'AMC and GM to Part-ATS — Issue 1, Amendment 1'

Annex IV 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.

AMC3 ATS.OR.205(a)(2) Safety 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 ATS 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 criticality of the required application.

(b) The ATS provider should use the software experience gained 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 a software malfunction (i.e. the inability of a programme to perform a required function correctly) or failure (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 severity classification scheme.

AMC4 ATS.OR.205(a)(2) Safety 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 upper level requirements, including the allocated system safety requirements as identified by the safety assessment of changes to the functional system (AMC2.ATS.OR.205(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 requirements allocated to the component.

(3) The software implementation does not contain functions that adversely affect safety.

(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 COTS, non-development 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 safety objectives and requirements, as identified by the safety risk assessment and mitigation processes. 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.

GM1 to AMC4 ATS.OR.205(a)(2) Safety assessment and assurance of changes to the functional system

ASSURANCE — SOFTWARE ASSURANCE PROCESS

In reference to the terms 'correct and complete software verification', 'software timing performances', 'software capacity', 'software accuracy', 'software resource usage', 'software robustness', 'overload tolerance', 'software life cycle data' and 'COTS', please refer to GM1 to AMC6 ATM/ANS.OR.C.005(a)(2) 'Safety support assessment and assurance of changes to the functional system'.

GM2 to AMC4 ATS.OR.205(a)(2) Safety assessment and assurance of changes to the functional system

ASSURANCE — SOFTWARE ASSURANCE LEVELS

(a) The assurance required by AMC4 ATS.OR.205(a)(2) can be provided with a level of confidence consistent with the criticality of the software in order to generate an appropriate and sufficient body of evidence to help to establish the required confidence in the argument.

(b) 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.

(c) 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 on several SWALs, they should define for each SWAL the rigour of the assurances to achieve compliance with the objectives set out in AMC4 ATS.OR.205(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 AMC4 ATS.OR.205(a)(2) Safety 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 risk assessment and mitigation process, should consider the following elements:

(a) The allocated SWAL should relate the rigour of the software assurances to the foreseen criticality of the software by using the combination of the used severity classification scheme with the likelihood of occurrence of a certain adverse effect.

(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. It should, in particular, take into account the risks associated with software malfunctions or failures and the architecture and/or procedural defences.

(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 of the ATS provider and of the non-ATS provider(s), when the safety case is based on the evidence presented in the corresponding safety support case(s).

GM4 to AMC4 ATS.OR.205(a)(2) Safety 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
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.

GM5 to AMC4 ATS.OR.205(a)(2) Safety assessment and assurance of changes to the functional system

ASSURANCE — SWAL COORDINATION

(a) Within the scope of this Regulation, only the ATS provider can identify hazards, assess the associated risks and mitigate or propose mitigating measures where necessary. This requirement is also applicable to software assurance evidence which may include information on the mitigation measures established to address software failures or unintended behaviours.

(b) ATS and non-ATS providers may rely on different sets of software assurance processes and, if applicable, different sets of SWALs.

(c) For a particular change to the functional system, the safety assessment performed by the ATS provider, and documented in the safety case, may rely on evidence associated with the services provided by a non-ATS provider, as documented in its corresponding safety support case. It should as a minimum demonstrate that the rigour of the assurances produced by the non-ATS provider within the safety support case provides the adequate level of confidence for the purpose of the ATS safety demonstration in the safety case.

(d) If SWALs are used, the ATS provider should evaluate the adequacy of the SWALs defined in the software assurance processes of the non-ATS providers and the consistency of the allocated SWALs for the parts of the functional system affected by the change at the non-ATS provider.