‘AMC and GM to Part-ATM/ANS.OR — Issue 1, Amendment 1’

Annex III to Decision 2017/001/R is amended as follows:

The text of the amendment is arranged to show deleted text, new or amended text as shown below:

(a) deleted text is marked with ~~strikethrough~~;

(b) new or amended text is highlighted in blue;

(c) an ellipsis (…) indicates that the remaining text is unchanged in front of or following the reflected amendment.

GM2 ATM/ANS.OR.A.045(a) Changes to a functional system

NOTIFICATION — SOFTWARE CRITICALITY

Depending on the complexity of the change to the functional system and the criticality of the software, the depth of the evaluation may vary. The service provider should coordinate as soon as possible with the competent authority in order to define a software oversight strategy as part of the change review activities, if a decision for change review is taken.

GM1 ATM/ANS.OR.A.050 Facilitation and cooperation

AUDITS — SOFTWARE ASSURANCE PROCESSES BY THE COMPETENT AUTHORITY

(a) The assessment of an effective application of the documented software assurance processes may necessitate a technical evaluation of the evidence and arguments produced for the software assurance by the competent authority when reviewing a notified change. In this context, the service provider should ensure access to the configuration management system for the competent authority, which may need to verify:

(1) the consistency of all the evidence; and

(2) the fact that all the evidence is derived from a known version of the software (i.e. all evidence and arguments are actually available and can be traced without ambiguity to the executable version).

(b) The service provider should:

(1) anticipate the possibility for on-site audits or inspections by the competent authority; and

(2) when evidence and arguments are developed by contracted organisations, include the corresponding rights of the competent authority to assess said organisations during on-site audits or inspections.

AMC5 ATM/ANS.OR.C.005(a)(2) Safety support assessment and assurance of changes to the functional system

ASSURANCE — SOFTWARE

(a) When a change to a functional system includes the introduction of new software or modifications to existing software, the service provider should ensure the existence of documented software assurance processes necessary to produce evidence and arguments

that demonstrate that the software behaves as intended (software requirements), with a level of confidence consistent with the needs of the required application.

(b) The service provider should use feedback of software experience to confirm that the software assurance processes are effective and, when used, the allocated software assurance levels (SWALs) and the rigour of the assurances are appropriate. For that purpose, the effects from software malfunctions (i.e. the inability of a programme to perform a required function correctly) or failures (i.e. the inability of a programme to perform a required function) reported according to the relevant requirements on reporting and assessment of service occurrences should be assessed in comparison with the effects identified for the system concerned as per the service specification demonstration.

AMC6 ATM/ANS.OR.C.005(a)(2) Safety support assessment and assurance of changes to the functional system

ASSURANCE — SOFTWARE ASSURANCE PROCESSES

(a) The software assurance processes should provide evidence and arguments that they, as a minimum, demonstrate the following:

(1) The software requirements correctly state what is required by the software, in order to meet the service and safety support requirements, as identified by the safety support assessment (AMC2.ATM/ANS.OR.C005(a)(2)). For that purpose, the software requirements should:

(i) be correct, complete and compliant with the upper level requirements; and

(ii) specify the functional behaviour, in nominal and downgraded modes, timing performances, capacity, accuracy, resource usage on the target hardware, robustness to abnormal operating conditions and overload tolerance, as appropriate, of the software.

(2) The traceability is addressed in respect of all software requirements as follows:

(i) Each software requirement should be traced to the same level of design at which its satisfaction is demonstrated.

(ii) Each software requirement allocated to a component should either be traced to an upper level requirement or its need should be justified and assessed that it does not affect the satisfaction of the safety support requirements allocated to the component.

(3) The software implementation does not contain functions that adversely affect the satisfaction of the service specification.

(4) The functional behaviour, timing performances, capacity, accuracy, resource usage on the target hardware, robustness to abnormal operating conditions and overload tolerance, of the implemented software comply with the software requirements.

(5) The software verification is correct and complete, and is performed by analysis and/or testing and/or equivalent means, as agreed with the competent authority.

(b) The evidence and arguments produced by the software assurance processes should be derived from:

(1) a known executable version of the software;

(2) a known range of configuration data; and

(3) a known set of software items and descriptions, including specifications, that have been used in the production of that version, or can be justified as applicable to that version.

(c) The software assurance processes should determine the rigour to which the evidence and arguments are produced.

(d) The software assurance processes should include the necessary activities to ensure that the software life cycle data can be shown to be under configuration control throughout the software life cycle, including the possible evolutions due to changes or problems' corrections. They should include, as a minimum:

(1) configuration identification, traceability and status accounting activities, including archiving procedures;

(2) problem reporting, tracking and corrective actions management; and

(3) retrieval and release procedures.

(e) The software assurance processes should also cover the particularities of specific types of software such as commercial-off-the-shelf (COTS), non-developmental software and previously developed software where generic assurance processes cannot be applied. The software assurance processes should include other means to give sufficient confidence that the software meets the service and safety support requirements. If sufficient assurance cannot be provided, complementary mitigation means aiming at decreasing the impact of specific failure modes of this type of software, should be applied. This may include but is not limited to:

(1) software and/or system architectural considerations;

(2) existing service level experience; and

(3) monitoring.

GM2 ATM/ANS.OR.C.005(a)(2) Safety support assessment and assurance of changes to the functional system

ASSURANCE LEVELS

(a) The use of assurance level concepts, e.g. design assurance levels (DALs), software assurance levels (SWALs), hardware assurance levels (HWALs), can be helpful in generating an appropriate and sufficient body of evidence to help establish the required confidence in the argument.

(b) The term 'software assurance level (SWAL)' is understood to be the level of rigour of the software assurances throughout the software lifecycle. In this context, the software life cycle is understood to be:

(1) an ordered collection of processes determined by an organisation to be sufficient and adequate to produce a software item;

(2) the period of the time that begins with the decision to produce or modify a software item and ends when the item is retired from service.

GM1 to AMC6 ATM/ANS.OR.C.005(a)(2) Safety support assessment and assurance of changes to the functional system

ASSURANCE — SOFTWARE ASSURANCE PROCESS

(a) The term 'correct and complete software verification' is understood to be all software safety requirements, which 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.

(b) The term 'software timing performances' is understood to be the time allowed for the software to respond to given inputs or to periodic events, and/or the performance of the software in terms of transactions or messages handled per unit time.

(c) The term 'software capacity' is understood to be the ability of the software to handle a given amount of data flow.

(d) The term 'software accuracy' is understood to be the required precision of the computed results.

(e) The term 'software resource usage' is understood to be the amount of resources within the computer system that can be used by the application software.

(f) The term 'software robustness' is understood to be the behaviour of the software in the event of unexpected inputs, hardware faults and power supply interruptions, either in the computer system itself or in connected devices.

(g) The term 'overload tolerance' is understood to be the behaviour of the system in the event of, and in particular its tolerance to, inputs occurring at a greater rate than expected during normal operation of the system.

(h) The term 'software life cycle data' is understood to be the 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 item.

(i) The term 'COTS' is understood to be a commercially available application sold by vendors through public catalogue listings and not intended to be customised or enhanced.

GM2 to AMC6 ATM/ANS.OR.C.005(a)(2) Safety support assessment and assurance of changes to the functional system

ASSURANCE — SOFTWARE ASSURANCE LEVELS

(a) The assurance required by AMC6 ATM/ANS.OR.C.005(a)(2) can be provided with different levels of confidence depending on the rigour to which the evidence and arguments are produced. Whereas, for air traffic services (ATS) providers, the use of the SWAL concept can be helpful to provide an explicit link between the criticality of the software and the rigour of the assurance, for service providers other than ATS providers, the use of the SWAL concept may not be relevant considering that non-ATS providers may not be aware of the safety aspects of the ATS provider using their services. However, considering that the safety support assessment will be based on the evidence and arguments generated by the software assurance processes and that the safety support assessment will support a safety assessment, it is foreseen that, in many changes, the software assurance evidence and arguments will have to demonstrate a certain level of confidence and therefore will have to show compliance with the SWAL allocated by the ATS provider.

(b) The use of multiple SWALs would also allow the possibility of managing several criticalities of the different software components within the system (with partitioning or other architectural strategies) by the same set of software assurance processes. When the software assurance processes employ several SWALs, they should define for each SWAL the rigour of the assurances to achieve compliance with the objectives set out in AMC6 ATM/ANS.OR.C.005(a)(2). As a minimum:

(1) the rigour should increase as the criticality of the service supported by the software solution increases; and

(2) the variation in rigour of the evidence and arguments per SWAL should include a classification of the activities and objectives according to the following criteria:

(i) required to be achieved with independence, i.e. the verification process activities are performed by a person (or persons) other than the developer of the item being verified;

(ii) required to be achieved; and

(iii) not required.

GM3 to AMC6 ATM/ANS.OR.C.005(a)(2) Safety support assessment and assurance of changes to the functional system

ASSURANCE — SOFTWARE ASSURANCE LEVELS ALLOCATION

The process to allocate a SWAL to a software consistently with its foreseen criticality, as identified by the safety support assessment and requirements, should consider the following elements:

(a) The SWAL allocation should relate the rigour of the software assurances to the foreseen criticality of the software.

(b) The allocated SWAL should be commensurate with the worst credible effect that software malfunctions (i.e. the inability of a programme to perform a required function correctly) or failures (i.e. the inability of a programme to perform a required function) may cause, as assessed by the ATS provider that is planning to make use of the non-ATS services.

(c) The software components that cannot be shown to be independent of one another should be allocated to the SWAL of the most critical of the dependent components. In this context, the term 'software components' is understood to be a building block that can be fitted or connected together with other reusable blocks of software to combine and create a custom software application, and 'independent software components' are those software components which are not rendered inoperative by the same failure condition.

(d) The allocated SWALs should be consistent with the levels defined in the software assurance processes.

GM4 to AMC6 ATM/ANS.OR.C.005(a)(2) Safety support assessment and assurance of changes to the functional system

ASSURANCE — EXAMPLES OF EXISTING INDUSTRIAL STANDARDS

(a) The service provider is responsible for the definition of the software assurance processes. In this definition of processes, the service provider may consider the guidance material contained in existing industrial standards for the software assurance considerations of software. It should be considered that not all standards address all aspects required and the service provider may need to define additional software assurance processes. The guidance material typically includes:

(1) objectives of the software life cycle processes;

(2) activities for satisfaction of those objectives;

(3) descriptions of the evidence, in the form of software life cycle data, that indicates that the objectives have been satisfied;

(4) variations according to the SWAL, to accommodate the different levels of rigour of the software assurances; and

(5) particular aspects (e.g. previously developed software) that may be applicable to certain applications.

(b) The following table presents some of the existing industrial standards (at the latest available issue) used by the stakeholders:

Document title	Reference	Date
Software Integrity Assurance Considerations for Communication, Navigation, Surveillance and Air Traffic Management (CNS/ATM) Systems	EUROCAE ED-109A/ RTCA DO-278A	January 2012
Guidelines for ANS Software Safety Assurance	EUROCAE ED-153	August 2009
Standards for Processing Aeronautical Data (only for AIS providers)	EUROCAE ED-76A/ RTCA DO-200B	June 2015
Functional safety of electrical/electronic/programmable electronic safety-related systems – Part 3: Software requirements	IEC 61508 – Part 3	April 2010
Software Considerations in Airborne Systems and Equipment Certification	EUROCAE ED-12C/ RTCA DO-178C	January 2012

EUROCAE ED-109A/RTCA DO-278A and EUROCAE ED-12C/RTCA DO-178C make reference to some external documents (supplements), which are integral part of the standard for the use of some particular technologies and development techniques. The supplements are the following:

(1) Formal Methods Supplement to ED-12C and ED-109A (EUROCAE ED-216/RTCA DO-333)

(2) Object-Oriented Technology and related Techniques Supplement to ED-12C and ED-109A (EUROCAE ED-217/RTCA DO-332)

(3) Model-Based Development and Verification Supplement to ED-12C and ED-109A (EUROCAE ED-218/RTCA DO-331)

When tools are used during the software development lifecycle, EUROCAE ED-215/RTCA DO-330 'Software Tool Qualification Considerations' may be considered in addition to EUROCAE ED-12C RTCA/DO-178C and EUROCAE ED-109A/RTCA DO-278A.

(c) The definition of the software assurance processes may be based on one of these industrial standards, without combining provisions from different standards as far as the consistency and validation of each of the industrial standards have only been performed at individual level by each specific standardisation group.