EASA	NOTIFICATION OF A PROPOSAL TO ISSUE A CERTIFICATION MEMORANDUM
	**EASA Proposed CM No.:** **EASA CM - SWAEH – 002 Issue: 02** **Issue Date: 22nd of October 2013** **Issued by: Safety, Software & Airborne Electronic Hardware section** **Approved by: Head of Certification Experts Department** **Regulatory Requirement(s): CS 25.1309 for Large Aeroplanes, CS 23.1309 for Small Aeroplanes, CS 27.1309 for Small Rotorcraft, CS 29.1309 for Large Rotorcraft, CS E-50 (d, f) for engines, CS-P, CS-APU and CS-ETSO.**

In accordance with the EASA Certification Memorandum procedural guideline, the European Aviation Safety Agency proposes to revise the EASA Certification Memorandum (CM) on the subject identified below.

All interested persons may send their comments, referencing the EASA Proposed CM Number above, to the e-mail address specified in the "Remarks" section, prior to the indicated closing date for consultation.

EASA Certification Memoranda clarify the European Aviation Safety Agency's general course of action on specific certification items. They are intended to provide guidance on a particular subject and, as non-binding material, may provide complementary information and guidance for compliance demonstration with current standards. Certification Memoranda are provided for information purposes only and must not be misconstrued as formally adopted Acceptable Means of Compliance (AMC) or as Guidance Material (GM). Certification Memoranda are not intended to introduce new certification requirements or to modify existing certification requirements and do not constitute any legal obligation.

EASA Certification Memoranda are living documents into which either additional criteria or additional issues can be incorporated as soon as a need is identified by EASA.

Subject

Software Aspects of Certification

Log of Issues

Issue	Issue date	Change description
01	11.08.2011	First issue.
01 rev 1.	14.12.2011	First issue, first revision. Editorial corrections in sections 1, 2, 4, 9, 10, 12, 16, 20, 21, 22, 23, 24 and 26.
02	22.10.2013	Second issue (migration of CM-SWCEH-002 Issue 01 Rev.1) for projects using ED-12C / DO-178C. Coordination between AMC 20-115C and AC 20-115C is ensured with this CM. The summary of the changes below explains how and why the sections of CM SW issue 1 have been removed or changed: Issue 1 Section 1 – Introduction - has been changed and issue 2 clarifies that the SW CM is used to ensure coordination with FAA AC 20-115C and provide all Eurocae ED-12C/DO-178C and supplements references. Issue 1 Section 2 – Background – has been changed and issue 2 provides the items of FAA AC 20-115C that are not in AMC 20-115C; 3 objectives are defined with respect to coordination between AC/AMC 20-115C. Issue 1 Section 3 – EASA Certification Policy – has been changed and issue 2 introduces ED-12C/DO-178C, the related ED-215/DO-330 and ED-94C/DO-248C documents and the related supplements. Issue 1 Section 4 – Guidelines for the software review process – has been changed and issue 2 contains only the objective that the applicant needs to define its SW review process; the EASA software review process has been transferred to the EASA Certification Handbook and is available under request. Issue 1 Section 5 – Organisation, role and level of involvement of EASA and applicants in software projects - has been changed and issue 2 contains only the objective that the applicant needs to define its LOI; the EASA LOI definition has been transferred to the EASA Certification Handbook and is available under request. Issue 1 section 6 – Reserved – has been removed in Issue 2. Issue 1 section 7 – Guidelines for the approval of field loadable software – has been removed in Issue 2 as the coordination with FAA SW policies is achieved in Issue 2 section 2.2. Issue 1 section 8 – Reserved - has been removed in Issue 2. Issue 1 section 9 – Guidelines for the Approval of Airborne Systems and Equipment Containing User modifiable Software – has been removed in Issue 2 as the coordination with FAA SW policies is achieved in Issue 2 section 2.2. Issue 1 section 10 – Guidelines for Applying the ED-12B / DO-178B Level D Criteria to Previously-developed Software – has been removed in Issue 2 as the coordination with FAA SW policies is achieved in Issue 2 section 2.2. Issue 1 section 11 – Guidelines for the Qualification of

Issue	Issue date	Change description
		Software Tools using ED-12B/DO-178B – has been removed in Issue 2 as the coordination with the FAA SW policies is achieved in Issue 2 section 2.2. Issue 1 section 12 – Guidelines for the Certification of Software in Legacy Systems using ED-12B/DO-178B – has been removed in Issue 2 as the coordination with the FAA SW policies is achieved in Issue 2 section 2.2. Issue 1 section 13 – Oversight of Software Change Impact Analyses used to Classify Software Changes as Major or Minor – has been kept but is Issue 2 section 6. Issue 1 section 14 – Guidelines for Approving Reused Software Life Cycle Data – has been removed in Issue 2 as the coordination with the FAA SW policies is achieved in Issue 2 section 2.2. Issue 1 section 15 – Properly Overseeing Suppliers – has been removed in Issue 2 as the coordination with the FAA SW policies is achieved in Issue 2 section 2.2 and the subject is now covered in multiple places of ED-12C/DO-178C. Issue 1 section 16 – Management of Problem Reports – has been clarified in ED-94C/DO-178C (DP#9) but a clarification is made in Issue 2 section 7. Issue 1 section 17 – Embedded Software Configuration Files – has been removed as the subject is now covered by ED-12C/DO-178C in multiple places under the name "Parameter Data Files – PDI". Issue 1 section 18 – Managing the Software Development and Verification Environment – has been removed in Issue 2 as the coordination with the FAA SW policies is achieved in Issue 2 section 2.2. Issue 1 section 19 – The use of Object Oriented Techniques at the Design or Source Code Level – has been removed in Issue 2 as the subject is covered by the ED-217/DO-332. Issue 1 section 20 – The Use of (OCC) Object Code Coverage for Equivalence to Modified Condition Decision Coverage (MCDC) – has been removed in Issue 2 as the subject has been clarified in ED-12C/DO-178C. Issue 1 section 21 – Merging High-level and Low-level Requirements – has been removed in Issue 2 as the subject has been clarified in ED-94C/DO-248C (FAQ#81). Issue 1 section 22 – Clarification of Structural Coverage Analyses of Data Coupling and Control Coupling – has been kept but is now in Issue 2 section 8. Issue 1 section 23 – The Validation and Verification of Model-based Software Requirements and Designs – has been removed in Issue 2 as the subject is covered by ED-218/DO-331. Issue 1 section 24 – The Use of Pseudocode as Low-level Requirements – has been removed in Issue 2 as the subject has been clarified in ED-94C/DO-248C (FAQ#82). Issue 1 section 25 – Stack Overflows – has been kept but is now in Issue 2 section 8.

Table of Contents

1 INTRODUCTION

1.1 PURPOSE AND SCOPE

EASA AMC 20-115C was published on 20.09.2013 and recognised that the EUROCAE ED-12C standard and its related documents and supplements (referenced in section 1.2 below) constitute an acceptable means of compliance for software (SW) aspects of certification.

The purpose of this Certification Memorandum is to provide specific clarification and additional guidance material to applicants on various aspects complementary to AMC 20-115C.

This Certification Memorandum is also used to ensure final coordination between AMC 20-115C and FAA AC 20-115C.

This issue of the EASA Software Certification Memorandum only applies to projects for which AMC 20-115C is applicable.

1.2 REGULATORY REFERENCES

It is intended that the following reference materials be used in conjunction with this Certification Memorandum:

Reference	Title	Code	Issue	Date
ED-12B / DO-178	Software Considerations In Airborne Systems and Equipment Certification	EUROCAE ED-12B RTCA DO-178B	B	December 1992
ED-12C / DO-178	Software Considerations In Airborne Systems and Equipment Certification	EUROCAE ED-12C RTCA DO-178C	C	December 2011
ED-94C / DO-248	Supporting Information for DO-178C and DO-278A	EUROCAE ED-94C RTCA DO-248C	C	December 2011
ED-215 / DO-330	Software Tool Qualification Considerations.	EUROCAE ED-215 RTCA DO-330	-	December 2011
ED-216 / DO-333	Formal Methods Supplement to ED-12C and ED-109A.	EUROCAE ED-216 RTCA DO-333	-	December 2011
ED-217 / DO-332	Object-Oriented Technology and Related Techniques Supplement to ED-12C and ED-109A.	EUROCAE ED-217 RTCA DO-332	-	December 2011
ED-218 / DO-331	Model-Based Development and Verification Supplement to ED-12C and ED-109A.	EUROCAE ED-218 RTCA DO-331	-	December 2011
AMC 20-115	Software considerations for	AMC-20	C	September

	certification of airborne systems and equipment			2013
AC 20-115	Airborne Software Assurance	AC 20-115C	C	July 2013
UG.TC.00002	EASA Certification Handbook	UG.TC.00002	002	April 2013

1.3 ABBREVIATIONS

The following abbreviations are used in this Certification Memorandum:

Abbreviation	Meaning
AC	Advisory Circular
AMC	Acceptable Means of Compliance
ATA	Air Transport Association
CAST	Certification Authorities Software Team
CEH	Complex Electronic Hardware
CM	Certification Memorandum
CRI	Certification Review Item
CS	Certification Specification(s)
DOA	Design Organisation Approval
EASA	European Aviation Safety Agency
ETSO	European Technical Standard Order
FAA	Federal Aviation Administration
FAQ	Frequently Asked Question
FDAL	Functional Development Assurance Level
GM	Guidance Material
IDAL	Item Development Assurance Level
JAA	Joint Aviation Authorities (predecessor of EASA)
LOI	Level of Involvement
MBD	Model-Based Development
OOTRT	Object-Oriented Technology and Related Techniques
OPR	Open Problem Report
PDI	Parameter Data Item
PR	Problem Report
RTC	Restricted Type Certificate
SOI	Stage of Involvement
SW	Software
STC	Supplemental Type Certificate
TC	Type Certificate

2 BACKGROUND AND EXISTING GUIDANCE

Current aircraft systems include items of digital equipment that contain software components. Compliance with CS 25.1301 and 1309[1] is partly addressed through development assurance activities conducted on the system itself. Additionally, in accordance with AMC 20-115C, the applicant may choose EUROCAE ED-12C / RTCA DO-178C as an approved method to secure software approval.

ED-215 / DO-330 - *Software Tool Qualification Considerations* - provides guidance for tool qualification and is called up by ED-12C / DO-178C. As stated in AMC 20-115C, whenever model-based development and verification techniques, object-oriented technology or formal methods are used in software-based systems, the corresponding supplement or supplements to ED-12C / DO-178C should be applied in addition to ED-12C / DO-178C itself.

The EUROCAE ED-12C / RTCA DO-178C document and its supplements do not, however, provide sufficient guidance regarding some important aspects such as the use of legacy software, tool qualification for legacy software, the use of model simulation for certification credit, the data and control coupling aspects of structural coverage analysis, the monitoring and handling of stack overflows and the coordinated use of multiple guidance documents and supplements that may be necessary for ED-12C / DO-178C projects. The aim of this Certification Memorandum is to provide additional guidelines to the applicant on these aspects for ED-12C / DO-178C projects.

This document also provides the objectives of this EASA Software Certification Memorandum with which applicants should comply in addition to the objectives of ED-12C / DO-178C and the objectives of any of the related documents/supplements that apply to their project. Additional activities are also proposed to meet those objectives, but an applicant may propose alternative activities to those proposed by EASA, provided that those activities provide EASA with an equivalent level of confidence.

Projects for which ED-12B / DO-178B is the applicable software guidance material should continue to use Issue 1 Revision 1 of the EASA Software Certification Memorandum, which applies to ED-12B / DO-178B projects, as called up in the applicable CRI for the project.

Many of the topics that were explained by specific sections of Issue 1 of the EASA Software Certification Memorandum for ED-12B / DO-178B projects are now covered by ED-12C / DO-178C, its supplements or by ED-94C / DO-248C. Those sections have been deleted from this Issue 2 of this document, as the purpose of this document is to provide guidance on topics for which there is not sufficient information for ED-12C / DO-178C projects.

As so many of the sections that were included in Issue 1 of this Certification Memorandum have been deleted from this Issue 2, the remaining sections of this document have been renumbered. As a result, the section numbers of this document no longer correspond to those of Issue 1 or to the existing FAA documentation.

2.1 ADDITIONAL GUIDANCE MATERIAL FOR AIRBORNE SOFTWARE

For projects in which the FAA is the prime authority or in which a concurrent certification or validation process is conducted by EASA and the FAA, then the latest versions of the current FAA guidance material (e.g. FAA Order 8110.49) for airborne software will also apply in addition to the guidance of this document.

To ensure full coordination with FAA AC 20-115C, section 4 of this Certification Memorandum explains which parts of FAA AC 20-115C need to be considered when an applicant proposes their software for approval.

[1] This applies for Large Aeroplanes. For other products, please refer to CS23.1301 and 23.1309 for Small Aeroplanes, CS27.1301 and 27.1309 for Small Rotorcraft, CS29.1301 and 29.1309 for Large Rotorcraft, CS E-50 (d,f) for engines, CS-P, CS-APU and CS-ETSO.

3 EASA CERTIFICATION POLICY

3.1 EASA POLICY

AMC 20-115C recognises Eurocae ED-12C / RTCA DO-178C as an acceptable means of compliance for software (SW) aspects of certification that pertain to the production of software for airborne systems and equipment used on aircraft, engines, propellers and auxiliary power units.

ED-215 / DO-330 provides guidance for tool qualification and is called up by ED-12C / DO-178C. As stated in AMC 20-115C, whenever model-based development and verification techniques, object-oriented technology or formal methods are used in software-based systems, the corresponding supplement or supplements to ED-12C / DO-178C should be applied in addition to ED-12C / DO-178C.

3.2 WHOM THIS CERTIFICATION MEMORANDUM AFFECTS

The guidance contained in this Certification Memorandum applies to any applicants seeking approval from EASA for software embedded in aircraft systems or engines that is intended to comply with ED-12C / DO-178C. It also applies to any personnel involved in the ED-12C / DO-178C activities related to the airborne software of those applicants.

For TCs and STCs, applicants should ensure that they use the appropriate version of the Certification Memorandum called up in the applicable CRI.

For an ETSO, the applicant may decide to take into account all or part of this guidance contained herein, and may substantiate the details of their compliance in specific documentation (i.e. Declaration of Design and Performance, Software Accomplishment Summary, Hardware Accomplishment Summary or equivalent). Caution should be taken as the content of Certification Memoranda may have changed by the time the equipment is installed in the Aircraft/Engine. In any case, the installed equipment should finally comply with the Aircraft/Engine Certification Basis (including certain Certification Review Items).

When this Certification Memorandum is used outside of the scope of a TC, STC or ETSO (e.g. for pre-consultancy, pre-application, etc.), this guidance is provided for information only and caution should be taken as the content of the Certification Memorandum may have changed by the time of the application.

3.3 THE USE OF EUROCAE ED-94C / DO-248C CLARIFICATIONS

The purpose of ED-94C / DO-248C is to provide clarification of the guidance material in ED-12C / DO-178C.

ED-94C / DO-248C should be used for either or both of the following purposes:

- Clarification of a specific section or topic of ED-12C / DO-178C.
- Resolution of an inconsistency between ED-12C / DO-178C and any other relevant civil aviation standards.

4 COORDINATION WITH FAA AC 20-115C AND EASA AMC 20-115C.

4.1 BACKGROUND

Some items of guidance included in FAA Advisory Circular AC 20-115C are more specific than the guidance contained in the corresponding sections of AMC 20-115C. In order to promote coordination with FAA AC 20-115C, the following sub-paragraphs contain that additional more specific FAA guidance material, which should also be complied with by applicants seeking software approval from EASA. The sub-paragraph below that deals with modifying and re-using legacy software also quotes a section of EASA AMC 20-115C to ensure that applicants have all the necessary information available and that they can coordinate their work with both the EASA and FAA policies on this subject.

At the end of each of the following sub-paragraphs, the EASA objectives of this Certification Memorandum corresponding to the topic covered by the sub-paragraph are stated.

4.2 MODEL-BASED DEVELOPMENT AND VERIFICATION.

The following activities contained in section 8.c. of FAA AC 20-115C are applicable for ED-12C / DO-178C projects that use model-based development and verification for their airborne software:

"If you are using models as defined in ED-218 / DO-331, section MB.1.0, as the basis for developing software, you should apply the guidance in ED-218 / DO-331.

(1) Section MB.6.8.1 identifies certain objectives and describes the activities for using model simulation to satisfy those objectives. When applying section MB.6.8.1:

> *(a) You should identify which of the objectives you propose to satisfy using model simulation.*
>
> *(b) If you propose to use model simulation in combination with reviews and analysis to satisfy the objectives in MB.6.8.1, you should show that the errors detected include all errors that could be detected by reviews and analysis alone.*

(2) Section MB.6.8.2 identifies certain objectives relating to verification of the Executable Object Code and describes the activities for using model simulation to satisfy those objectives. When applying section MB.6.8.2:

> *(a) You should identify which of the objectives you propose to satisfy using model simulation.*
>
> *(b) If you propose to use model simulation in combination with testing to satisfy the objectives in MB.6.8.2, you should show that the errors detected include all errors that could be detected by testing on the target platform alone."*

EASA_SWCM_Objective_1 – If an applicant uses the guidance contained in ED-218 / DO-331 and wishes to use model-simulation to gain certification credit, the applicant has identified the objectives for which they wish to claim credit by the use of simulation and demonstrated the equivalence of their simulation for error detection as requested in FAA AC 20-115C.

4.3 MODIFYING AND RE-USING SOFTWARE APPROVED TO EARLIER VERSIONS OF ED-12 / DO-178.

4.3.1 EASA Legacy Software Guidance from AMC 20-115C

In order to avoid confusion about the use of previous ED-12 versions, EASA would like to remind applicants that they should apply the following guidance provided in section 8 of AMC 20-115C:

"8 USE OF PREVIOUS VERSIONS
8.1 Previous ED-12 versions may continue to be accepted for modifications to the software of already approved systems and equipment or for reuse of already approved software components in new applications for certification of products or parts and appliances.

8.2 Paragraph 8.1 applies, provided that:

- *The software level is not higher;*
- *The techniques described in the ED-12C supplements (MBD, OOTRT, Formal Methods) are not introduced into the new project; otherwise, ED-216 and/or ED-217 and/or ED-218 should be applied;*
- *The change to the ETSO authorized article is minor (see 21A.611);*
- *No new software criteria 1 or 2 tool qualification is needed; otherwise ED-215 should be applied only on the new software criteria 1 or 2 tools if the existing tools are not significantly changed;*
- *No new Parameter Data Item files are introduced, otherwise ED-12C should only be applied on the new Parameter Data Item files if the existing PDIs are not significantly changed and it should be demonstrated that software using the new Parameter Data Item files is compliant with the ED-12C sections related to Parameter Data Items;*
- *Software plans, processes, and the life cycle environment, including process improvements have been maintained;*

8.3 Where a modification is made to an existing software-based equipment or system, and the criteria in this section indicate the use of ED-12C and related supplements, they may apply, under justification, only to the software components affected by the modification.

For major changes to ETSO authorised articles, a previous version of ED-12 may continue to be accepted under justification.

Early coordination with EASA is strongly recommended to validate the above assumptions."

This above EASA guidance defined in section 8 of AMC 20-115C is equivalent to the guidance provided in FAA AC 20-115C Section 9 (7), 9(8), 9(9)

4.3.2 FAA AC 20-115C Legacy Software Guidance (section 9)

The following sections of FAA AC 20-115C have been copied/pasted into this document for harmonisation/coordination purposes and are applicable to EASA projects using ED-12C / DO-178C.

"AC 20-115C Section 9 (1):
Assess the legacy system software to be modified, or re-used in a different product, for its usage history from previous installations. If the software has safety-related service difficulties, airworthiness directives, or open problem reports that may have a safety impact on the proposed installation, correct the known software and development process deficiencies prior to modifying or re-using it in a different product.

AC 20-115C Section 9 (2):
The guidance of ED-12B / DO-178B applies to four levels of software assurance, whereas the guidance of ED-12 / DO-178 and ED-12A / DO-178A applies to three levels. ED-12C / DO-178C has retained the ED-12B / DO-178B software levels. Use Table 1 to determine if your legacy system software level satisfies the software level assigned by the system safety assessment for the proposed installation. A "✓" in the intersection of the row and column indicates that the legacy system software level is acceptable. For example, legacy system software with assurance to ED-12A / DO-178A software Essential/Level 2 can be considered to satisfy ED-12B / DO-178B or ED-12C / DO-178C software Levels C and D. A blank indicates that the software level is not acceptable. Therefore, ED-12A / DO-178A software developed to Essential/Level 2 would not be acceptable where ED-12B / DO-178B or ED-12C / DO-178C software Levels A or B are required.

Table 1 - Assurance Level Relationships ED-12B/C/DO-178B/C Software Level Assigned by the System Safety Assessment	**Legacy System Software Level per ED-12/DO-178/ED-12A/DO-178A**			**Legacy System Software Level per ED-12B / DO-178B**			
	Critical/ Level 1	***Essential/ Level 2***	***Non-Essential/ Level 3***	***A***	***B***	***C***	***D***
A	✓			✓			
B	✓	*		✓	✓		
C	✓	✓		✓	✓	✓	
D	✓	✓		✓	✓	✓	✓

****** For legacy system software developed using ED-12 / DO-178 or ED-12A / DO-178A at Essential/Level 2 that was previously shown to be equivalent to ED-12B/DO-178B Level B per Order 8110.49, paragraph 10-3.a.(1), equivalency remains valid for the new project.*

(a) If your legacy system software was developed using ED-12 / DO-178 or ED-12A / DO-178A, and the software level is not acceptable, upgrade the software development baseline using ED-12C / DO-178C, section 12.1.4.

(b) If your legacy system software was developed using ED-12B / DO-178B, and the software level is not acceptable, upgrade the software development baseline using ED-12B / DO-178B or ED-12C / DO-178C, section 12.1.4.

AC 20-115C Section 9 (3):
If the usage history of your legacy system software is acceptable, the software level has a "✓" entry in Table 1 (or the baseline has been upgraded appropriately), and modifications to the software are not required, then the original approval may serve as the basis for the software in the installation approval of the proposed system. If you upgraded the software development baseline using DO-178C and you want to declare your software as having satisfied DO-178C, you should update your processes and procedures, including tool qualification processes, to DO-178C. However, you cannot declare your unmodified tools as having satisfied DO-178C. All subsequent modifications are to be made using your processes and procedures that satisfy DO-178C.

AC 20-115C Section 9 (6):
If you upgraded the software baseline to ED-12C / DO-178C, or as an alternative to modifying your legacy system software using DO-178, ED-12A / DO-178A, or ED-12B / DO-178B, make all modifications to the software using ED-12C / DO-178C, section 12.1. If you want to declare your software as having satisfied ED-12C / DO-178C, you should accomplish all software modifications using ED-12C / DO-178C and update your processes and procedures, including tool qualification processes, to ED-12C / DO-178C. Your declaration applies to both modified and unmodified software and is valid even if you use unmodified tools that have not been qualified using ED-12C / DO-178C. However, you cannot declare your unmodified tools as having satisfied ED-12C / DO-178C. All subsequent modifications are to be made using your processes and procedures that satisfy ED-12C / DO-178C."

EASA_SWCM_Objective_2 – When using the guidance contained in ED-12C / DO-178C regarding the reuse of legacy software, the applicant has ensured that the conditions explained in FAA AC 20-115C sections 9 (1), 9 (2), 9 (3) and 9 (6) are met as well as the guidance of section 8 of EASA AMC 20-115C.

4.4 TOOL QUALIFICATION FOR LEGACY SOFTWARE.

4.4.1 FAA AC 20-115C Tool Qualification Guidance (section 10)

As EASA AMC 20-115C does not provide guidance regarding the use of legacy tools, the following sections of the FAA AC 20-115C have been copied/pasted into this document for harmonisation/coordination purposes and are applicable to EASA projects using ED-12C / DO-178C.

"AC 20-115C Section 10:

ED-12C / DO-178C, section 12.2, and ED-215 / DO-330 provide an acceptable method for tool qualification. ED-215 / DO-330 contains its own complete set of objectives, activities, and life cycle data for tool qualification.

***a.** If your legacy system software was previously approved using ED-12 / DO-178 or ED-12A / DO-178A, and you intend to use a new or modified tool for modifications to the legacy system software, use the criteria of ED-12C / DO-178C, section 12.2, to determine if tool qualification is needed. If you need to qualify the tool, use the software level assigned by the system safety assessment for determining the required TQL, and use ED-215 / DO-330 for the applicable objectives, activities, guidance, and life cycle data. You may declare your qualified tool as having satisfied ED-215 / DO-330 and not the legacy system software as having satisfied ED-12C / DO-178C.*

***b.** If your legacy system software was previously approved using ED-12B / DO-178B, and you do not intend to claim compliance to ED-12C / DO-178C, you can use your ED-12B / DO-178B tool qualification processes for qualifying new or modified tools in support of modifications to ED-12B / DO-178B legacy system software.*

***c.** If your legacy system software was previously approved using ED-12B / DO-178B, you intend to claim compliance to ED-12C / DO-178C, and you have ED-12B / DO-178B legacy tools that need to be qualified, follow the guidance of this subparagraph.*

(1) ED-12C / DO-178C establishes five levels of tool qualification based on the tool use and its potential impact in the software life cycle processes (see ED-12C / DO-178C, section 12.2.2 and Table 121). However, ED-12C / DO-178C does not address the use of tools previously qualified to the ED-12B / DO-178B criteria. For a tool previously qualified as a ED-12B / DO-178B development tool or verification tool, use Table 2 (below) to determine the correlation between the ED-12B / DO-178B tool qualification type and ED-12C / DO-178C tool criteria and tool qualification levels (TQLs).

Table 2 - Correlation Between ED-12B / DO-178B Tool Qualification Type and ED-12C / DO-178C Tool Criteria/TQL ***ED-12B / DO-178B Tool Qualification Type***	***Software Level***	***ED-12C / DO-178C Tool Criteria***	***ED-12C / DO-178C/DO-330 TQL***
Development	A	1	TQL-1
Development	B	1	TQL-2
Development	C	1	TQL-3
Development	D	1	TQL-4
Verification	A, B	2	TQL-4
Verification	C, D	2	TQL-5
Verification	All	3	TQL-5

(2) Development Tools Previously Qualified Using ED-12B / DO-178B.

(a) If the ED-12B / DO-178B software level assigned to the tool correlates with or exceeds the required TQL established by ED-12C / DO-178C, you may continue to use your ED-12B / DO-178B tool qualification processes for a ED-12C / DO-178C project or use ED-215 / DO-330.

(i) If there are changes to the tool's operational environment, refer to ED-215 / DO-330, section 11.2.2, for guidance on performing an analysis to determine what activities need to be performed or re-performed.

(ii) If there are changes to the tool, refer to ED-215 / DO-330, section 11.2.3, for conducting a tool change impact analysis. Use the tool change impact analysis to determine the potential impact of the change on the generated code and the needed re-verification activities.

(b) If the ED-12B / DO-178B software level assigned to the tool does not satisfy the required TQL for a ED-12C / DO-178C project, you should re-qualify the tool using ED-215 / DO-330.
(c) You may declare your tool as having satisfied ED-215 / DO-330 if all changes to the tool and your tool qualification processes satisfy ED-215 / DO-330.

(3) Verification Tools Previously Qualified Using ED-12B / DO-178B.

(a) If the tool qualification level required for a ED-12C / DO-178C project is TQL5, and your verification tool was previously qualified using ED-12B / DO-178B:

(i) You may continue to use your ED-12B / DO-178B tool qualification process.

(ii) If there are changes to the tool or the tool's operational environment, you should conduct a tool change impact analysis and re-verify the tool using your ED-12B / DO-178B tool qualification processes or re-qualify the tool using ED-215 / DO-330."

(b) If the tool qualification level required for a DO-178C project is TQL4, you should re-qualify your verification tool using DO-330.

(c) You may declare your tool as having satisfied DO-330 if all changes to the tool and your tool qualification processes satisfy DO-330."

EASA_SWCM_Objective_3 – If an applicant re-uses legacy tools, the applicant has ensured that their use of those tools will comply with the guidance on legacy tools in ED-12C / DO-178C and with the conditions explained in section 10 of FAA AC 20-115C.

5 GUIDELINES WHEN SUBMITTING PSAC(S) TO EASA

5.1 BACKGROUND

EASA AMC 20-115C recognises the EUROCAE ED-12C standard and five related documents and supplements. Due to the potential use of multiple documents/supplements together and in combination with each other, applicants may experience some difficulties in clearly stating how their proposed means of compliance allows them to meet all the objectives and activities defined in all the applicable supplement(s). In particular, the supplement(s) that address specific software techniques may add, delete or modify objectives, activities, and life cycle data items requested by the core ED-12C / DO-178C document. This section provides guidance on the information EASA requests applicants to provide in their PSAC(s) so as to clarify these aspects.

Some of the guidance contained in this section has been copied from FAA AC 20-115C so as to provide applicants with a complete picture of the combined EASA and FAA guidance regarding additional items that need to be contained in PSACs for ED-12C / DO-178C projects.

NOTE: in the guidance below, the word 'supplement' applies to either the Formal Methods Supplement (ED-216 / DO-333), the Object-Oriented Technology and Related Techniques Supplement (ED-217 / DO-332) or to the Model-based Development and Verification Supplement (ED-218 / DO-331). It does not apply to ED-215 / DO-330 (Software Tool Qualification Considerations), which is considered by EUROCAE / RTCA to be a document rather than a supplement.

5.2 GUIDANCE

When an applicant intends to use multiple software development techniques together, the applicant should submit a PSAC describing the applicable supplement objectives and activities related to each of the developed software components. The proposed means of compliance for all objectives should be consistent and traceable to those objectives.

EASA_SWCM_Objective_4 – The applicant has ensured that, for all software components and qualified tools proposed for approval, all their related proposed means of compliance are defined consistently and completely and are traceable to all the applicable ED-12C / DO-178C objectives and activities (including those of the applicable documents and supplements).

Proposed activities:

If one or more of the documents or supplements applies, each proposed PSAC should:

1. describe how ED-12C / DO-178C and the applicable supplements will be used together,
2. explain how the applicable ED-12C / DO-178C objectives and those objectives added or modified by the supplements are addressed,
3. define which objectives apply to which software components,
4. substantiate how the applicant's proposed means of compliance will satisfy all the applicable objectives and activities,
5. contain a traceability matrix showing how each applicable objective and its related activities trace to each proposed means of compliance.

   NOTE – When software components are developed using several different software techniques and consequently multiple ED-12C / DO-178C supplements apply, the applicant may need to develop multiple traceability matrices.

If ED-215 / DO-330 (Software Tool Qualification Considerations) applies and the applicant intends to use any techniques addressed by the supplements to satisfy the ED-215 / DO-330 objectives, then the applicant should use the applicable supplements for those objectives (tool qualification levels (TQLs) 1, 2, 3, and 4 only).

If ED-215 / DO-330 (Software Tool Qualification Considerations) applies, each Tool Qualification Plan (TQL) should:

6. describe how the applicant will apply ED-215 / DO-330 and the supplement guidance to the tool development or verification,

7. describe how the applicant will address the applicable ED-215 / DO-330 objectives and those added or modified by the supplements, which objectives apply to which components of each software tool, and how the planned activities will satisfy all applicable objectives,

8. contain a traceability matrix in which each applicable objective and its related activities trace to the corresponding proposed means of compliance.

6 GUIDELINES FOR THE SOFTWARE REVIEW PROCESS

The applicant should define their own software review process so as to provide confidence to EASA that the applicant's reviews will ensure that the software processes under review include the activities and meet the objectives of ED-12C / DO-178C. This review process should at least include the four typical SOI reviews described in the EASA Review Process and the software life-cycle data items that are shown as inputs to those reviews.

Note: The EASA Review Process for conducting software reviews is now described in the EASA Certification Handbook (available upon request).

For projects in which the FAA is the prime authority or in which a concurrent certification or validation process is conducted by EASA and the FAA, then the Applicant may choose to define their own software review process based on the software review process defined in the latest issue of FAA Order 8110.49 or the FAA Job Aid "Conducting Software Reviews Prior to Certification".

EASA_SWCM_Objective_5 – The applicant has defined its software review process to ensure that the software under review complies with the objectives of ED-12C / DO-178C (and its related supplements) and that the activities corresponding to those objectives are conducted.

NOTE - This objective covers the applicable objectives and activities of EASA SW CRIs where applicable.

7 LEVEL OF INVOLVEMENT OF APPLICANTS IN SOFTWARE PROJECTS

Each applicant should define its process for defining its Level Of Involvement (LOI) in the oversight of software developed for its aircraft. The Applicant's LOI definition process should include a clear proposal in which the levels of involvement, corresponding reviews and the categories of documents for delivery are similar to those described by EASA in the EASA Certification Handbook (available on request).

The applicant should produce a document (at aircraft level or alternatively at ATA chapter or system level) for EASA concurrence that lists the software components in all the systems on the aircraft and shows the DAL (or FDAL or IDAL), the applicant's planned level of involvement and the suppliers involved for each software component. The applicant should present to EASA the activities they plan to monitor (including a list of reviews and a schedule) and state the rationale for the activities they plan to conduct under their DOA system or equivalent.

If an Applicant chose to not define their own LOI for the software embedded in each of the systems of their aircraft, it would therefore be assumed that the Applicant would perform all the Stage of Involvement (SOI) reviews described in its review process for all the software used on their aircraft, irrespective of the DAL of the software or of whether the software is new or modified.

EASA_SWCM_Objective_6 – In order to customize its review process, the applicant has defined its Level Of Involvement for all the embedded software of their aircraft and provided EASA with documentation showing for each embedded software component the applicant's LOI, the software reviews to be conducted corresponding to the LOI, the DAL(s) of the software and the supplier.

8 OVERSIGHT OF SOFTWARE CHANGE IMPACT ANALYSES USED TO CLASSIFY SOFTWARE CHANGES AS MAJOR OR MINOR

8.1 BACKGROUND

ED-12C / DO-178C, Section 12.1.1, identifies analysis activities to be performed for proposed software changes. ED-12C / DO-178C also states that re-verification should be accomplished on all software changes and areas affected by those changes.

Subpart D of Part 21 addresses the classification of changes to type design as minor or major. Paragraph 21.A.91 proposes criteria for the classification of changes to a type design as minor or major.

The purpose of this classification is to determine the certification route to be followed in Part 21 Subpart D (either 21.A.95 or 21.A.97) or alternatively in Subpart E.

For approved ETSO'd articles, Subpart O of Part 21 is applicable for the classification of design changes (21.A.611).

8.2 PROCEDURES

Detailed guidance for the classification of changes to type design is given in GM 21.A.91.

9 MANAGEMENT OF PROBLEM REPORTS

The text that was formerly in this section has been superseded by the text contained in DP #9 of ED-94C / DO-248C. Please refer to that text for guidance on Open Problem Reports (OPRs).

DP #9 of ED-94C / DO-248C states: "A goal of the software life-cycle processes is to achieve zero OPRs at the time of SAS submittal for approval. However, if Problem Reports are open at this time, then the objective is to minimize their number and analyze their impact at the system level, in any software release presented for approval."

(For the purposes of this section of this Certification Memorandum, the term "failure to comply" shall henceforth mean "a failure to comply with ED-12C / DO-178C or with this Certification Memorandum".)

There should be no Problem Reports (PRs) recording "failures to comply" that still remain open (i.e. as OPRs) at the time of applying for software approval. Any such "failures to comply" should have been rectified prior to the applicant submitting their documentation to EASA for the approval of their software, so the PRs that recorded those problems should therefore have already been closed.

EASA wishes to inform applicants and their suppliers that approval would not be granted for any software for which PRs recording "failures to comply" were found to remain open at the time of application for software approval.

Applicants might, therefore, find it useful to record which of their Problem Reports record "failures to comply". For example:

- They may create a new category of Problem Report called "Category 5: Non-compliance with ED-12C / DO-178C / Software Certification Memorandum"; this would make it easier at the time of approval to identify these particular PRs and to ensure that they are closed prior to submission for approval.
- They may add an attribute to their OPR database to indicate for each Problem Report whether or not it records a "failure to comply".

NOTE - EASA has seen examples in the past in which suppliers have listed problems as, for example, Type 3 OPRs, but upon examination by EASA, these PRs have been found to actually capture cases of non-compliance with ED-12B / DO-178B processes that the supplier had not rectified prior to submission for software approval and about which they had not informed EASA.

EASA_SWCM_Objective_7 – In addition to handling their Problem Reports in the manner described in DP#9 of ED-94C / DO-248C, the applicant has ensured that no Problem Reports that record failures to comply with ED-12C / DO-178C or with this Software Certification Memorandum remain open at the time of applying for software approval.

10 CLARIFICATION OF STRUCTURAL COVERAGE ANALYSES OF DATA COUPLING AND CONTROL COUPLING

10.1 BACKGROUND

Numerous misinterpretations exist regarding the purpose of structural coverage analyses of data coupling and control coupling, and acceptable approaches to satisfying EUROCAE ED-12C / RTCA DO-178C Objective 8 of Annex A Table A-7. This objective ("Test coverage of software structure (data coupling and control coupling) is achieved") references section 6.4.4.d of ED-12C / DO-178C, which states that structural coverage "Test coverage of software structure, both data coupling and control coupling, is achieved."

Section 6.4.4.2.c of ED-12C / DO-178C states that activities for structural coverage analysis include "Analysis to confirm that the requirements-based testing has exercised the data and control coupling between code components." Applicants should note that this is different from what was required for ED-12B / DO-178B, where the data and control coupling only had to be confirmed, rather than actually exercised during testing.

This section of this Certification Memorandum discusses the purpose, benefits, challenges, and future concerns of data coupling and control coupling coverage analyses.

Objective 8 of Table A-7 in ED-12C / DO-178C specifies the analysis of data and control coupling for Levels A, B, and C software. ED-12C / DO-178C Annex B defines data coupling and control coupling as follows:

"Data coupling - The dependence of a software component on data not exclusively under the control of that software component."

"Control coupling - The manner or degree by which one software component influences the execution of another software component."

Additionally, ED-12C / DO-178C defines a component as: "A self-contained part, combination of parts, sub-assemblies, or units that performs a distinct function of a system."

Note: ED-94C / DO-248C, Final Report for Clarification of DO-178C "Software Considerations in Airborne Systems and Equipment Certification", Frequently Asked Question (FAQ) #67 provides some clarification on data coupling and control coupling.

10.2 CLARIFICATIONS

10.2.1 Purpose of data coupling and control coupling analyses

The intent of structural coverage analyses is to provide a measure of the completeness of the testing process of software to ensure that the requirements-based testing (R-BT) of a software program exercised that program's functions and structure adequately to an appropriate level of "completeness" depending on that program's software level and needed integrity. For example, structural coverage analysis of Level C software only needs to provide a measure that all statements were exercised; Level B needs a measure that all statements and all decisions were exercised; and Level A needs a measure that all statements, all decisions and all conditions (plus some independence) were exercised (see Objectives 5, 6 and 7 of Annex A Table A-7). These measurements can be taken and analyzed at the computer program "module" level by reviewing test cases and executing requirements-based tests of that module in isolation from other program modules, and examining, either manually or with a tool, that every statement, decision, and condition (depending on the software level of the module) were exercised, and the module functioned correctly as designed.

"Module" is used in this context to denote a piece or component of the software program rather than the entire program. If an airborne software program consisted of one "module," the above structural coverage would likely be adequate to ensure that the software program functioned correctly and would not have any side effects leading to anomalous behaviour. However, because of the size and complexity of embedded airborne software programs, having the program consist of a single, self-contained module is neither practical nor good

engineering. Thus, developers construct programs of modules or components (both functional and data components) that interact with one another and depend on one another to perform the program's functions.

The intent of the structural coverage analyses of data coupling and control coupling is to provide a measurement and assurance of the correctness of these modules/components' interactions and dependencies. That is, the intent is to show that the software modules/components affect one another in the ways in which the software designer intended and do not affect one another in ways in which they were not intended, thus resulting in unplanned, anomalous, or erroneous behaviour. Typically, the measurements and assurance should be conducted on R-BT of the integrated components (that is, on the final software program build) in order to ensure that the interactions and dependencies are correct, the coverage is complete, and the objective is satisfied.

Satisfaction of this objective is dependent on the detail of the specification of the modules/components' interfaces and the thoroughness of the R-BT for normal range and robustness of the software program. That is, if the interfaces and dependencies are specified in the design requirements, and if those requirements are tested for both normal functioning and robustness, satisfaction of the objective may be a by-product of the design and verification processes. However, if the interfaces and dependencies are not well-specified, and the testing program is minimal, it will be much more difficult to demonstrate the objective has been satisfied.

10.2.2 Design versus integration verification activity

A number of manufacturers perform activities during software design to minimize the data coupling and control coupling issues during integration. This is recognized as a good engineering practice as the Meiler Page-Jones' book entitled *The Practical Guide to Structured Systems Design* (1980) points out. Page-Jones identifies different kinds of coupling to be considered during design (e.g., data coupling, stamp coupling, common coupling, control coupling, and content coupling). In some cases (depending on the architecture), this analysis in the software design phase can be used to supplement the software/software integration activity.

However, objective 8 of ED-12C / DO-178C Table A-7 is primarily intended to be a verification of the integration activity; that is, verification that the interfaces and dependencies between the software program's modules/components were implemented as designed and are correct. Satisfying the objective is intended to provide a measure of the completeness of integration verification (R-BT of the integrated software program's structure, interfaces, and dependencies).

Many applicants find that documenting the data coupling and control coupling during software design provides the requirements to verify during the software/software integration and hardware/software integration verification process. That is, good documentation of the design helps to satisfy the objective during the integration testing (R-BT coverage of interface and dependency requirements).

10.2.3 EASA perspective on the purpose of data coupling analysis

EASA proposes that the purpose of data coupling analysis is to:

- Be a completion check of the integration testing effort. The analysis also provides insight into the structural robustness of the data structures used by the program. Basically, data coupling analysis is intended to enforce good software engineering practices. Data coupling analysis becomes particularly important when partitioning and other protection means are implemented in the software.
- Identify data dependencies. As an example, a data dependence exists between two components when one component defines a data object and the other component uses the definition of that data object under some operational scenario. In this example, the data user is dependent on the data definer.
- Verify data interfaces between modules/components through testing and analysis (test it, then measure it).
- Identify inappropriate data dependencies.

- Define and evaluate the extent of interface depth.
- Determine and minimize coupling interdependencies.
- Determine and maximize cohesion.
- Evaluate the need for and accurate use of global data.
- Evaluate input/output data buffers.
- Bound impact of change and requirements effect(s).

10.2.4 EASA Perspective on the purpose of control coupling analysis

EASA proposes that the purpose of control coupling analysis is to:

- Be a complementary completion check of the integration testing effort (i.e., it complements data coupling analysis). The analysis also provides insight into the structural robustness of the execution, timing, and scheduling. Basically, control coupling is intended to enforce good software engineering practices. Control coupling becomes particularly important when partitioning and other protection means are implemented in software.
- Identify control dependencies. A control dependence exists between two components when the execution of one depends on the other. For example, one module/component calls the other under some operational scenario (i.e., the callee is dependent on the caller). Another example is where one module/component defines the data objects that determine the execution sequence taken by the other module/component under some operational scenario.
- Identify inappropriate control dependencies.
- Verify correct execution call sequence (across modules / components / parts / units / objects).
- Define and evaluate the extent of interface depth.
- Assist in verifying scheduling (e.g., detect problems with call sequences that may cause frame overrun).
- Assist in worst-case execution time (WCET) analysis (a side benefit).
- Bound impact of change and requirements effect(s).

10.3 Common Benefits With Applying Data Coupling and Control Coupling Analyses

Certification authorities have observed that applicants who and have well-defined design and integration practices can identify and address data coupling and control coupling issues, and are able to:

- Provide a better awareness of functionality.
- Reduce the number of test cases needed to cover functionality and the supporting code structure, code interfaces, and requirements.
- Perform more efficient and effective change impact analysis.
- Find errors that are difficult to find in the lab testing and could be costly to fix in the field.
- Perform more effective maintenance.

10.4 Guidance for Satisfying the Data Coupling and Control Coupling Analyses Objective

EASA and other certification authorities have observed a number of problems with the application of data coupling and control coupling analyses to airborne software.

EASA_SWCM_Objective_8 – The applicant has complied with objective 8 of ED-12C / DO-178C table A-7 by achieving coverage of the data coupling and the control coupling of their software during requirement-based tests.

Proposed activities:

1. Applicants should address the data coupling and control coupling analyses in their plans (i.e., plan upfront how they will perform these analyses). In some cases, it may be distributed among several plans. However it is documented, it should provide a complete and accurate rationale (i.e., it should be thorough).

2. Applicants should consider data coupling and control coupling as part of their development/design effort (e.g., specify interface (I/O) requirements and dependencies between components).
3. Applicants should conduct an analysis to confirm that the requirements-based testing has completely exercised both the data coupling and the control coupling between code components and provide a rationale in their Software Verification Results for any part of the data and control coupling that was not exercised during requirement-based testing.
4. Since data coupling and control coupling analyses are two different and separate activities, applicants should develop their plans and procedures accordingly, and provide a report for data coupling analysis and a separate report for the control coupling analysis. If tools are used, the determination of whether they need to be qualified or not should be evaluated and justified.
5. If selective linkers are used, their effect on data coupling and control coupling should be analyzed.

An applicant may propose alternative activities to those proposed by EASA, provided that those activities provide EASA with an equivalent level of confidence.

11 STACK OVERFLOWS

11.1 PURPOSE

The purpose of this Section is to provide guidance for determining whether software designs should implement protection mechanisms to cope with stack overflows.

ED-12C / DO-178C already provides guidance regarding stack usage and stack overflows in the areas of code reviews and requirement based verification, however, it does not cover the possibility that a data corruption may occur and lead to a stack overflow.

The purpose of this section is not to change the intent of ED-12C / DO-178C but rather to provide details of specific aspects of stack usage and stack overflows to which attention should be paid. Although this section only refers to stack overflows, it should be understood that stack underflows also need to be taken into account in each case where stack overflows are mentioned and that the guidance in this section also applies to any other area of dynamic memory that is used during the runtime of an airborne software program, such as a heap.

11.2 BACKGROUND

Most avionics systems incorporate a real-time operating system, which may have many stacks that are used dynamically. In addition, there may be secondary stacks which may be used to manage aggregate data (for example dynamically sized arrays) or data whose lifetime extends beyond its calling frame (e.g. a locally declared record being returned from the function where it is declared).

Depending on the programming language used and the implementation of the compiler, various problems may arise in the event of a stack overflow. The problems may manifest themselves as:

1. A corruption of data (if the stack overflow causes the software to write into a data location).
2. A machine exception (if the stack overflow is adjacent to protected memory, either code or read-only memory).
3. Unpredictable program execution (if the corruption of data occurs and code addresses are corrupted, e.g. the return address of a function).
4. A software exception (if the program contains code to check the stack).

As secondary stacks are used less frequently and their use is typically under the control of run-time functions, stack checks are often put in place to monitor all secondary stack usage. Typically, run-time checks on primary stacks are avoided for performance reasons.

ED-12C / DO-178C requests that stack usage should be addressed appropriately by review of the source code and stack overflows should be addressed by requirement-based hardware/software integration testing.

Analysis of the theoretical worst case scenario helps to determine whether the use and implementation of stacks has been well-designed in order to adequately manage data that needs to be handled through stack usage.

Static analysis of worst case stack use consists of identifying all the functions in the program and obtaining their stack frame sizes, determining the call graph of each separate execution thread (typically a task), and combining this data into a set of worst case stack sizes for each stack. This can be done at the source code level by counting the sizes of all data declarations and parameters. This raises the problems that the sizes chosen by the compiler, any alignment gaps and any temporary compiler-allocated data on the stack would all need to be accounted for.

Stack analysis may also be performed by testing. The typical approach is to fill the memory with a certain memory pattern and to execute tests which force the maximum usage of the stack. As this behaviour is dynamic, it may be difficult to determine the worst-case scenario except for simple programs.

Nonetheless, there is an additional potential risk that an unintended stack overflow may occur during the execution of a program for various reasons, such as:

- A hardware failure.
- A software development error.
- Unintended software behaviour.
- A memory corruption.
- A single event upset (SEU).
- etc.

In such cases, when a stack overflow occurs during the execution of level A software, the consequences could be catastrophic as the software may be out of control.

Although an SEU is unlikely to be repeated, if a stack corruption or an overflow occurs due to a software error, then given the same program state and input conditions, the software error will be repeated. Such a repetition may cause the recovery mechanism to be ineffective.

11.3 GUIDANCE

EASA considers that for Level A software, there is a need to consider the possibility of a stack overflow occurring in flight. This means that theoretical measurements performed to determine that stack overflows cannot occur may be not sufficient to ensure that the software behaves as intended. Consequently, stack monitoring may be necessary to detect any stack overflows and handle the potential risks.

To conduct run time monitoring of stack overflow is not trivial and usually implies the incorporation of specific requirements to cover those concerns.

EASA_SWCM_Objective_9 – For level A software, the applicant has analyzed the usage of the stack(s) for each processor in their equipment, determined whether or not continuous stack monitoring is necessary to ensure that the behaviour of the hosted software is not affected by stack overflows and incorporated and tested any necessary stack monitoring.

Proposed activities:

a) An analysis should be performed to define whether or not it is necessary to perform continuous stack monitoring, based on criteria such as the stack type, the use of built-in monitors, etc.,
b) If continuous monitoring of the stack is used, it should be performed in real time,
c) The monitoring mechanism (e.g. the monitoring of the stack pointer) should be specified in the requirements,
d) In the event that the monitor detects an overflow, the expected behaviour (e.g. exception) should be specified and verified accordingly.

There may be several stacks with different monitoring policies (data, executive-level, program counters, etc.) and the above analyses may be needed for each stack.

An applicant may propose alternative activities to those proposed by EASA, provided that those activities provide EASA with an equivalent level of confidence.

12 REMARKS

1. The EASA Proposed Certification Memorandum will be closed for public consultation on the **3rd of December 2013**. Comments received after the indicated closing date for consultation might not be taken into account.
2. Comments regarding this EASA Proposed Certification Memorandum should be referred to the Certification Policy and Planning Department, Certification Directorate, EASA. E-mail CM@easa.europa.eu or fax +49 (0)221 89990 4459.
3. For any question concerning the technical content of this EASA Proposed Certification Memorandum, please contact:

   Name, First Name: Bridge, Andrew

   Function: Certification Expert - Software and Airborne Electronic Hardware

   Phone: +49 (0)221 89990 4324

   Facsimile: +49 (0)221 89990 4825

   E-mail: andrew.bridge@easa.europa.eu