety-critical software. The closing sentence of this clause reiterates the importance of keeping this information up to date both within the initial development life-cycle and beyond into service until decommissioning. In some respects, the documentation of version information is more critical during these subsequent phases when the initial development team may no longer be available to help in the process of software identification, for instance following bug reports.

6.2 The ATM service-provider, as a minimum, within the Software Safety Assurance System, shall ensure that - Problem reporting, tracking and corrective actions exist such that safety related problems associated with the software can be shown to have been mitigated.

This again builds on comments in earlier sections of this guidance document, particularly in reference to ESARR2 within the Safety Measurement and Improvement Programmes and ESARR 4 on Safety Management Systems. The key concern here, as before, is to provide mechanisms and appropriate techniques to feed back operational experience into the maintenance and subsequent development of software systems. It is important to reiterate that this information relates not just to software incident and bug reports. It is equally important to monitor any occurrence of the system level hazards that the software is intended to guard against. If such failures occur then it is likely that the software requirements may have been incomplete or incorrect even though an implementation may have met the constraints that were to be

imposed upon it by previous stages of analysis within the software assurance framework.

6.3 The ATM service-provider, as a minimum, within the Software Safety Assurance System, shall ensure that - Retrieval and release procedures exist such that the software life cycle data can be regenerated and delivered throughout the ATM software life cycle.

Most of the previous requirements within ESARR 6 create processes that generate documentation. It is impossible, for instance, to demonstrate the traceability that was advocated in the Software Assurance Framework without having sufficient documentation to support comparisons between the various activities involved in risk assessment, mitigation, software design and implementation. It is clearly important for ANSPs to be able to manage and maintain the mass of documentation that can be generated by these different activities. Similarly, there is little prospect of ensuring consistency between different teams or development projects if key documents cannot easily be shared, for instance to show that similar hazards are related to similar risks in different development projects.

At the same time, designated national entities need to be able to monitor the activities of ANSPs to ensure that they have implemented the many different safety related process that are advocated in each of the ESARR documents. In order to do this, they must be able to access the products of those processes through the kinds of retrieval and document sharing systems mentioned in clause 6.3. Several ANSPs have begun to develop knowledge management tools to meet these and similar regulatory requirements.

5.7 ESARR 6 - Section 7 – Requirements Applying to the Software Requirements Traceability Assurances

Guidance in this section elaborates on the requirements applying to the Software Traceability Assurances from ESARR 6 section 7 of Obligatory Provisions.

7.1 The ATM service-provider, as a minimum, within the Software Safety Assurance System, shall ensure that :- Each software requirement is traced to the same level of design at which its satisfaction is demonstrated.

This clause refines some of the comments made in earlier sections of ESARR 6. In previous sections, this guidance has argued that traceability requirements need to be followed through to the code that implements them. This does not imply, however, that they should be traced to individual lines of code. For example, static and dynamic verification techniques might be used to demonstrate that key properties hold over high-level components. This is likely to be the case when these requirements are discharged by COTS applications. In such circumstances, it will typically not be possible to trace a system safety constraint through to the individual lines of code. Such limitations help to underline previous caveats about the problems of establishing complete traceability using 'non-developmental' software.

7.2 The ATM service-provider, as a minimum, within the Software Safety Assurance System, shall ensure that :- Each software requirement, at each level in the design at which its satisfaction is demonstrated, is traced to a system requirement.

ESARR 6 builds upon an integrated approach to the development of safety related systems in Air Traffic Management. Previous sections have described the overall philosophy that motivates and guides the individual requirements within the regulatory document. The key stages of risk assessment in ESARR3 and Safety Management in ESARR4 help to identify high level objectives for the development of software. Hence, it follows that if there are key software requirements that are not strongly related to critical system level concerns then this is likely to indicate omissions in the initial risk assessments that help to drive software development.

5.8 ESARR 6 - Section 8 – Applicability

8.1 This safety regulatory requirement shall apply to civil and military ATM service providers who have the responsibility for the management of safety in ground-based ATM systems and other supporting services (including CNS) under their managerial control.

This clause clarifies the scope of ESARR 6 and stresses that military as well as civilian systems should be considered. This raises important issues when, for example, military systems interface with the software that controls civilian flights. Previous sections have described how assurance levels should be propagated between different components. If dependencies exist between two or more components then the level of assurance for each individual element should be at the highest level of any component. Hence it may be necessary to propagate assurance levels between military and civilian software systems in order to ensure that each reaches the appropriate level of safety assurance. The key issue here in terms of the scope of ESARR 6 is that it views military and civilian applications within the wider context of total air navigation systems safety. This has implications both for the management and the technical implementation of the regulatory requirements.

8.2 The obligatory provisions of this ESARR shall be enacted as minimum national safety regulatory requirements.

The key implication of this requirement is that designated authorities and ANSPs must work together to implement the requirements of ESARR 6 within their national systems. However, as mentioned in previous sections, it is important not to underestimate the importance of international cooperation and exchange in the development of software safety assurance methods. For example, the low frequency of many types of safety-related software failures creates a pressing need to share information across national boundaries when any failures do occur. Similarly, the highly technical nature of some of the validation and verification techniques mentioned in previous sections will create training and competency requirements that can be reinforced by international collaboration via mechanisms such as those provided within EUROCONTROL.

5.9 ESARR 6 - Section 9 – Implementation

The provisions of this requirement are to become effective within three years from the date of approval by the EUROCONTROL Commission

The consequences of this requirement are self evidence. The core components of ESARR 6 are effective from 2007.

[[CWJ – we might need to mentioned something about the phased implementation of the regulatory requirements? Or this is probably the sort of topic that we could address in the Additional Guidance of section 6?]]

5.10 ESARR 6 - Section 10 – Exemptions

None

6. ADDITIONAL GUIDANCE

[[CWJ: This section needs to be discussed with everyone – it is mentioned as TBD in the original draft from Tony – I can write it but need to know a little more about the proposed content.

Excerpt from the opening sections: "The document includes a Section 6 to provide guidance considered necessary to achieve the stated safety objectives. This section includes all applicable mandatory requirements (expressed using the word "shall"), including those relating to implementation".]]

7. CONCLUSIONS

The General Requirement identifies the five minimum assurances to be considered by ATM service-providers in order to meet the safety objective i.e. – *risks associated with operating ATM software have been reduced to a tolerable level -*

To ensure that the five assurances are achieved ATM service-provider is required to detail his SMS by implementing a Software Safety Assurance System.

It is then the responsibility of the ATM safety regulator to ensure the adequate safety oversight of the service-provider SSAS.

In the consideration of SSAS the following aspects are required;

- Allocation of the Software Assurances level,
- Software Requirements Validity Assurances,
- Software Verification Assurances,
- Software Configuration Management Assurances,
- Software Requirements Traceability Assurances.

Regulatory processes shall ensure that these elements, or equivalent ones (e.g. for COTS) , are properly considered throughout the complete safety management (Safety Software Assurance system) documented system arising from high level safety policy statements.

8. APPENDIX A

Glossary – Terms and Definitions

Definitions for specific terms used in this document are given in the EUROCONTROL Safety Regulatory Requirements – Software in ATM Systems (ESARR 6), and repeated for ease of reference in this appendix.

TERM	DEFINITION
Assessment	An evaluation based on engineering, operational judgement and/or analysis methods.
ATM	The aggregation of ground based (comprising variously ATS, ASM, ATFM) and airborne functions required to ensure the safe and efficient movement of aircraft during all appropriate phases of operations.
ATM Equipment approved for operational use	All engineering systems, facilities or devices that have been used either by airspace users (e.g. ground navigation facilities) directly, or are used in the provision of operational air traffic management services.
ATM Service	A service for the purpose of ATM.
ATM Service-Provider	An organisation responsible and authorised to provide ATM service(s).
ATM Software	Software used in ATM Environment. *See later the definition for software.*
CNS	Communication, Navigation and Surveillance.
Configuration data	Data that configures a generic software system to a particular instance of its use (for example, data that adapts a flight data processing system to a particular airspace, by setting the positions of airways, reporting points, navigation aids, airports and other elements important to air navigation).
Hazard	Any condition, event, or circumstance which could induce an accident.
Independent software components	Those software components which are not rendered inoperative by the same failure condition that causes the hazard.

TERM	DEFINITION
Mitigation or Risk Mitigation	Steps taken to control or prevent a hazard from causing harm and reduce risk to a tolerable or acceptable level.
Operating Software	For the purpose of ESARR 6 it is understood the software used in ATM equipment approved for operational use. *See above the definition for ATM Equipment approved for operational use.*
Risk	The combination of the overall probability, or frequency of occurrence of a harmful effect induced by a hazard and the severity of that effect.
Risk Assessment	Assessment to establish that the achieved or perceived risk is acceptable or tolerable.
Risk Mitigation	See mitigation.
Safety	Freedom from unacceptable risk.
Safety Achievement	The result of processes and/or methods applied to attain acceptable or tolerable safety.
Safety Assurance	All planned and systematic actions necessary to provide adequate confidence that a product, a service, an organisation or a system achieves acceptable or tolerable safety.
Safety Management System (SMS)	A systematic and explicit approach defining the activities by which safety management is undertaken by an organisation in order to achieve acceptable or tolerable safety.
Safety Regulatory Requirement	The formal stipulation by the regulator of a safety related specification which, if complied with, will lead to acknowledgement of safety competence in that respect.

TERM	DEFINITION
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 failure	The inability of a program to perform a required function correctly.
Software life cycle 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 Requirements	The specifications, if met, will ensure that ATM software performs safely and according to operational need.
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.

(PAGE INTENTIONALLY LEFT BLANK)

9. APPENDIX B

Applicability of ESARR 6

The Requirement includes a Section TBD, '*Applicability*' to specify the scope of applicability of its provisions in term of categories of organisations that are subject to the requirements. The scope of ESARR 6 is the same as of ESARR 3 i.e. the Software Safety Assurance System as part of the Safety Management System is to be implemented by those organisations determined in Section TBD. This appendix is intended to provide guidance on these aspects.

B1 Applicability to EUROCONTROL Member States

The Safety Regulation Commission (SRC) is responsible for the development of harmonised safety regulatory objectives and requirements for the ATM System, which will be implemented and enforced by Member States after being approved by EUROCONTROL.

The requirements are known as ESARR (EUROCONTROL Safety Regulatory Requirements). In practical terms, each ESARR is developed by the SRC, approved by the EUROCONTROL Permanent Commission through the Provisional Council, and implemented and enforced by the Member States.

Member States are bound by decisions taken under either the current or revised EUROCONTROL Convention, and consequently have to implement and enforce within their national legal order the safety regulatory requirements contained in such decisions.

This also concerns those ESARR that apply to ATM service-providers and/or Designated Authorities and/or individuals, such as ESARR 3, ESARR 5 and ESARR 6. Member States will have to ensure through appropriate safety oversight that ATM community meets these requirements.

B2 Applicability to ATM providers

ESARR 6 is applicable to all providers of ATM services that fall under the jurisdiction of the national ATM safety regulatory body.

Accordingly, the implementation concerns all organisations providing not only ATS services (encompassing ATC, FIS, and alerting and advisory services), but also other ATM services such as Air Traffic Flow Management (ATFM) and Airspace Management (ASM). That scope is consistent with ICAO and EUROCONTROL definitions for Air Traffic Management.

(Figure B.1 – Applicability of ESARR 6 to ATM Service-Providers)

NOTE: Applicability of ESARR 6 is the same as for ESARR 3.

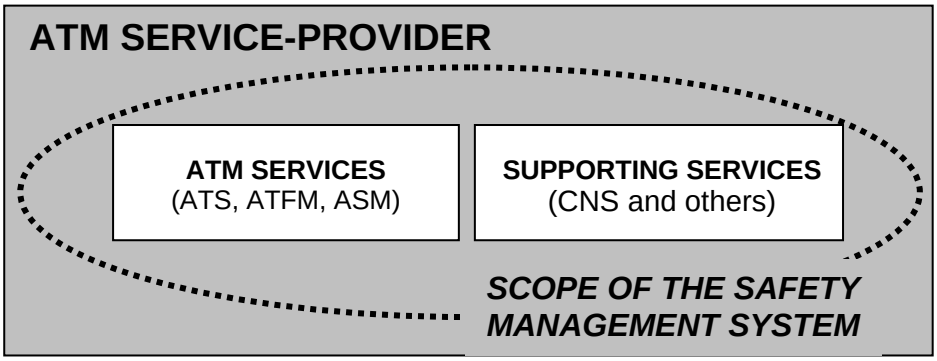
Situations exist where different organisations provide these services separately. Requirements will apply to all of them when those functions uses operational software.

ATM services can be provided simultaneously by different organisations operating within specific geographical regions or having responsibilities for parts of the navigable airspace associated with a flight phase. For instance, we may conceive situations where a national organisation is responsible for en-route ATM, while TWR or AFIS services are delivered by organisations owning local airports. Again, we may say that all those organisations will have to meet ESARR 6 requirements.

B3 Applicability to ATM safety regulators (Designated Authority)

B4 The SMS Scope

The SMS operated by each ATM service-provider will have to cover not only its ATM services, but also any supporting service (including CNS functions and services) which are under the managerial control of the organisation. As such the Software Safety Assurance System should be a distinct component ensuring safety assurances when operating ATM software.



(Figure B.2 – Scope of the SSAS required by ESARR 6)

Supporting services include systems, services and arrangements, including Communication, Navigation and Surveillance services, which support the provision of an ATM service. Any supporting service under the managerial control of the organisation has to be covered by the SSAS.

Supporting services outside the managerial control of the organisation should be considered as external inputs and addressed in accordance with the External Services requirement (ESARR 3, Section 5.2.6).