EASA
European Union Aviation Safety Agency

FAA
Aviation Safety

SW&AEH TASK FORCE
“ABSTRACTION LAYER”

Issue 1

“Criteria for accepting alternative standards to ED-12C/DO-178C and ED-80/DO-254”

Revision History

Revision	Description	Date
Issue 1	Initial document	October 2023

Table of Contents

Executive Summary

Development assurance for Software and Airborne Electronic Hardware (SW&AEH) has become the common methodology for certifying the complex and integrated systems and equipment used on aircraft. The guidance in ED-12C/DO-178C and ED-80/DO-254, as well as aspects of ARP-4754A, are the current industry standards and recommended practices that development teams and certifying authorities rely on to provide the confidence that is necessary for the level of safety for aviation products.

While the use of these standards and recommended practices has been shown to meet the current safety needs of systems and equipment development for compliance with applicable airworthiness regulations, alternatives to the common approach are sometimes needed. It may be desirable to incorporate software or electronic hardware components that have been developed to other safety standards, and emerging technologies and novel techniques, where a development assurance approach is viable, may require other standards or methodologies, which need to be assessed prior to acceptance as an acceptable means for safety critical systems and equipment. Additionally, anecdotal stakeholder feedback indicates that the current standards do not offer sufficient flexibility and can be prescriptive and burdensome.

In June 2019, EASA and the FAA established the first phase of the Task Force “Abstraction Layer” (TFAL) to develop a means to assess other standards or publicly available methodologies. This Task Force (TF), chaired by EASA and the FAA, was initially composed of representatives from EASA, FAA, and industry representatives from SW and AEH domains, nominated by aviation industry associations. In a second phase, the group was extended to TCCA, ANAC, and additional industry members for a trial application ofthe Abstraction Layer (AL) material on a public alternate standard.

As a result of its work, the TFAL delivered a set of twenty criteria, defining goals for development assurance in the domain of Software and Airborne Electronic Hardware. Each criterion describes one intent of an overall development assurance process. Each criterion was developed with the intent to be SMART[1] and process-centric (e.g., “the process encompasses, ensures, allows, …”).

Each criterion is accompanied with rationale and evaluation items. The rationale provides the reason why the criterion is necessary to support development assurance. Evaluation items describe attributes that the process ensures or provides; or, in some cases, assurance that other evaluation items are achieved. In addition, consideration for gradual level of rigor in development assurance is given through four ‘safety levels’. The indication for applicability to a safety level is added in front of each evaluation item in order to reflect a safety continuum into the Abstraction Layer.

The AL is not intended to serve as an alternative standard. It does not intend to invalidate or put at risk the current development assurance practices. It does not mirror what the currently acceptable standards already define.

[1] SMART: Specific, Measurable, Achievable, Realistic, Tangible

The most challenging aspects in formulating criteria were to select the right level of abstraction to be, on one hand, high enough to offer flexibility for alternate standards/public methodologies and, on the other hand, specific enough to address the challenges and essentials of Software and Airborne Electronic Hardware development assurance, enabling the detection of insufficiencies for use in aviation safety applications.

A key achievement of the TFAL is to have successfully formulated a common set of criteria for software and Airborne Electronic Hardware while considering the respective challenges of each domain. The second phase successfully confirmed the Abstraction Layer material through a trial case to assess a public alternate standard as applied to software and electronic hardware development. Feedback from this trial case supported the improvement and clarification of a few elements of the Abstraction Layer material.

This document also includes a User Guide describing 'how to use the Abstraction Layer'.

1 Introduction

1.1 Background

Development assurance for Software and Airborne Electronic Hardware (SW&AEH) has become the common methodology for certifying the complex and integrated systems and equipment used on aircraft. The guidance in ED-12C/DO-178C[2] and ED-80/DO-254, as well as aspects of ARP-4754A, are the current industry standards and recommended practices that development teams and certifying authorities typically rely on to provide the confidence that is necessary for the level of safety for aviation products.

While the use of these standards and recommended proactices has been shown to meet the current safety needs of systems and equipment development for compliance with applicable airworthiness regulations, alternatives to the common approach may be needed. It may be desirable to incorporate equipment that has been developed to other safety standards, and emerging technologies and novel techniques, where a development assurance approach is viable, may require other standards or publicly available methodologies, which need to be assessed prior to acceptance as an acceptable means for safety critical systems and equipment. Additionally, anecdotal stakeholder feedback indicates that the current standards do not offer sufficient flexibility and can be prescriptive and burdensome. The Abstraction Layer provides a means to assess other standards or methodologies and helps create a framework for their use in aviation and certification.

1.2 Purpose

The objective of the Task Force was to develop an "Abstraction Layer", extracting the key concepts from ED-12C/DO-178C, ED-80/DO-254, EASA & FAA A(M)C 20-115D, and EASA & FAA A(M)C 20-152A, and formulating criteria for the assessment of alternative approaches.

This Abstraction Layer is intended to be a 'bridging tool', a set of criteria to assess potential alternate standards used in other industry domains. It will also facilitate introduction of novel technologies by enabling the assessment of other development assurance standards.

When the Task Force initiated its activity, an important first step was to define an "Abstraction Layer". In order to gain a consistent understanding, and to dispel misinterpretations about what is meant by an "Abstraction Layer", it may first be instructive to understand what the Abstraction Layer is not. It is not intended to serve as some kind of new alternative standard. It does not intend to invalidate or put at risk the current development assurance practices. It does not mirror what the currently acceptable standards already define.

The effort of the Task Force, resulted in the Abstraction Layer described in this report, which establishes SMART criteria to facilitate the assessment of other standards or publicly available methodologies without prescribing the means used to satisfy the criteria.

Note: this Abstraction Layer does not cover Artificial Intelligence and Machine learning, where challenges go far beyond the development assurance topics and for which novel "learning assurance" concepts will need to be investigated.

[2] Reference to ED-12C/DO-178C includes use of ED-215/DO-330 and supplements ED-218/DO-331, ED-217/DO-332, and ED-216/DO-333, as applicable.

A criterion is satisfied through an assessment of the corresponding processes described in the standard or methodology. When a criterion is not satisfied or partially satisfied, the Abstraction Layer helps understand the identified gaps to facilitate decisions related to electing usage of the standard or methodology in the development of safety related items, based on the information provided within the criterion.

1.3 Document overview

The document is structured as follows:

- Section 1 presents the introduction of the Abstraction Layer.
- Section 2 presents the scope of the Abstraction Layer.
- Section 3 presents the working method.
- Section 4 presents the introduction of the Abstraction Layer criteria.
- Section 5 presents the Abstraction Layer criteria.
- Section 6 presents the User Guide on “How to use the Abstraction Layer”
- Section 7 presents the conclusion.
- Appendix I introduces definitions and acronyms.
- Appendix II is reserved.
- Appendix III lists the referenced documents.
- Appendix IV explains the safety levels.
- Appendix V provides the Abstraction Layer User Guide.

2 Scope of the Abstraction Layer

ED-12C/DO-178C and EASA/FAA A(M)C 20-115D relate to development assurance of software items while ED-80/DO-254 and EASA/FAA A(M)C 20-152A relate to development assurance of airborne electronic hardware items.

The scope of the applicability of the Abstraction Layer is on development assurance methods for an item or a collection of items.

In the context of Abstraction Layer, "collection of items" is a set of items closely interacting with each other at the first level integration above the items (e.g., typically at the level of a core processing platform). This collection of items excludes incremental certification, such as for Integrated Modular Avionics (IMA).

Note: aviation standards ED-12C/DO-178C and ED-80/DO-254 include guidance for Certification Liaison, underlying the expectations from Aviation authority within a development project, regarding deliverables and interactions with industry. Criteria for the Certification Liaison process were not 'abstracted' as this process is specific to the Aviation domain. It is not expected that a development standard from another domain would have an equivalent process. This Certification Liaison process remains but is considered a parallel process, in relation to the development assurance process.

3 Working method

The Abstraction Layer criteria were defined following a bottom-up approach. The work started by extracting or capturing the fundamental concepts of development assurance from ED-12C/DO-178C and ED-80/DO-254 standards that have a history of successful use. The AL criteria help understand the fundamental aspects behind the objectives and activities described in those standards.

The criteria are defined in the form of SMART goals. Each criterion is supplemented with:

- Rationale: describes the intent of a criterion to facilitate its understanding.
- Evaluation items: are attributes to support the evaluation of the notion of 'goodness' of the process, i.e., supporting information for determining or evaluating process performance or if the level of confidence brought by the process is acceptable.

These criteria, rationale, and evaluation items provide a foundation supporting exploration of other standards and publicly available methodologies used to develop safety critical systems.

The Abstraction Layer includes aspects that are not specifically addressed in the standards ED-12C/DO-178C and ED-80/DO-254 but are necessary when taking into consideration the fundamental aspects of the domains (e.g., EASA AMC 20-115D & FAA AC 20-115D, EASA AMC 20-152A & FAA AC 20-152A, SAE ARP-4754A).

3.1 Breakdown of the Task

The following work breakdown was performed to obtain the Abstraction Layer material.

- T1: Identify:
    - Definitions of terms and Abstraction Layer wording convention.
    - Basis to categorize criteria and their attributes according to the safety criticality level.
    - Structure of the information for each criterion.
      Each criterion is structured with 3 types of information:
        - The criterion itself
            - The rationale
            - Evaluation items
- T2: Assess current standards and develop the Abstraction Layer.
  To perform this task, ED-12C/DO-178C and ED-80/DO-254 were split by development processes. The concepts were exchanged between hardware and software and converged into a definition of common SW and AEH concepts, where possible.
    - T2.1: Capture the fundamental concepts of development assurance from the aviation standards.
    - T2.2: Capture the goals and formulate the rationale from the fundamental aspects captured in step T2.1.
    - T2.3: Merge SW and AEH criteria and consolidate the capture activity, where appropriate.
    - T2.4: Complement definition of criteria & rationale, where necessary, using concepts of A(M)C 20-152A, A(M)C 20-115D and ARP 4754A where different from or in addition to ED-12C/DO-178C and ED-80/DO-254.
    - T2.5: Organize and categorize the criteria and consider relationship to development assurance level.

- T2.6: Review the Abstraction Layer:
  - Review its content, and check if it supports both independent and concurrent SW and AEH development.
  - Assess the usability of the Abstraction Layer criteria by the Certification Authority/Certifying Personnel through use cases.

Section 4 provides the outcome of this task.

3.2 Method for Task 2 (T2)

To perform Task 2, for the purpose of extracting the key concepts of development assurance and defining the criteria, the Task Force opted for a stepped approach starting from a process or group of processes of the ED-12C/DO-178C, ED-80/DO-254, A(M)C 20-152A and A(M)C 20-115D, as follows:

- Step1 – identify what activity is DONE.
- Step 2 – abstract WHAT to ACHIEVE?
- Step 3 – abstract WHY DO IT?

The criteria are then captured from the outcome of the step 2 ‘what to achieve’, formulating the goal(s) or expectation(s) in single statements. The rationale of the criteria is captured from the outcome of step 3 ‘why do it’. The evaluation items describe elements of achievements that the process ensures or provides; or, in some cases, assurance that other evaluation items are achieved.

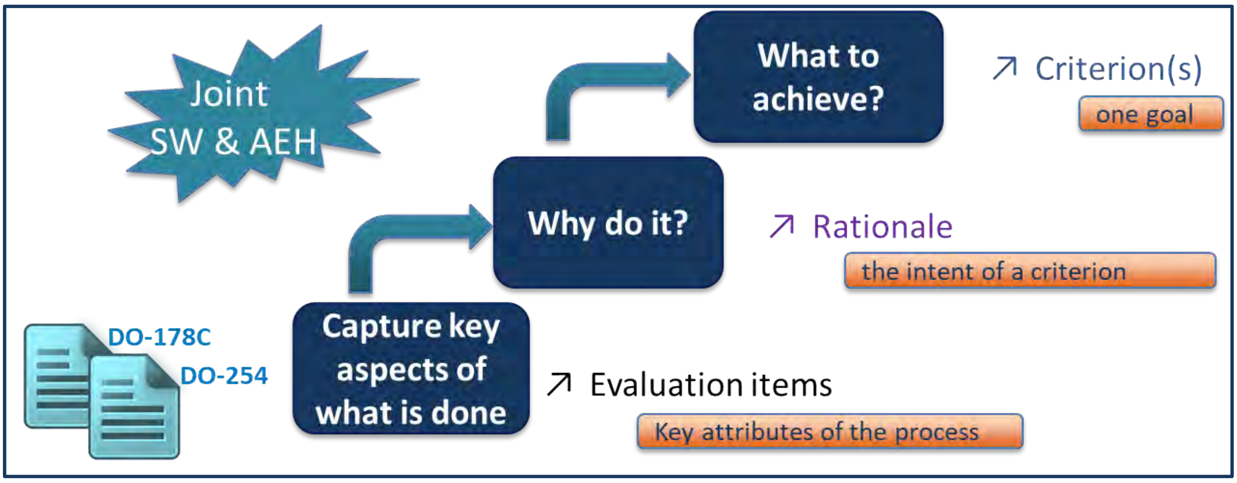


Figure 1 – Illustration of the applied methodology

Note on ED-12C/DO-178C supplements:

These supplements detail activities to be performed for specific techniques. Extracting goals into criteria for these individual techniques was not done for each detailed technique, but rather kept high level so as to enable assessment of other standards, having different set(s) of activities. For instance, for tools, the Abstraction Layer focused on formulating goals from generic standard documents (ED-12C/DO-178C) and not from ED-215/DO-330. For model-based development, the Abstraction Layer abstracted defined goals to address ED-218/DO-331 but together with ED-12C/DO-178C. For Object Oriented Technology (ED-217/DO-332) and Formal Methods (ED-216/DO-333), there were no specific analyses performed.

3.3 Safety levels

When all criteria and associated evaluation items were defined, safety levels were associated to each criterion and specifically to each evaluation item in order to reflect a safety continuum into the Abstraction Layer. Safety levels are defined in a similar manner as the current development assurance levels, keeping the naming convention A/B/C/D as defined in SAE ARP 4754A.

Additionally, safety levels were associated with the evaluation items using a methodology that considered the nature of errors that could occur in a development process and the contributions that the process makes in detecting and resolving the errors. An error typology was established with five types of errors that may occur in a development process, associated with two categories of means: those directly impacting that the item behaves as intended and safely, and those adding confidence. This methodology helps to understand why an evaluation item is to be addressed for the development at a given safety level.

Having established this typology of error types and the category of means, each evaluation item of every criterion was analyzed and attributed safety levels. To reflect at which safety levels the evaluation item is applicable, each evaluation item is preceded by the safety levels within brackets: [A/B/C/D].

The association of evaluation items to error typology and category of means per safety levels is identified in Appendix IV.

3.4 Independence

Independence has been defined in the Abstraction Layer from the concept of ED-12C/DO-178C and ED-80/DO-254. Independence is explicitly stated within the Abstraction Layer for verification and process assurance. The independence is also expected in a general manner for evaluation items related to “check” (see section 5.3), for safety levels A & B. This expectation is made general (following the concept of ED-80/DO-254) and the Task Force acknowledges the departure from the detailed allocation within ED-12C/DO-178C.

4 Introduction of the Abstraction Layer Criteria

4.1 Abstraction Layer Purpose

Inspired from the concepts for development assurance embodied within the ED-12C/DO-178C and D0-254/ED-80 standards, the Abstraction Layer provides criteria, each associated with rationale and evaluation items, to facilitate the assessment of other standards or publicly available methodologies, without prescribing the means used to satisfy the criteria.

The Abstraction Layer criteria applies to a standard or publicly available methodology that deals with the development process of hardware or software items, or collection of items.

The Abstraction Layer is based on the existence of requirements, specifying the intended function(s).

4.2 Structure of Abstraction Layer Criteria

Each criterion describes one intent of an overall development assurance process. Each criterion was developed with the intent to be SMART and process-centric (e.g., 'the process encompasses, ensures, allows, …"). Each criterion is worded in the present tense and, if possible, with a positive formulation.

The rationale provides the reason why the criterion is necessary to support development assurance.

Evaluation items describe attributes that the process ensures or provides; or, in some cases, assurance that other evaluation items are achieved. An evaluation item:

- Is dedicated to one single criterion.
- Is used to assess that a standard or methodology is described in sufficient detail and with sufficiently identified outcome evidence to fully satisfy the criterion.
- May contain multiple subparagraphs, each of which characterizes the evaluation item in more specific and measurable aspects.
- May also contain a 'note'. A note is sometimes helpful to illustrate the intent of the words of the evaluation item or provide additional information.

Some evaluation items, starting with the word "Check", state the expectation for a standard/methodology to have a process to confirm that another process is adequately performed. For example, there is a criterion that has an evaluation item stating "a process completely and correctly defines the item functions […]" and an evaluation item "Check" stating "A process provides a means to confirm the completeness and correctness of the captured requirements […]".

The information of the criterion is structured in the following manner. Each criterion is given a title and an identification number, e.g.:

1. Criterion text

Rationale

- text

Evaluation items

1. Text
   a. *[Safety Level xx]* Text
      i. Text

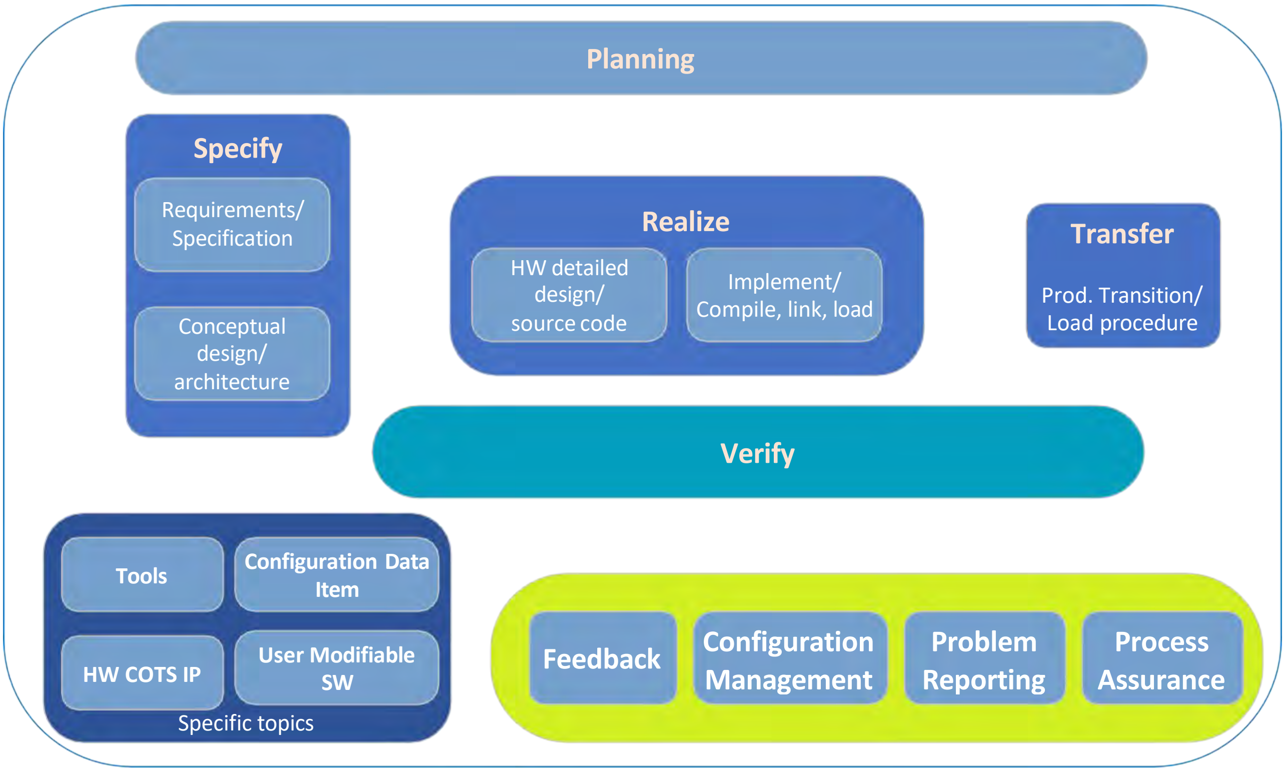
The indication for applicability to a safety level is added in front of each evaluation item in order to reflect a safety continuum into the Abstraction Layer. Safety levels are defined in a similar manner as the current development assurance levels, keeping the naming convention A/B/C/D as defined in SAE ARP 4754A. The convention is explained in section 5.2.

4.3 Introduction of the Abstraction Layer to the user

The Abstraction Layer is intended to be used to assess alternate standards or methodologies for development assurance of software and electronic hardware items. The alternate standard or methodology is expected to describe a set of processes that will be subject to the assessment against the Abstraction Layer criteria and associated evaluation items. A criterion is satisfied if all evaluation items associated with that criterion are assessed and fulfilled commensurate with the assigned safety level. Consequently, a criterion might be partially satisfied when one or more evaluation items are not fulfilled. In this case, gaps are identified, and the expectation is that a gap needs to be addressed and compensated for in order for the standard to be used in the development of safety related items.

4.4 Technical guidance to the reader

- **The generic development process:** To aid in achieving a common understanding of the Abstraction Layer criteria, the different criteria have been organized as generic development assurance processes. These processes are defined and relate to each other as depicted in the following Figure. This decomposition of processes is intended to better understand the framework while allowing different decomposition or organisation in the alternate standard or methodology and allowing assessment of each criterion in a stand-alone manner.



Figure– llustration of Generic development assurance process

- PLANNING refers to the process that consists of defining the processes and methodologies to be followed to develop the item.
- SPECIFY refers to the process that consists of defining both the intended functions of the item, expressed as requirements, and the conceptual design/architecture of the item. This means both the requirements that describe what to do (intent from system level) as well as how to do it. These two processes are grouped because it is understood that the conceptual design/architecture of the item supports the refinement of the requirements.
- REALIZE refers to the process that consists of creating the item in its final form from its requirements and the conceptual design/architecture by following transformation steps. For hardware, it covers the detailed design and implementation activities and for software the writing of the code and the generation of the executable object code.
- VERIFY refers to the verification processes of the item.
- TRANSFER refers to the process that consists of transferring the necessary information to production to correctly manufacture the item or correctly load the item.

- SPECIFIC TOPICS are processes for specific topics and need to be covered when a particular technology or technique is used and include:
  - TOOLS,

    - HW COTS IP: Hardware Commercial off-the-shelf Intellectual Property,
    - CONFIGURATION DATA ITEM,
    - USER MODIFIABLE SOFTWARE (UMS).
  - TRANSVERSE processes support all other development processes and include:
    - FEEDBACK,
    - CONFIGURATION MANAGEMENT,
    - CHANGE MANAGEMENT,
    - PROBLEM REPORTING,
    - PROCESS ASSURANCE.

- **Artefacts and Records:** While the criteria/evaluation items typically don't explicitly identify artefacts or documentation produced, an implicit expectation of a process is that it produces artefacts and records of activities, and that the artefacts and records are subject to appropriate configuration management for evidence of development assurance. It is understood that configured item(s) (cf. criteria CONFIGURATION MANAGEMENT and CHANGE MANAGEMENT) encompass artefacts and records.

- **Examples:** Some evaluation items provide examples. Examples are only to provide an illustration of certain process means but are not themselves to be considered as a recommended or sufficient approach, or as excluding other ways to address the same expectation.

5 Abstraction Layer Criteria for accepting alternative standards

5.1 Applicability

The Abstraction Layer criteria apply to a standard or publicly available methodology that deals with the development process of hardware or software items, or collection of items.

5.2 Safety levels

Safety levels are associated to each criterion and specifically to each evaluation item in order to reflect a safety continuum into the Abstraction Layer, and the gradual level of rigor in development assurance. Safety levels are defined in a similar manner as the current development assurance levels, as defined in SAE ARP 4754A, and commensurate with the classification of the failure condition(s) to which a hardware or software item contributes. The same naming convention A/B/C/D has been chosen for the Abstraction Layer, from A reflecting the most demanding development assurance level and D the lowest.

To reflect at which safety levels the evaluation item is applicable, each evaluation item is preceded by the safety levels within brackets: [A/B/C/D]. When an evaluation item is attributed a different set of safety levels for SW or AEH, there is another bracket to indicate the specific additional safety level and for which domain. The following examples illustrate how to interpret the safety level attribute:

- *[A/B/C]:* evaluation item is attributed to safety levels A, B and C for both SW & AEH domains.
- *[A/B/C][D(SW)]:* evaluation item is attributed to safety levels A, B and C for both SW & AEH domains, and to safety level D for SW.
- *[A/B/C][D(AEH)]:* evaluation item is attributed to safety levels A, B and C for both SW & AEH domains, and to safety level D for AEH.

5.3 Independence

Independence is a means to minimize the likelihood of errors that could occur when the same person or tool is used to perform a process step and the “Check” step, or to perform the REALIZE process and the VERIFY process.

Independence is expected in the following aspects:

a. The process ensures that ‘Check’ of an evaluation item is performed with independence with the evaluation item(s) for safety level A & B.
b. The process ensures that VERIFY process is achieved with independence from the REALIZE process.
c. The process ensures that PROCESS ASSURANCE is achieved with independence from other processes.

5.4 Criteria and rationale

This section lists the criteria associated to their rationale and evaluation items.

5.4.1 *Criteria for PLANNING*

1. *Criterion PLAN DEFINITION*
   A defined process ensures that plans are defined that can be consistently executed with the appropriate level of rigor.

Rationale

- To ensure the life cycle processes of the item are under control and repeatable.
- To ensure that the level of rigor of the planned processes is appropriate such that the item will be produced and will perform its intended function safely, correctly, and completely in its operating environment.
- To provide direction to the personnel performing the life cycle processes so that the life cycle processes are consistently applied.
- To identify evidence to be produced by the life cycle processes that will support the demonstration that processes are followed.

Evaluation

1. [*A/B/C/D*] The process ensures that plans address the entirety of the life cycle processes and artefacts to be produced.
2. [*A/B/C/D*] The process ensures that plans describe the processes and activities with sufficient detail to:
   a. Ensure the process can be consistently applied, regardless of the team member who applies it
   b. Demonstrate the level of rigor to achieve the required confidence
   c. Demonstrate the level of independence is achieved commensurate with the safety level.
3. [*A/B/C*][*D(AEH)*] The process ensures that plans describe the inter-relationships within and between processes including inputs, outputs, interdependencies, conditions for transitioning between processes, and feedback.
4. [*A/B/C*] The process ensures that plans define the standards, methods, environment, and tools that:
   a. Support uniform design and implementation
   b. Support error prevention and defect detection
5. [*A/B/C*] The process ensures that plans describe subcontracted activities and a method to ensure adherence to the plans.
6. [*A/B/C*] The process ensures that development and revision of plans are coordinated and controlled.
7. [*A/B/C*] The process ensures that plans define a method by which deviations from plans are identified and addressed.
8. [*A/B/C/D (*)*] Check PLAN DEFINITION: the process provides a means to confirm the plans cover evaluation items 1 through 7.

(*) Applicability of this “check” evaluation item depends on the safety levels of the evaluation items the “check” process is confirming.

2. *Criterion PLAN AGREEMENT*
   A defined process ensures that plans are communicated to and agreed with applicable stakeholders, with agreed deliverables.

   Rationale

   - To identify stakeholders.
   - To communicate and reach agreement on proposed life cycle processes and activities with applicable stakeholders.

   Evaluation

   1. [*A/B/C/D*] The process ensures that plans identify all applicable and appropriate stakeholders.
   2. [*A/B/C/D*] The process ensures plans are communicated to relevant stakeholders.
   3. [*A/B/C/D*] The process ensures plans are evaluated and agreed by stakeholders who need to follow them.
   4. [*A/B/C/D*] The process ensures deliverables needed to perform the process are coordinated and agreed by each stakeholder.

5.4.2 Criteria for SPECIFY

3. *Criterion SPECIFY REQUIREMENTS*
   A defined process ensures that a complete, correct, and detailed understanding of what the item is expected to do in its operating environment, is established and recorded.

   Rationale

   - To minimize the risk of unintended functionalities, mismatch with the system expectations and interface with other items.
   - To support the REALIZE and VERIFY processes (by defining what the item should do).

   Evaluation

   1. A process completely and correctly defines the item functions, their performance and interfaces from the system-level requirements allocated to the item(s), from the design constraints and from the consideration of the safety related aspects. In particular the process ensures that:
      a. *[A/B/C/D]* All allocated system requirements are transformed correctly into item requirements.
      b. *[A/B/C]* For software, the item requirements are further developed into sufficiently refined requirements (typically one or more tiers) in order to enable that software code can be directly produced from the requirements. A tiered approach may not be needed if software code can be directly produced from the item requirements.
      Note: requirements or tiered requirements for the item may be captured as a model.
      c. *[A/B/C/D]* For model-based development, textual requirements exist to specify the model or the model hierarchy.

  d. *[A/B/C]* When the AEH item uses a COTS device, the following have been transformed into requirements:
     i. the used functions of the COTS device
     ii. the device configuration
     iii. the deactivation means of unused functions of the COTS device
  e. *[A/B/C]* When the AEH item uses a HW COTS IP, the process ensures that requirements related to the allocated HW COTS IP functions are captured to an extent commensurate with the verification strategy (according to criterion DA FOR HW COTS IP (13)).
     In addition, item requirements are established to cover the configuration and control of both used and unused functions of the HW COTS IP.
  f. *[A/B/C/D]* The relationship between each allocated system requirement and corresponding item requirement(s) is established, and conversely, the relationship between each item requirement and corresponding allocated system requirement(s) is established.
  g. *[A/B/C]* When tiered requirements are developed, the relationship between item requirements and corresponding tiered requirements is established, and conversely, the relationship between tiered level requirements and corresponding item requirements is established.
  h. *[A/B/C][D(SW)]* Emerging functionality of the item is captured into requirement(s), and with rationale.
  i. *[A/B/C/D]* Captured requirements are unambiguous, complete, correct, consistent.
  j. *[A/B/C]* Captured requirements can be verified.

2. *[A/B/C/D]* The process ensures that requirements capture all functions of the item and consider the:
   a. Operating environment,
   b. Interface with other subsystems,
   c. Physical and resource constraints.
3. *[A/B/C]* The process ensures that standards exist and are followed to develop the requirements.
4. *[A/B/C/D] (*)* Check SPECIFY REQUIREMENTS: A process provides a means to confirm the completeness and correctness of the captured requirements, identifying omissions, errors and unintended functionality.
   Note 1: This includes tiered-requirements.
   Note 2: For model-based development, model simulation may be used to assess the model, only if simulation is based on the textual requirements specifying the model.
   (*) Applicability of this "check" evaluation item depends on the safety levels of the evaluation items the "check" process is confirming.

4. Criterion COORDINATE EMERGING FUNCTIONALITY

A defined process ensures that the emerging functionality of the item is coordinated with and agreed upon by all stakeholders and interfacing processes.

Rationale

- To minimize the risk of mismatch with the system expectations and with interfacing processes.
- To minimize the risk of unacceptable impact on safety.

Evaluation

1. *[A/B/C/D]* The process ensures the emerging functionality is defined in coordination and acceptability with:
    a. the system-level stakeholders
    b. safety assessment process
    c. other interfacing processes applicable to the item (e.g. cybersecurity, development of interconnected device, …)
2. *[A/B/C/D]* Check COORDINATE EMERGING FUNCTIONALITY: A process provides a means to confirm that the evaluation item 1 is met.

5. Criterion SPECIFY DESIGN

A defined process ensures that a conceptual design/architecture is defined consistently with the requirements and safety concerns with sufficient detail to develop the item.

Rationale

- To provide the item conceptual design/architecture with sufficient details for its realization, with the goal to meet the requirements.
- To capture and maintain knowledge of how the item is to be realized.
- To ensure the conceptual design/architecture does not adversely affect safety, is consistent with the requirements, and is compatible with the operating environment, available resources, and the item interfaces.

Evaluation

1. [*A/B/C/D*] The process ensures that conceptual design/architecture is defined from the requirements, and represents the high-level functional description (e.g. as functional block diagrams, design and architecture descriptions, external interfaces).
2. [*A/B/C*] The process ensures that conceptual design/architecture identifies internal components/blocks and their interfaces with sufficient details to develop the item, allocate to the functional architecture, and manage the impact of unused functions. Note: when the item is a CBA, an internal component/block and its interface is typically one device (or a group of devices or a part of one device).
    a. [*A/B/C*] Conceptual design/architecture considers constraints related to safety, including those necessary to address design errors and robustness defects.
    b. [*A/B/C*] For hardware, any implementation constraint, reliability, maintenance and test features are also identified.
    c. [*A/B*] When the item is a CBA, the selection of the COTS device considers its maturity and where risks are identified, a mitigation means is defined.
3. [*A/B/C/D*] When the item is a custom device and the conceptual design/architecture of the item allows the use of an HW COTS IP, the process ensures that criterion DA FOR HW COTS IP (13) and related evaluation items are considered.
4. When using a COTS device, the process ensures that:
    a. [*A/B/C/D*] there is a complete understanding of the COTS device behavior

b. [*A/B*] the unused functions of the COTS device do not adversely impact the used functions of the COTS device

c. [*A/B*] impact on the device functions in the event of inadvertent alteration of susceptible features (such as configuration, memory, registers) is assessed

d. [*A/B*] architectural means is defined to mitigate the inadvertent alteration

e. [*A/B/C/D*] when the device embeds a SW item that is not qualified within the device manufacturer qualification process or that is modified by the user, development assurance for the software item is proposed, in a manner commensurate with the usage of the COTS device.

5. [*A/B/C/D*] For hardware, the process ensures that conceptual design/architecture addresses the COTS device errata through an assessment of the errata to determine the errata applicable to the use of the device and to define a mitigation means where needed.
6. *[A/B][C(SW)]* The process ensures that design standard(s) exists and is followed to develop the conceptual design/architecture.
7. Check SPECIFY DESIGN:

a. [*A/B/C*] A process confirms the conceptual design/architecture is defined consistently with the requirements and safety concerns, identifying omissions and errors.

b. [*A/B/C/D (*)*] The process confirms that evaluation items 2 to 5 are fully addressed by the conceptual design/architecture.
Note: for model-based development, model simulation may be used to assess the model only if simulation is based on the requirements specifying the model.

c. *[A/B][C(SW)]* A process confirms the conceptual design/architecture conforms to the design standard.

(*) Applicability of this “check” evaluation item depends on the safety levels of the evaluation items the “check” process is confirming.

5.4.3 Criteria for REALIZE

6. *Criterion REQUIREMENTS REALIZATION*
   A defined process ensures that the item is correctly and completely realized from the requirements and the conceptual design/architecture into its final form, and can operate safely within the target operating environment.

Rationale

- To ensure that:
  - The item implements all requirements and conceptual design/architecture correctly and completely.
  - Interfaces are fully defined to support HW integration and SW integration.
  - The final form of the item is compatible with available resources.

Evaluation

1. [*A/B/C*] The process ensures that the hardware detailed design/software source code is completely and correctly developed from the item requirements and the conceptual design/architecture, following identified transformation steps.
2. [*A/B/C/D*] The process ensures that:
   a. The final form is developed from the hardware detailed design/software source code following identified transformation steps.
   b. The hardware item implements the hardware detailed design using representative manufacturing processes and defined design constraints.
3. [*A/B/C*] The process ensures that:
   a. For Hardware, Hardware-Software interfaces, Hardware-Hardware interfaces and electrical characteristics are defined and constrain the implementation.
   b. Design features and characteristics, including test features, are taken into account.
   c. Unused functions are identified and their safety impact is mitigated.
4. [*A/B*] [*C(SW)*] The process ensures that the relationship between each item requirement and corresponding hardware detailed design/software source code element(s) is established, and conversely, relationship between each hardware detailed design/software source code element and corresponding item requirement(s) is established.
5. [*A/B/C (*)*] Check REQUIREMENTS REALIZATION: the REALIZE process provides means for adding confidence in the generation of the final form, especially:
   a. Means to confirm that evaluation items 1 to 4 are met.
   b. Means for assessing compatibility with available resources

(*) Applicability of this “check” evaluation item depends on the safety levels of the evaluation items the “check” process is confirming.

7. *Criterion TRANSFORMATION CONTROL*
   A defined process ensures that appropriate intermediate transformation steps guarantee a sufficient level of control of the REALIZE activities, in order to detect transformation errors or

introduction of unforeseen features taking into account technology complexity, design complexity and safety-related aspects of the design.

Rationale

- To ensure the REALIZE activities are sufficiently controlled to support the detection of transformation errors or of inadvertently introduced features.

Evaluation

1. [*A/B/C*][*D(AEH)*] The process ensures that appropriate intermediate transformation steps exist to guarantee a sufficient level of control of the REALIZE process, where accessibility and observability are available.

2. [*A/B/C*] At each transformation step, the process includes means to prevent injection of typical errors and to ensure consistency in the process.

Example: use of standards for the REALIZE process

3. [*A/B/C*] The process provides means for checking that each transformation step output is correct relative to its inputs.
Note: checking of transformation steps may be combined when justified.

4. [*A/B/C*] The process provides means for detecting additional functionalities, unexpected or errant behavior introduced by each transformation step.

Examples: code review to design and coding standards, review of tool output reports, etc.

8. Criterion DOCUMENT FOR USE

For hardware, a defined process ensures that device functions and usage constraints are documented.

Rationale

- To ensure that the documentation and the usage constraints allow proper usage of the device and protection against undesired behavior or destruction of the device.

Evaluation

1. [*A/B/C/D*] The process ensures that interface description of the item, performance and electrical characteristics, limitations and usage constraints, are produced. This includes user documentation of HW/SW and HW/HW interfaces.

Examples: configuration description, power sequence (ASIC) and associated mitigation of unused functions (through configuration), etc.

2. [*A/B/C*] Check DOCUMENT FOR USE: the process provides a means for confirming that evaluation item 1 is met.

5.4.4 Criteria for VERIFY

For VERIFY process also refer to section 5.3 for independence aspects.

9. Criterion VERIFY REQUIREMENTS
A defined process ensures that the item is completely and correctly verified against its requirements, when the item functions in its operating environment.

Rationale

- To ensure the item completely fulfils and is robust with its requirements.
- To ensure inconsistencies between the item implementation and its requirements, including the interfaces, is detected and assessed.
- To detect remaining errors not discovered during the capture of the requirements.

Evaluation

1. [*A/B/C/D*] A defined process ensures that requirements-based verification means are developed to verify the item. The process ensures:
   a. Verification completely covers the requirements.
   b. Verification completely covers SW/SW and HW/SW interfaces.
   c. Appropriate methods are defined to verify the item under normal/abnormal operating conditions, considering:
      i. Environment
      ii. Procedures
      iii. Resources/means
      iv. Analysis
2. [*A/B/C/D*] The process ensures that verification is performed on the target device or in a representative environment when justification is provided.
3. [*A/B/C/D*] The process ensures that any intermediate representation of the item, used to demonstrate correctness of the item, is correct.
   a. The process ensures that the limits of such verification activities are identified.

   Note: for model-based development, if model simulation is used to verify the item, simulation cases are developed based on the layer of requirements specifying the model.
4. [*A/B/C*] For HW COTS IP, the process ensures the verification of the HW COTS IP after the transformation steps performed by the device developer (e.g. synthesis/place and route) is correct.
5. [*A/B/C*] For HW COTS IP, the process ensures that verification covers the integrated HW COTS IP functions within the device in appropriate test environment.
6. [*A/B/C/D*] The process ensures that its outputs are described to a level of detail that allows:
   a. the assessment of the requirements-based verification cases, the means by which they are satisfied, and of the verification results,
   b. repeatability of the verification.

7. [*A/B/C/D*] The process ensures that the relationship between each item requirement (including tiered) and corresponding verification artefact(s) is established, and

conversely, the relationship between each verification artefact and corresponding item requirement(s) is established.

8. Check VERIFY REQUIREMENTS:
   a. [*A/B/C/D*] The process provides an effective method to detect and provide feedback for any deficiency to the appropriate process.
   b. [*A/B/C/D*] The process confirms the correctness and completeness of the relationship between item requirements and verification artefacts.
   c. [*A/B*][*C(SW)*] the process ensures the detection of a lack of coverage of any requirement during the execution of requirement based verification.
   d. [*A/B/C/D*] The process proposes a method to ensure that verification procedures correctly implement the verification cases.
9. [*A/B*] The process ensures verification is performed with independence to REALIZE activities (as per 5.3) to enforce the correctness demonstration.

10. Criterion DETECT UNINTENDED FUNCTION

A defined process ensures that each element of the item has been verified through requirement-based verification, to preclude unexpected or undefined behavior.

Rationale

- To establish an objective criterion for item verification completeness (different from the completeness of requirements verification).
- To preclude unexpected behavior of the item.

Evaluation

1. [*A/B*][*C(SW)*]The process ensures that a well defined method (like structural/elemental analysis) is used and fulfills the following aspects:
   a. For Hardware: the part of the item to be covered by elemental analysis is defined and the definition of elements is at a level appropriate with the objective to detect unintended hardware function.
   b. The method and associated criteria to measure the coverage of the item elements during requirement-based verification is defined and is appropriate to the type of elements.
   c. The results are analyzed and each gap identified is either justified or fed back to the appropriate process, particularly:
      i. elements 'not covered by requirement-based verification' are justified (the unused functions are identified with their deactivation means),
      ii. or feedback is provided to the REALIZE process to remove unintended function,
      iii. or feedback is provided to the VERIFY process.
2. [*A/B*][*C(SW)*]The process provides a means ensuring that any intermediate representation of the item does not lead to implementation of any unintended function that impacts safety functions *(e.g. detection of a feature in the implementation that is not supported by requirement through structural coverage, etc)*.

5.4.5 *Criteria for TRANSFER*

11. Criterion TRANSFER

A defined process ensures that necessary information is provided for the production of the item or loading of the item into its target.

Rationale

- To transfer all necessary design information for production of the equipment implementing the item.
- To provide control means ensuring that the item is correctly loaded into the system or equipment with appropriate safeguards (e.g., compatibility with the airborne system or equipment, specific loading procedures or tools, security keys, etc).
- To ensure, when the item is loadable, that the item identification can be retrieved by a defined means.

Evaluation

1. [*A/B/C/D*] The process provides means to transfer to production all necessary design data to manufacture the item or load the item.
    a. The process provides means for the recording and release of instructions for item load and configuration identification.
    b. The process ensures that, if some parameters are intended to be configured or generated in production, appropriate procedures, methods and tools are documented.
2. [*A/B/C/D*] The process ensures that key attributes of the item and data required for the test in production are identified and provided.
3. [*A/B/C/D*] Check TRANSFER: a process confirms that evaluation items 1 and 2 are correct and complete.

5.4.6 Criteria for SPECIFIC TOPICS

12. Criterion TOOLS

A defined process ensures that a means is provided to demonstrate confidence in a tool such that the tool use does not compromise the integrity of the item or its verification.

Rationale

- To provide confidence that a tool used for item realization or verification will not introduce errors into the item, or fail to detect errors in the item.

Evaluation

1. [*A/B/C/D*] The process provides a means for assessing the use of the tool and its impact on item realization or verification to allow the determination whether qualification of a tool is required:
   a. The process requires either tool qualification or requires that the assessment of the tool outputs detects, with an independent means, any potential error that the tool could introduce into the design or fail to detect during verification, with supporting evidence.
2. [*A/B/C/D*] The process ensures that, when tool qualification is required, the objectives and activities to qualify the tool are defined in a manner commensurate to the safety risk introduced with the tool usage (e.g. fail to detect errors, introduction of errors into the item, etc)

   Note 1: For instance, a tool that can cause introduction of error (REALIZE process tool) requires a higher qualification than a tool can fail to detect an error (VERIFY process tool)

   Note 2: the Abstraction Layer doesn't further detail the means for evaluating other standards for tool qualification, the expectations are to obtain an equivalent credit to the applicable standard in the aviation domain.
3. [*A/B/C/D*] The process ensures that the tool usage complies with the determined tool environmental and functional constraints.

Note: the evaluation items 1 to 3 of the criterion TOOLS include tools used to perform implementation and/or verification steps for HW COTS IP.

13. Criterion DA FOR HW COTS IP

When the conceptual design/architecture of the item (custom device) allows the use of HW COTS IP, the process ensures that a selection of the HW COTS IP for its intended function and an assessment of the HW COTS IP are performed, and that development activities exist to mitigate the associated risk of development errors and for the correct usage and integration of the HW COTS IP in the device.

Rationale

- To mitigate the risk of development errors associated with the lack of access to the development process of the HW COTS IP.

Evaluation

1. [*A/B/C/D*] A process ensures that the selection of the HW COTS IP checks at least that the following points are met:
   a. The IP is technically suitable for implementing the intended function.
   b. The description of the HW COTS IP architecture or IP design concept allows an understanding of the functionality, modes, and configuration of the IP. The description also includes an understanding of the source format or combination of source formats of the HW COTS IP.
   c. The availability and quality of data and documentation enable the integration and verification of the HW COTS IP (e.g. datasheets, application notes, user guide, knowledge of errata, etc.).
   d. The configurations, selectable options, and scalable modules of the HW COTS IP design are documented and information to support that the implementation of the HW COTS IP can be properly managed (e.g. synthesis constraints, usage and performance limits, physical implementation, and routing instructions) is provided.
2. HW COTS IP ASSESSMENT - A process ensures that the HW COTS IP provider and the associated data of the HW COTS IP are assessed based on at least the following points:
   a. [*A/B/C/D*] The HW COTS IP development and verification process satisfies at least the Abstraction Layer criteria SPECIFY REQUIREMENTS (3), SPECIFY DESIGN (5), VERIFY REQUIREMENTS (9);
      i. The verification of the HW COTS IP covers the specific use case for the HW COTS IP.
   b. [*A/B*] The HW COTS IP verification process satisfies the criterion DETECT UNINTENDED FUNCTION (10).
   c. [*A/B/C/D*] The known errors and limitations are available to the IP user, and there is a process to provide updated information to the IP user.
   d. [*A/B/C*] The HW COTS IP has service experience data that shows reliable operation for the specific use case for the HW COTS IP.
3. [*A/B/C/D*] When the HW COTS IP ASSESSMENT reveals that some evaluation points cannot be completely met using the IP provider’s data, a process ensures that an appropriate set of development assurance activities is defined to mitigate the associated risk of development errors, and specifically covers the requirement-based verification strategy for the HW COTS IP.
4. [*A/B/C/D*] A process ensures that an appropriate set of development assurance activities, including verification strategy, is defined to cover the design integration and the correct usage of the HW COTS IP within the custom device.

Configuration Data Item

The following criterion addresses Configuration Data item that is developed separately from the item it configures. The criterion regroups by itself evaluation items related to steps in the process already defined through other criteria, and it applies when generating the Configuration Data item.

For the item using the configuration data, the evaluation items within CONFIGURATION DATA ITEM criterion are additional to those already listed in the other criteria, and those evaluation items are marked with "(i)" at the end.

For the Configuration Data item, only the criterion CONFIGURATION DATA ITEM is applicable, and only those evaluation items not marked with "(i)".

14. Criterion CONFIGURATION DATA ITEM

When a Configuration Data item is planned to be developed and verified separately from the item using it, a defined process ensures that the Configuration Data are developed and verified according to the constraint defined by the item using those Configuration Data.

Rationale

- To ensure that the life cycle data of the item using the Configuration Data accounts for an independent development of the Configuration Data.
- To ensure the item using the Configuration Data has defined constraints for the development and verification of the Configuration Data.
- To ensure that the development and verification of the Configuration Data item is consistent with the constraint defined by the item using them.
- To ensure the compatibility of the Configuration Data item with the item using it can be verified/assessed.

Evaluation

SPECIFY

1. [*A/B/C/D*] The process ensures that requirements exist defining the dependencies and interactions between the Configuration Data item and the item using it. These dependencies and interactions may include the following considerations:
    a. Requirements exist defining structure, attributes, and set of acceptable values of the Configuration Data (i).
    b. Requirements define constraint by which the item processes the configuration data, and addresses both normal usage and robustness aspects (i).
    c. Means for defining the actual value of a Configuration Data item and checking for completeness and correctness.

REALIZE

2. [*A/B/C/D*] The process ensures that the Configuration Data item is developed according to its requirements and identified transformation steps consistent with the defined constraints.

VERIFY

3. [*A/B/C/D*] The process ensures that the item using the Configuration Data is verified for robustness and for normal behavior resulting from values of the Configuration Data item (i).
4. [*A/B/C/D*] The process ensures that the Configuration Data item is verified for completeness and correctness.
   Note: the verification may be addressed either by checking the completeness and correctness of the Configuration Data item and/or by test activity with the item using the Configuration Data.
5. [*A/B/C*] The process ensures that each element of the Configuration Data item is covered by the verification.

PLANNING

6. [*A/B/C/D*] The process ensures that planning documentation of the item addresses the specific aspect of the Configuration Data item (i).
7. [*A/B/C/D*] Criteria PLAN DEFINITION (1) and Criteria PLAN AGREEMENT (2) apply for Configuration Data items.

TRANSFER

8. [*A/B/C/D*] The process defines procedures, methods and tools and all necessary data required for the independent development of the Configuration Data item (i).

CONFIGURATION MANAGEMENT

9. [*A/B/C/D*] Criteria CONFIGURATION MANAGEMENT (17) and CHANGE MANAGEMENT (18) apply ensuring an independent Configuration Data item life cycle management.
10. [*A/B/C/D*] The process ensures that Configuration Data item states a clear compatibility reference to the item(s) using it.

FEEDBACK, PROBLEM REPORTING, PROCESS ASSURANCE and TOOLS

11. [*A/B/C/D*] Criteria TOOLS (12), FEEDBACK (16), PROBLEM REPORTING (19) and PROCESS ASSURANCE (20) apply.

15. Criterion USER MODIFIABLE SOFTWARE

When an item has a user-modifiable capability, a defined process ensures that operation of the non-modifiable software components is protected from adverse effect by the user-modifiable capability.

Rationale

- To ensure that user modifications of the item performed after certification remain within the scope assessed at the time of certification.
- To ensure that user modification, although not constrained by a rigorous process, cannot affect the non-modifiable portion of the item and its safety related capability.

Evaluation

PLANNING & SPECIFY

1. [*A/B/C/D*] The process ensures that the capability influenced by the user modifiable software does not affect the safety related functions.
2. [*A/B/C/D*] The process ensures that a means is defined to protect non-modifiable functionality from adverse effect(s) by the user modifiable software.
   a. When protection means is provided with the support of a tool, the criterion TOOLS (12) applies.

   Note: if the protection means is provided by software or hardware, the protection means is expected to be developed at minimum to the same safety level as the non-modifiable software component(s).

TRANSFER

3. [*A/B/C/D*] The process ensures that means and constraints for modifying the user modifiable software are identified and available to user(s).

5.4.7 Criteria for TRANSVERSE PROCESSES

16. Criterion FEEDBACK

A defined process ensures coherency between processes.

Rationale:

- To ensure coherency between processes.
- To ensure that any omissions, inadequacies, or errors detected are adequately addressed by the relevant process(es).
- To ensure improvement and refinement of the SPECIFY process outputs.

Evaluation

1. [*A/B/C/D*] The process provides a method to ensure that:
   a. Inadequate or incorrect inputs detected during a life cycle process are fed back to the appropriate process(es) for clarification and/or correction.
   b. Inadequate or incorrect inputs, design decision(s) and information detected during a development process leading to definition of new requirements or refinement of requirements are fed back to the SPECIFY process(es). This includes emerging functionalities.
2. [*A/B/C/D*] The process ensures that the feedback is appropriately addressed.

17. Criterion CONFIGURATION MANAGEMENT

A defined process ensures the identification, retention and retrieval of the configured item and its life cycle data for certification credit and release, consistent with the lifecycle processes.

Rationale:

- To ensure identification and control of each configured item and its life cycle data, in relationship with the lifecycle process.
- To support coherency between configured item(s) and its life cycle data.
- To ensure consistent and accurate replication and retrieval of each configured item, including tools, and any other data, where configuration is essential.

Evaluation

1. [*A/B/C/D*] The process ensures that each configured item and its lifecycle data are uniquely identified.
2. [*A/B/C/D*] The process ensures that status information supports the management of configured items and its life cycle data, consistent with the development process, using defined and repeatable process steps.
3. [*A/B/C/D*] The process ensures that there is a method to ensure coherency between the life cycle data of the configured item(s) (e.g. through the establishment of baselines).
4. [*A/B/C/D*] The process ensures that status information is recorded to enable reporting of configuration management status, definition of where data will be kept, how it will be retrieved for reporting, and when it will be available.
5. [*A/B/C/D*] The process ensures that all the data needed to replicate the hardware/software item released for certification and production are under configuration control, including means to regenerate/verify.

18. *Criterion CHANGE MANAGEMENT*

A defined process ensures the control of the changes to configured items.

Rationale

- To track and manage changes.
- To provide means of preserving integrity.

Evaluation

1. [*A/B/C/D*] The change control process ensures the management of:
   a. What the changes are for,
   b. Which configured items are subject to change,
   c. How changes are controlled for correctness and consistency,
   d. Tracking and traceability of changes for PLANNING, SPECIFY, REALIZE, [only for A/B] VERIFY cases and procedures, over the life cycle.
2. [*A/B/C/D (*)*] Check CHANGE MANAGEMENT - A process confirms the change has been incorporated and that evaluation item 1 is met.

   (*) Applicability of this “check” evaluation item depends on the safety levels of the evaluation items the “check” process is confirming.

19. *Criterion PROBLEM REPORTING*

A defined process ensures that any deficiencies found in configured life cycle data of the item or process non-compliance with plans and standards are recorded, assessed and appropriately addressed.

Rationale

- To provide means to capture and address any deficiencies detected within any process, and report to the appropriate process(es).

Evaluation:

1. [*A/B/C/D*] The process provides an effective method to record and address any deficiency for any life cycle process and is implemented no later than the establishment of the configuration baseline(s) from which certification credit is to be obtained.
2. [*A/B/C/D*] The process ensures that each deficiency and its impact are assessed.
3. [*A/B/C/D*] The process ensures that each deficiency is addressed such that
   a. Either
      i. a resolution is defined, including a description of any required corrective action, or
      ii. a mitigation means is provided, or
      iii. a justification is provided for retaining the deficiency without the need of any further actions.
   b. Evidence is provided that the described corrective actions and/or mitigations have been taken.
   c. The disposition status of the deficiency is identified.

20. Criterion PROCESS ASSURANCE

A process provides independent assurance that agreed plans are adhered to and that any deviations are identified and accepted.

Rationale

- To independently provide confidence that the life cycle processes are consistently conducted, and their outputs produced, in conformance with applicable/approved plans and standards.
- To evaluate any deviations from the approved plans and standards in order to ensure the acceptability of those deviations.

Evaluation

1. [*A/B/C*][*D(SW)*] The process provides means to independently assure that life cycle processes, including transitions, are consistently conducted, in conformance with applicable/approved plans and standards.
2. [*A/B/C*][*D(SW)*] The process provides means to independently assure that life cycle processes outputs are produced in conformance with applicable/approved plans and standards.
   Note: one acceptable effective means could be sampling of the life cycle process outputs, but it is not the intent that process assurance repeat the entire life cycle processes.
3. [*A/B/C*][*D(SW)*] The process provides means to assure that deviations from applicable/approved plans and standards are detected, recorded, evaluated, tracked and accepted.
4. [*A/B/C*][*D(SW)*] The process provides means to assure that:
   a. The life cycle processes and data, including data to build the item, are complete, and
   b. The item is built for transfer to production in conformance with the associated life cycle data.
5. [*A/B/C*][*D(SW)*] The process provides means to assure that any defects identified by process assurance are resolved.
6. [*A/B/C*][*D(SW)*] The process provides means to assure that process assurance activities are recorded.

6 Abstraction Layer User Guide

The Task Force has gathered experience gained during a trial application of the Abstraction Layer on a public alternate standard into a User Guide, describing “how to use the AL”and recommendations for performing future assessments. This User Guide is not binding guidance in any way but represents consensus best practices.

The User Guide is available in Appendix V of this document.

7 Conclusion

This document represents a consensus opinion of the Task Force members comprised of representatives from industry and certification authorities.

The Abstraction Layer is intended to be a ‘bridging tool', a set of 20 criteria to assess potential alternate standards or public methodologies, used in other industry domains. It may also facilitate introduction of novel technologies by enabling the assessment of other development assurance standards.

As recommended by the Task Force, the Abstraction Layer is accompanied by a User Guide describing ‘how to use the Abstraction Layer’ material.

The Abstraction Layer is not intended to serve as some kind of new alternative standard. It does not intend to invalidate or put at risk the current development assurance practices. It does not mirror what the currently acceptable standards already define.

APPENDIX I: Definitions and Acronyms

Definitions

Abstraction Layer	A complete set of criteria used to assess development standards or methodologies for their use in complying with the applicable aircraft systems and equipment safety regulations.
Acceptance test	A test to demonstrate that the manufactured, modified or repaired product performs in compliance with the key attributes of the unit on which certification is based.
Activity	A documented and repeatable step in a process used to produce results.
Architecture (ED-12C/DO-178C)	The structure of the software selected to implement the software requirements.
Check	An evaluation item which ensures other evaluation criteria are met.
Collection of items	A set of items which may be composed of software and/or hardware elements having bounded and well-defined interfaces.
Conceptual design (ED-80/DO-254)	A high-level design concept that may be assessed to determine the potential for the resulting design implementation to meet the requirements.
Conceptual design/Architecture	These two terms are concatenated in the Abstraction Layer to refer to a common step in the process named conceptual design for hardware and architecture for software.
Criterion	A description of one intent of an overall development assurance process. *Note: the criterion should be SMART: Specific, Measurable, Achievable, Realistic, and Tangible.*
Custom device	A hardware item developed for a specific usage. For instance, programmable logic devices (PLDs), field-programmable gate arrays (FPGAs), or application-specific integrated circuits (ASICs), are referred to as “custom devices”.
Development Assurance Level (iDAL ED-79A/ARP-4754A)	The level of rigor of development assurance tasks performed on Item(s).
Emerging functionality	Additional functionality of the item emerging from the SPECIFY process, or REALIZE process, and which doesn’t come from the system-level requirements (e.g., derived requirements).
Evaluation item	Attributes to support the evaluation of the notion of ‘goodness’ of the process, i.e., supporting information for determining or evaluating process performance or if the level of confidence brought by the process is acceptable.
Final form (of the item)	Final form refers to the executable object code (EOC) for the software item, and to an ASIC device or FPGA device programmed with bitstream.
Hardware Commercial Off-The-Shelf Intellectual Property (HW COTS IP)	Electronic hardware design functions or modules previously developed with a methodology other than the custom device development process and used to design and implement a part of a custom device. Intellectual property is considered to be ‘HW COTS IP’ when it is a commercially available function used by a number of different users in a variety of applications and installations. HW COTS IP is available in various source formats: Soft IP, Firm IP, Hard IP.
Hardware detailed design	Intermediate representation of the hardware design, i.e., High level Design Language (HDL) source code or schematic.

Independence (Modified from ED-80/DO-254)	Separation of responsibilities which ensures the accomplishment of objective evaluation either by someone or by something, other than those used to produce the data, or other than those used to perform the process.
Integrity (ED-12C/DO-178C)	An attribute of the system or an item indicating that it can be relied upon to work correctly on demand.
Item (ED-79A/ARP475A)	A hardware or software element having bounded and well-defined interfaces.
Methodology	A set of activities that results in an item or collection of items to be submitted for approval. *Note: in the context of the Abstraction Layer, the description of a methodology should be available in the public domain.*
Rationale	The rationale describes the purpose of the criteria with the intent to facilitate their understanding.
Requirement	An identifiable element describing the behavior of the intended function(s) and attribute(s) of an item (or of a system).
Tier	A step-in refinement of the item requirements.
Tool qualification	The process to provide evidence that potential failures of a tool does not adversely affect the tool output in a safety related manner that is undetected by technical and/or organizational measures outside the tool.
Transformation step	One process step contributing to generate the item into its final representation from the item requirements. Example: Successive transformation steps are performed in order to produce a hardware or software item (e.g., software: code generation, compilation; hardware: synthesis, test features addition (SCAN, JTAG, …), place&route, etc.).
User modifiable software (UMS)	User modifiable software is designed to be modified by its users, if the system requirements provide for user modification. A user modifiable component is that part of the software that is intended to be changed by the user and a non-modifiable component is that part which is not intended to be changed by the user.
Unintended functionality	Unexpected/ undesired additional functionality of the item.
Verification (ED-80/DO-254)	The evaluation of an implementation of requirements to determine that they have been met. The evaluation of the outputs of a process to ensure correctness and consistency with respect to the inputs and standards provided to that process.
Verification artefact	This term refers to data that are used as evidence of the verification process (e.g., analysis, simulations, test cases/procedures, results, …).

Acronyms

AC	Advisory Circular
AEH	Airborne Electronic Hardware
AMC	Acceptable Means of Compliance
A(M)C	Reference to harmonized EASA AMC & FAA AC documents
ANAC	Agência Nacional de Aviação Civil
ARP	Aerospace Recommended Practice, SAE document
ASIC	Application-Specific Integrated Circuit
ATF	Assessment Task Force
CBA	Circuit Board Assembly
COTS	Commercial Off The Shelf
DA	Development Assurance
DAL	Development Assurance Level
EASA	European Aviation Safety Agency
FAA	Federal Aviation Administration
FPGA	Field-Programmable Gate Array
GM	Guidance Material
HDL	Hardware Description Language
HW	Hardware
IMA	Integrated Modular Avionics
IP	Intellectual Property
MBD	Model-Based Development
PLD	Programmable Logic Device
SMART	Specific, Measurable, Achievable, Realistic, Tangible
SME	Subject Matter Expert
SW	Software
TCCA	Transport Canada Civil Aviation
TF	Task Force
TFAL	Task Force Abstraction Layer
UAS	Unmanned Aircraft Systems
UMS	User Modifiable Software

APPENDIX II: Reserved

APPENDIX III: References

Ref.	Name	Title
1.	EUROCAE ED-80	Design Assurance Guidance for Airborne Electronic Hardware
2.	RTCA DO-254	Design Assurance Guidance for Airborne Electronic Hardware
3.	EUROCAE ED-12C	Software Considerations in Airborne Systems and Equipment Certification
4.	RTCA DO-178C	Software Considerations in Airborne Systems and Equipment Certification
5.	EUROCAE ED-215	Software Tool Qualification Considerations
6.	RTCA DO-330	Software Tool Qualification Considerations
7.	EUROCAE ED-216	Formal Methods Supplement to ED-12C and ED-109A
8.	RTCA DO-333	Formal Methods Supplement to DO-178C and DO-278A
9.	EUROCAE ED-217	Object-Oriented Technology and Related Techniques Supplement to ED-12C and ED-109A
10.	RTCA DO-332	Object-Oriented Technology and Related Techniques Supplement to DO-178C and DO-278A
11.	EUROCAE ED-218	Model-Based Development and Verification Supplement to ED-12C and ED-109A
12.	RTCA DO-331	Model-Based Development and Verification Supplement to DO-178C and DO-278A
13.	EUROCAE ED-79A	Guidelines for Development of Civil Aircraft and Systems
14.	SAE ARP 4754A	Guidelines for Development of Civil Aircraft and Systems
15.	EASA AMC 20-115D	Software Considerations for Certification in Airborne Systems and Equipment
16.	FAA AC 20-115D	Airborne Software Development Assurance Using EUROCAE ED-12() and RTCA DO-178()
17.	EASA AMC 20-152A	Development Assurance for Airborne Electronic Hardware (AEH)
18.	FAA AC 20-152A	Development Assurance for Airborne Electronic Hardware (AEH)

APPENDIX IV: Safety levels

Safety levels were associated to each criterion and specifically each evaluation item in order to reflect a safety continuum into the Abstraction Layer, and the gradual level of rigor in development assurance. Safety levels are defined in a similar manner as the current development assurance levels, as defined in SAE ARP 4754A, and commensurate with the classification of the failure condition(s) to which a hardware or software item contributes. The same naming convention A/B/C/D has been chosen for the Abstraction Layer, from A reflecting the most demanding development assurance level and D the lowest.

Additionally, safety levels were associated with the evaluation items using a methodology that considered the nature of errors that could occur in a development process and the contributions that the process makes in detecting and resolving the errors. An error typology was established with five types of errors that may occur in a development process:

- Transformation error:
  error due to the transformation of an input data into an output data (e.g., from system requirements to software requirements).
- Emerging/Unintended error:
  error that may affect the intended function due to introduction of emerging function or unintended function.
- Global error:
  error within the creation of an output data itself (e.g., consistency between requirements, between source files, etc.).
- Local error:
  error on a single element of an output data (e.g., unverifiable requirement, etc.).
- Interface error:
  HW/SW, HW/HW, and/or SW/SW interface error due to misinterpretation of the interface or a missing interface characteristic. It also covers user interface errors.
- All errors:
  Classification attributed to some evaluation items when the analysis led to attribute to all error types and not a particular one.

In combination with errors, a category of means was identified to ensure that a development item met its intended function with safety. While an item development may give rise to different types of errors, the various development processes provide either one of two different means to address errors:

- Directly-Impacting: those means which provide direct evidence to show that a development item behaves as intended and safely.
- Confidence-adding: those means which add confidence that provided evidence is sound.

This methodology helps to understand why an evaluation item is to be addressed for the development at a given safety level.

Having established this typology of error types and the category of means, each evaluation item of every criterion was analyzed and attributed safety levels. In some cases, the association identified differences in impact to error for SW and AEH made by an evaluation item. For example, evaluation item (SPECIFY REQUIREMENTS (3), 1.h) “Emerging functionality of the item is captured into requirement(s), and with rationale”. The analysis considered the evaluation item to be an

“Emerging/Unintended Error” type whose category was “Directly-Impacting” and associated with applicability of DAL A/B/C for AEH, and DAL A/B/C/D for SW.

The evaluation items were analyzed per criterion and the result of the analysis of type of errors, category of means and the applicability to safety levels is provided hereunder (one table for each criterion).

For this detailed analysis, ED-12C/DO-178C, A(M)C 20-115D, ED-80/DO-254 and A(M)C 20-152A were used.

The following convention for the table has been used:

- In each cell of the table, the number(s) corresponds to the reference of an evaluation item of the criterion.
- When this number is preceded by SW or AEH, it means that this evaluation item is classified differently for the SW domain, and for the AEH domain. This is reflected in the Abstraction Layer, using a prefix in front of the identifier of the evaluation item: e.g., SW, respectively AEH.
- If an evaluation item is explicitly for hardware only or for software only through the content of the evaluation text, the table doesn’t repeat this SW or AEH attribution.

1. PLAN DEFINITION

ERROR TYPE	MEANS	SAFETY LEVEL		
		A & B	A & B & C	A & B & C & D
All	Directly-impacting/ Confidence-adding		4, 5, 6 & 7 SW 3	1, 2, 8* AEH 3

*Applicability of this “check” evaluation item depends on the safety levels of the evaluation items the “check” process is confirming.

2. PLAN AGREEMENT

ERROR TYPE	MEANS	SAFETY LEVEL		
		A & B	A & B & C	A & B & C & D
All	Directly-impacting/ Confidence-adding			1, 2, 3, 4

3. SPECIFY REQUIREMENTS

ERROR TYPE	MEANS	SAFETY LEVEL		
		A & B	A & B & C	A & B & C & D

Transformation	Directly-impacting		1.b, 1.d, 1.e	1.a, 1.c, 1.i, 2.c
	Confidence-adding		1.g	1.f
Emerging/unintended	Directly-impacting		1.d, 1.e	1.h
Global	Directly-impacting		1.b	1.i
Local	Directly-impacting		3, 1.j	
Interface	Directly-impacting			2
All	Confidence-adding			4*

* Applicability of this “check” evaluation item depends on the safety levels of the evaluation items the “check” process is confirming.

4. COORDINATE EMERGING FUNCTIONALITY

ERROR TYPE	MEANS	SAFETY LEVEL		
		A & B	A & B & C	A & B & C & D
Emerging/unintended	Directly-impacting			1
	Confidence-adding			2

5. SPECIFY DESIGN

ERROR TYPE	MEANS	SAFETY LEVEL		
		A & B	A & B & C	A & B & C & D
Emerging/unintended	Directly-impacting	4.b, 4.c, 4.d		5
Global	Directly-impacting	2.c	2, 2.a, 2.b	1, 4.a
	Confidence-adding		6	
Local	Confidence-adding		6	
Interface	Directly-impacting	2.c, 4.c	2	1, 5
All	Directly-impacting			3, 4.e
	Confidence-adding		7.a, 7.c	7.b*

* Applicability of this “check” evaluation item depends on the safety levels of the evaluation items the “check” process is confirming.

6. REQUIREMENTS REALIZATION

ERROR TYPE	MEANS	SAFETY LEVEL		
		A & B	A & B & C	A & B & C & D
Transformation	Directly-impacting		1, 3.a, 3.b	2
	Confidence-adding	AEH 4	SW 4	
Emerging	Directly-impacting		3.c	
Local	Confidence-adding			
All	Confidence-adding		5*	

* Applicability of this “check” evaluation item depends on the safety levels of the evaluation items the “check” process is confirming.

7. TRANSFORMATION CONTROL

ERROR TYPE	MEANS	SAFETY LEVEL		
		A & B	A & B & C	A & B & C & D
Transformation	Directly-impacting		SW 1	AEH 1
	Confidence-adding		3	

Emerging/unintended	Directly-impacting		4	
All	Confidence-adding		2	

8. DOCUMENT FOR USE

ERROR TYPE	MEANS	SAFETY LEVEL		
		A & B	A & B & C	A & B & C & D
Interface	Directly-impacting			1
	Confidence-adding)		2	

9. VERIFY REQUIREMENTS

ERROR TYPE	MEANS	SAFETY LEVEL		
		A & B	A & B & C	A & B & C & D
Transformation	Directly-impacting		4	1.a, 1.c, 3
Emerging	Directly-impacting		4, 5	1.a, 1.c, 3
Local	Directly-impacting		4, 5	1.c, 3
Interface	Directly-impacting		4, 5	1.b, 2, 3
All	Confidence-adding	AEH 8.c, 9	SW 8.c	6, 7, 8.a, 8.b, 8.d

10.DETECT UNINTENDED FUNCTION

ERROR TYPE	MEANS	SAFETY LEVEL		
		A & B	A & B & C	A & B & C & D
Emerging/unintended	Directly-impacting	AEH 1, AEH 2	SW 1, SW 2	

11.TRANSFER

ERROR TYPE	MEANS	SAFETY LEVEL		
		A & B	A & B & C	A & B & C & D
Interface	Directly-impacting			1, 2, 3

12.TOOLS

ERROR TYPE	MEANS	SAFETY LEVEL		
		A & B	A & B & C	A & B & C & D
All	Confidence-adding/ Directly-impacting			1, 2, 3

13.DA FOR HW COTS IP

ERROR TYPE	MEANS	SAFETY LEVEL		
		A & B	A & B & C	A & B & C & D
Transformation				1.d
Emerging/unintended		2.b		
Global	Directly-impacting/ Confidence-adding		2.d	1.a, 3

Local				
Interface				1.d
All	Directly-impacting			1.b, 1.c, 2.a, 2.c, 4

14.CONFIGURATION DATA ITEM

ERROR TYPE	MEANS	SAFETY LEVEL		
		A & B	A & B & C	A & B & C & D
Transformation	Directly-impacting			1.c, 2
Global	Directly-impacting		5	1.a, 1.b, 1.c, 3, 4, 8
Interface	Directly-impacting			1.a, 1.b, 1.c, 2, 3
All	Directly-impacting/ Confidence-adding			6, 7, 11
	Directly-impacting			9, 10

15.USER MODIFIABLE SOFTWARE

ERROR TYPE	MEANS	SAFETY LEVEL		
		A & B	A & B & C	A & B & C & D
Global	Directly-impacting			1, 2, 3
	Confidence-adding			3
Interface	Directly-impacting			1, 2, 3
	Confidence-adding			3

16.FEEDBACK

ERROR TYPE	MEANS	SAFETY LEVEL		
		A & B	A & B & C	A & B & C & D
Emerging/unintended	Directly-impacting			1.b
Global	Directly-impacting			1.a
Local	Directly-impacting			1.a
All	Directly-impacting			2

17.CONFIGURATION MANAGEMENT

ERROR TYPE	MEANS	SAFETY LEVEL		
		A & B	A & B & C	A & B & C & D
All	Directly-impacting			1, 2, 3, 4, 5

18.CHANGE MANAGEMENT

ERROR TYPE	MEANS	SAFETY LEVEL		
		A & B	A & B & C	A & B & C & D
Transformation	Directly-impacting			1.c
	Confidence-adding	1.d, including verify cases and procedures		1.d, excluding verify cases and procedures
Global	Directly-impacting			1.c
All	Confidence-adding			1.a, 1.b, 2*

* Applicability of this "check" evaluation item depends on the safety levels of the evaluation items the "check" process is confirming.

19.PROBLEM REPORTING

ERROR TYPE	MEANS	SAFETY LEVEL		
		A & B	A & B & C	A & B & C & D
All	Confidence-adding			1, 2, 3

20.PROCESS ASSURANCE

ERROR TYPE	MEANS	SAFETY LEVEL		
		A & B	A & B & C	A & B & C & D
Global	Confidence-adding		AEH 4.a	SW 4.a
Local	Confidence-adding		AEH 6	SW 6
Interface	Confidence-adding		AEH 4.b	SW 4.b
All	Directly-impacting/ Confidence-adding		AEH 5	SW 5
	Confidence-adding		AEH 1, AEH 2, AEH 3	SW 1, SW 2, SW 3

To reflect at which safety levels the evaluation item is applicable, each evaluation item is preceded by the safety levels within brackets: [A/B/C/D]. When an evaluation item is attributed a different set of safety levels for SW or AEH, there is another bracket to indicate the specific additional safety level and for which domain. The following examples illustrate how to interpret the safety level attribute:

- *[A/B/C]:* evaluation item is attributed to safety levels A, B and C for both SW & AEH domains.
- *[A/B/C][D(SW)]:* evaluation item is attributed to safety levels A, B and C for both SW & AEH domains, and to safety level D for SW.
- *[A/B/C][D(AEH)]:* evaluation item is attributed to safety levels A, B and C for both SW & AEH domains, and to safety level D for AEH.

While this methodology established the relevant safety level to expect for an evaluation item to be addressed by a process, this report makes no conclusions about the error typology or the effect of the two different categories of means associated with the evaluation item. However, this association of error typologies and category of means may serve as the foundation to create a framework for assessing the risk associated with a gap detected when assessing a standard against the Abstraction Layer.

APPENDIX V: Abstraction Layer User Guide

Introduction

The Abstraction Layer (AL) is a collection of 20 criteria and associated evaluation items. These criteria and evaluation items were created as an abstraction of the assurance objectives from ED-12C/DO-178C and ED-80/DO-254. They are intended to be used to assess other assurance standards for suitability for use in aviation.

This User Guide is intended to be a "how to" guide containing the TFAL's recommendations for performing future assessments. It is based on the experience gained by the TFAL during a trial application of the AL on a public alternate standard. It is not binding guidance in any way but represents consensus best practices.

1 Forming an Assessment Task Force (ATF)

1.1 Organization

The most effective assessment will be one performed by a team representing industry and certification authorities. This team can in theory be organized by any party, but it seems that industry trade associations (e.g., GAMA, ASD, AIA) would be best suited to coordinate an assessment. Alternatively, a standards body updating or creating a new standard (e.g., EUROCAE, RTCA, SAE, ASTM) might be well-positioned to include an AL assessment task as a part of their standards work. Regardless of the initiator, an ATF should include sufficient representation and interest from the aviation industry to confirm the perceived value in the alternate standard, since airworthiness authorities may choose to scale their level of involvement based on their perception of industry interest.

1.2 Composition

1.2.1 Technical Expertise

The ATF should include technical subject matter experts (SMEs) from all applicable disciplines the alternate standard encompasses, i.e., if the alternate standard covers both Software and AEH, the ATF should include Avionics industry SMEs in both the Software and AEH disciplines. These SMEs should have significant experience and expertise in applying DO-178C & A(M)C 20-115D and/or DO-254 & A(M)C 20-152A.

1.2.2 Industry and Authorities Participation

The ATF will benefit from inclusion of perspectives of both industry members and airworthiness authorities. The time commitment from both groups will indicate both the seriousness of intent and represent an investment that will provide motivation to complete the effort and see the alternate standard brought to acceptance by the aviation community.

1.2.3 Geographic Location

At the time of this writing, EASA and the FAA are the two airworthiness authorities committed to recognizing the AL as guidance material and using it to assess alternate standards. The TFAL effort was initiated by these two authorities. However, the TFAL effort also included participation from Transport Canada Civil Aviation (TCCA) and the National Civil Aviation Agency of Brazil (ANAC), since civil aviation projects frequently span and require involvement from those agencies as well. In general, participants should be invited into the ATF from all geographic regions where the resulting alternate standard is anticipated to be used. Early engagement in the ATF should aid in collectively learning the alternate standard and smooth the acceptance process later.

1.2.4 Alternate Domain Subject Matter Expert(s)

Subject Matter Experts in the alternate standard should be sought out as trusted advisors for the ATF. Such SMEs may have valuable insight into the intent of the standard's objectives, the current state of the art in using the standard, and what audits or evidence might typically be available to substantiate compliance to the standard. The TFAL does not recommend including these SMEs as full members of an ATF assessing a standard since the SMEs may be unduly biased toward acceptance of that standard. The SME input as an advisor, however, is essential to the success of the ATF.

2 Selecting a Standard

Alternate standards may come from any source but should be chosen based on the perceived value to the aviation industry. For example, a product manufactured by a supplier to an automotive standard for the automotive industry may be useful to the aviation industry and would provide even better value if the automotive standard could be recognized as providing an acceptable reference for a development assurance process.

A standard needs a certain level of support from the aviation industry in order to convince the aviation authorities to invest the time and resources needed to perform the assessment. The authorities have indicated that requests for assessment of an alternate standard need to come from aviation industry members – i.e., they do not expect to support requests from unrelated alternate industries or suppliers in the hope of some commercial gain in the aviation industry. Those unrelated industries will need to collaborate with aviation industry members to demonstrate the value of and demand for the alternate standard to aviation.

3 Performing an Assessment

3.1 Defining the Scope of the Assessment

Depending on the full scope of the alternate standard, it may be desirable to define a more limited scope to bound the assessment activity. For instance, the assessment of a standard that addresses both systems and software development assurance may be limited to software aspects. It may also be necessary to consider Development Assurance Levels (DALs). For example, the scope of the assessment may bound to specific DAL(s). Scoping the assessment properly will help the ATF perform an effective assessment.

3.2 Assessment Process

The TFAL test assessment of the alternate standard followed a three-stage process that proved effective. Future ATFs may choose to follow this model.

Stage 1: Familiarity
At this stage, the ATF is likely unfamiliar with the alternate standard. The first activity is to procure and read the document and to broadly identify sections of the document that seem applicable to the various evaluation items. SMEs in the alternate standard should be engaged to provide an overview of the document, its history and use. This will give the ATF enough insight into the general structure of the alternate standard to organize the assessment effort. An ongoing record of questions posed, and corresponding SME responses should be retained for reference.

Stage 2: Detailed Analysis

At this stage, the criteria and associated evaluation items are assigned to individuals or small sub-teams (2-4 members each) from the ATF. These individuals and sub-teams fill out the Assessment Template for their assigned criteria, using the applicable sections identified in Stage 1, following the guidance listed in section 4.3. This includes the collection of raw data from the alternate standard (Table 1) and a proposed draft of the final assessment for the criteria (Table 2). Use of a collaboration site is recommended to share the completed assessment templates. The ATF should then review the detailed assessment templates to ensure concurrence with the assessments. As questions come up during this stage, regular interaction with SMEs in the alternate standard can provide clarification and help avoid misinterpretation.

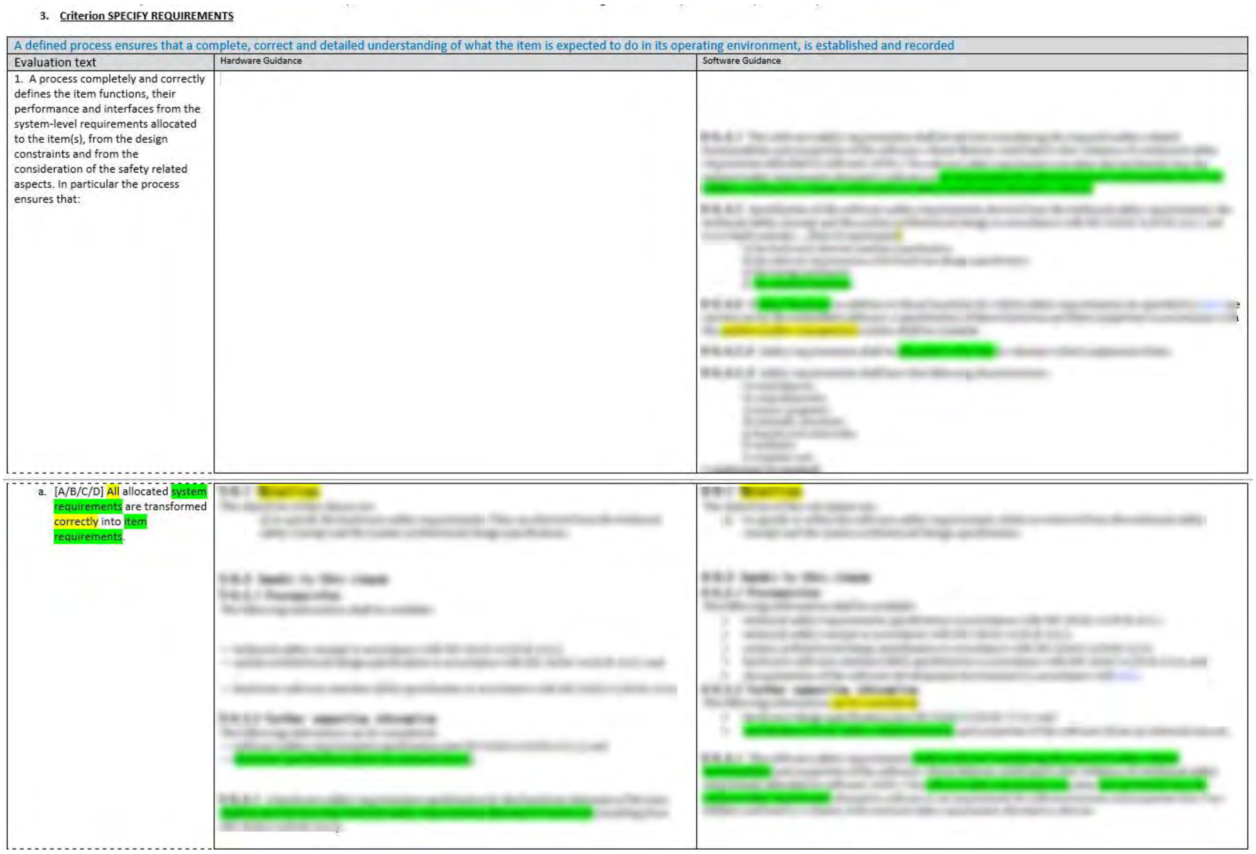
State 3: Final Assessment

At this stage, the full ATF meets (either in-person or via online meeting) and reviews each of the completed Assessment Templates. The goal of the ATF in this phase is to reach a consensus on each assessment result, and to provide clear explanations for each evaluation item. The final Assessment Templates should be archived for future reference.

3.3 Assessment Templates

The TFAL produced a set of Assessment Templates that are recommended for use in performing an assessment using the AL. These templates help facilitate a structured assessment that documents the assessment rationale along with a final, consistent assessment. The set of Assessment Templates is available as a separate file.

There is one assessment template for each of the 20 criteria. Each template includes two tables. The first table is used to capture any relevant guidance from the alternate standard that seems to address the assessment text.

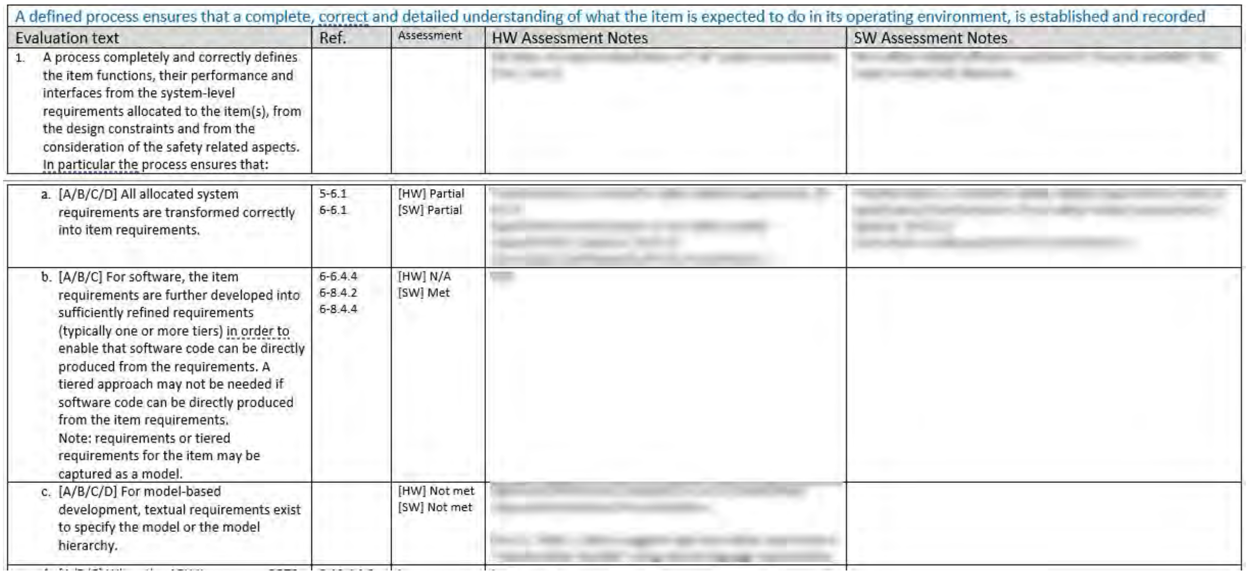


3. Criterion SPECIFY REQUIREMENTS

A defined process ensures that a complete, correct and detailed understanding of what the item is expected to do in its operating environment, is established and recorded		
Evaluation text	Hardware Guidance	Software Guidance
1. A process completely and correctly defines the item functions, their performance and interfaces from the system-level requirements allocated to the item(s), from the design constraints and from the consideration of the safety related aspects. In particular the process ensures that:		[illegible]
a. [A/B/C/D] All allocated system requirements are transformed correctly into item requirements.	[illegible]	[illegible]

In this table, insert or copy/paste reference text from the alternate standard into the appropriate cell to document the guidance. This table is not used to document the final assessment (because of copyright issues) – only to provide a broad overview of relevant material from the alternate standard that should be assessed. Color coding may be used to show the correspondence, or lack thereof, between the standard under assessment and the AL evaluation items.

The second table in the Assessment Template is for the final assessment.



A defined process ensures that a complete, correct and detailed understanding of what the item is expected to do in its operating environment, is established and recorded

Evaluation text	Ref.	Assessment	HW Assessment Notes	SW Assessment Notes
1. A process completely and correctly defines the item functions, their performance and interfaces from the system-level requirements allocated to the item(s), from the design constraints and from the consideration of the safety related aspects. In particular the process ensures that:			[illegible]	[illegible]
a. [A/B/C/D] All allocated system requirements are transformed correctly into item requirements.	5-6.1 6-6.1	[HW] Partial [SW] Partial	[illegible]	[illegible]
b. [A/B/C] For software, the item requirements are further developed into sufficiently refined requirements (typically one or more tiers) in order to enable that software code can be directly produced from the requirements. A tiered approach may not be needed if software code can be directly produced from the item requirements. Note: requirements or tiered requirements for the item may be captured as a model.	6-6.4.4 6-8.4.2 6-8.4.4	[HW] N/A [SW] Met	[illegible]	
c. [A/B/C/D] For model-based development, textual requirements exist to specify the model or the model hierarchy.		[HW] Not met [SW] Not met	[illegible]	

This table should include the following information:

Ref: Reference to the section of the alternate standard relevant to this assessment.

Assessment: The assessment may be separate for Hardware and Software, and should be one of the following values:

- MET (evaluation item is fully met by the alternate standard)
- NOT MET (evaluation item is not addressed at all by the alternate standard)
- PARTIAL (parts of the evaluation item are met by the alternate standard)

HW / SW Assessment Notes: Use these columns to provide notes on why the listed assessment was chosen. A MET assessment does not need notes; the reference to the alternate standard is sufficient. For NOT MET, use the assessment column to document what is missing. Particularly relevant is if there were portions of the alternate standard that seemed possibly applicable but were dismissed for one reason or another. For PARTIAL, use the assessment column to indicate which portions of the evaluation item text are met and which are missing. For both NOT MET and PARTIAL, try to provide sufficient rationale that future users of the assessment worksheets understand why the assessment was made to that status.

Note: greyed-out cells in the tables are not expected to have an entry, since they are heading-level items that are covered by the assessment of the sub-items that follow or are not applicable to either the Software or AEH discipline.

4 Proposals to Address Gaps

A completed assessment of an alternate standard is likely to identify gaps where the “Assessment” result is PARTIAL or NOT MET. In those cases, measures will need to be identified to close the gaps to allow recognition of the alternate standard. While it is not required to cover the gaps prior to submitting the assessment for authority recognition, proposed development assurance activities to close a gap may be included within the Assessment Templates or in a final report detailing the results of the assessment.

5 Submitting the Assessment

The final assessment of the alternate standard, including the completed Assessment Templates and a final summary report, should be provided to the airworthiness authorities who participated in the ATF or other authority leadership that they identify. This final assessment may include suggestions related to applicability of discipline (e.g., hardware vs. software) or DAL. The assessment will be provided together with a statement of interest (or not) to retain the standard for aviation.