Tool Qualification Symposium

Principles and benefits of using DO-330/ED-215 (*Software Tool Qualification Considerations*)

Frédéric POTHON - ACG Solutions
Frederic.pothon@acg-solutions.fr
Tel: (33)4. 67. 609.487
www.acg-solutions.fr

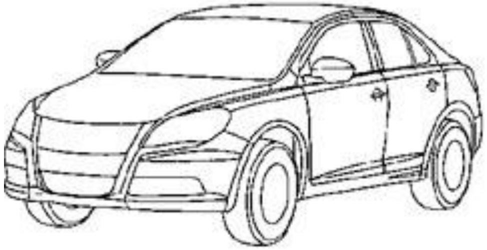
First Tool Qualification Symposium:
April 9th-10th, 2013 in Munich

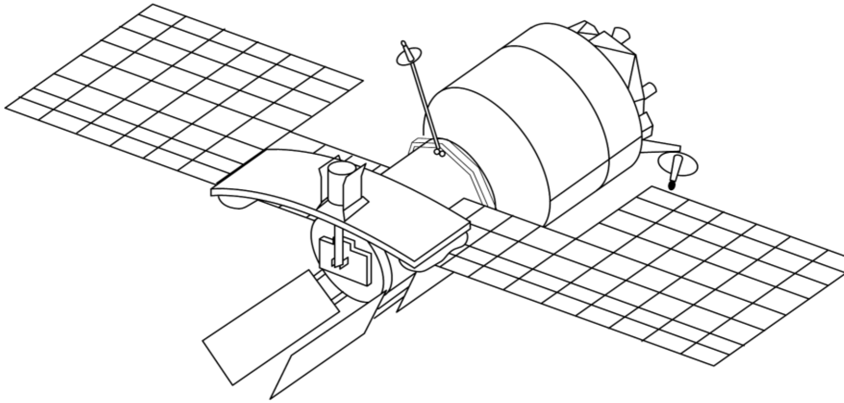
Principles and Benefits of using DO-330/ED-215: Agenda

1. Tool "Qualification", what does it mean?
2. A dream: A domain independent document
3. Tool Qualification "Levels"
4. Why qualifying a code generator?

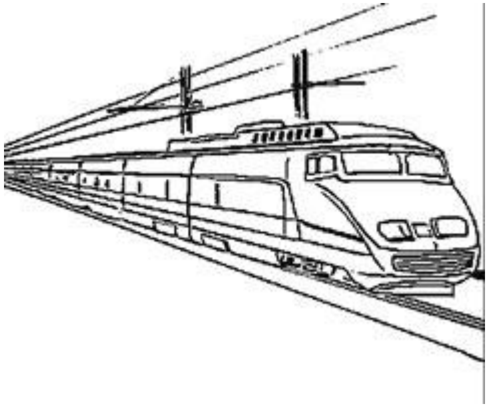
1. “Qualification”, what does it mean?

Tool qualification definition:





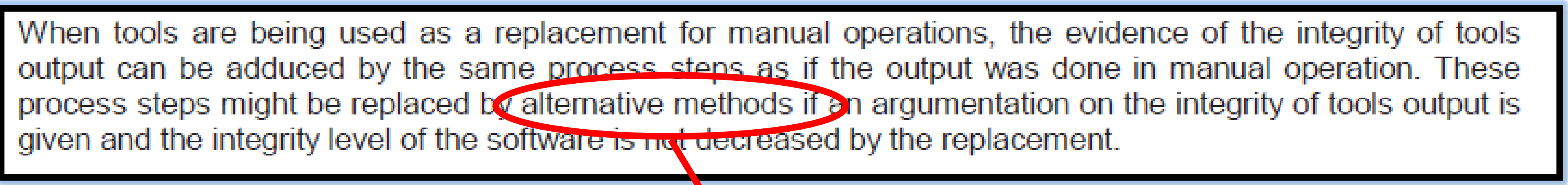
The term is widely used, but are we talking about the same thing ?



1. “Qualification”, what does it mean?

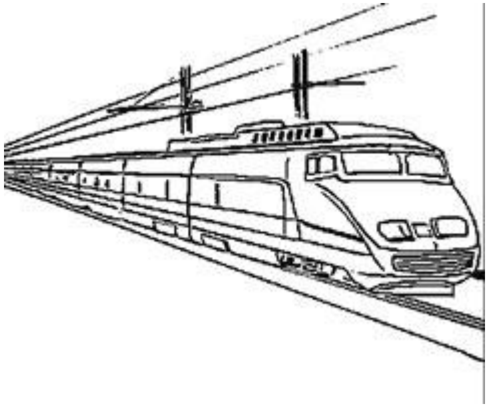
Tool qualification definition:

“Qualification” not directly used
3 tool classes: T1 (no impact), T2(Verification), T3 (Compiler, code generator)



When tools are being used as a replacement for manual operations, the evidence of the integrity of tools output can be adduced by the same process steps as if the output was done in manual operation. These process steps might be replaced by alternative methods if an argumentation on the integrity of tools output is given and the integrity level of the software is not decreased by the replacement.

EN 50128:2011



Term “Tool validation” used
Few guidance provided.
Implicitly certification credit: No tool outputs verification

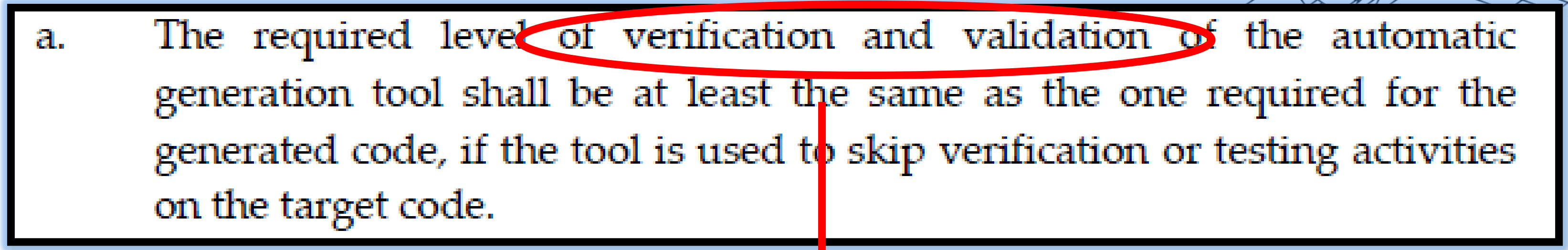
1. “Qualification”, what does it mean?

Tool qualification definition:

ECSS-Q-ST-80C

“Qualification” not directly used

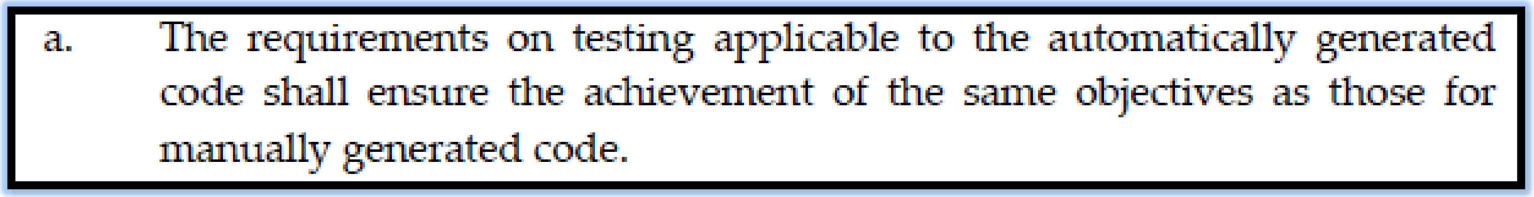
Focuses on Automatic Code Generator



a. The required level of verification and validation of the automatic generation tool shall be at least the same as the one required for the generated code, if the tool is used to skip verification or testing activities on the target code.

“verification” and “validation” of the tool

Certification credit unclear



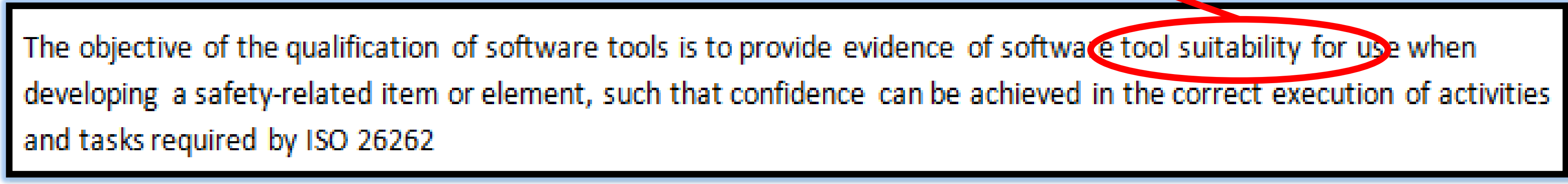
a. The requirements on testing applicable to the automatically generated code shall ensure the achievement of the same objectives as those for manually generated code.

1. “Qualification”, what does it mean?

Tool qualification definition:

ISO26262

3 Tool Confidence Level TCL3 (High impact), TCL2(Verification), TCL1 (No impact) “Qualification” of software tools defined



The objective of the qualification of software tools is to provide evidence of software tool suitability for use when developing a safety-related item or element, such that confidence can be achieved in the correct execution of activities and tasks required by ISO 26262

Certification credit unclear

1. “Qualification”, what does it mean?

Tool qualification definition:

ISO2626

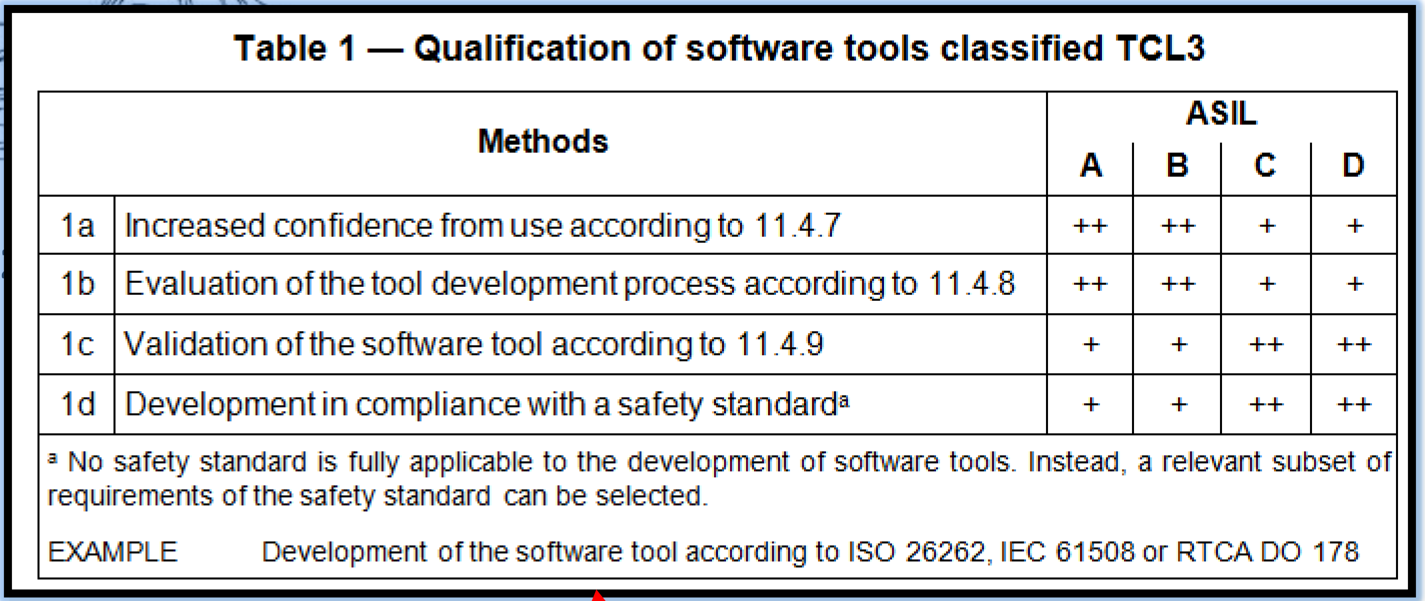


Table 1 — Qualification of software tools classified TCL3

	Methods	ASIL			
		A	B	C	D
1a	Increased confidence from use according to 11.4.7	++	++	+	+
1b	Evaluation of the tool development process according to 11.4.8	++	++	+	+
1c	Validation of the software tool according to 11.4.9	+	+	++	++
1d	Development in compliance with a safety standard[a]	+	+	++	++

[a] No safety standard is fully applicable to the development of software tools. Instead, a relevant subset of requirements of the safety standard can be selected.

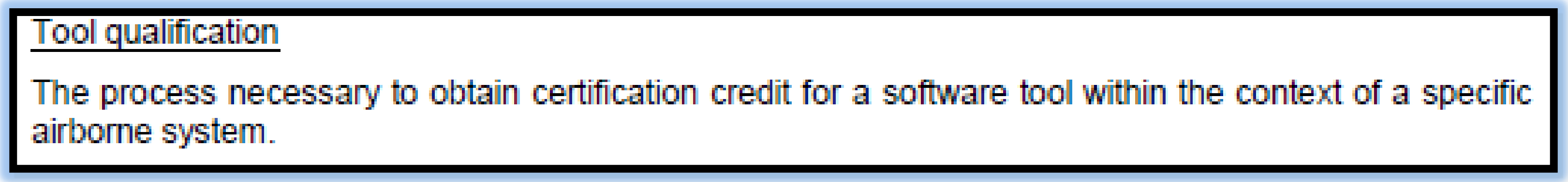
EXAMPLE Development of the software tool according to ISO 26262, IEC 61508 or RTCA DO 178

Several approaches allowed, including the use of another standard

1. “Qualification”, what does it mean?

Tool qualification definition:

The qualification approach is based on a software engineering approach, so it is applied only to the “software tools” .



Tool qualification

The process necessary to obtain certification credit for a software tool within the context of a specific airborne system.

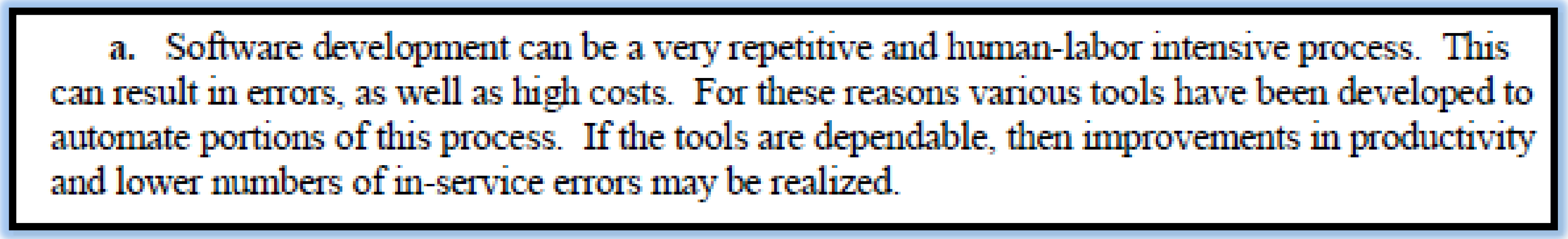
Certification credit ⇔ Confidence in the tool
It will be obtained in regard of the tool qualification process!

Qualification is achieved in a **<u>context for a specific usage,</u>** including the operational environment, the inputs definition, the options used… and the certification credit!

1. “Qualification”, what does it mean?

Tools in the software life cycle

- FAA N8110_91



a. Software development can be a very repetitive and human-labor intensive process. This can result in errors, as well as high costs. For these reasons various tools have been developed to automate portions of this process. If the tools are dependable, then improvements in productivity and lower numbers of in-service errors may be realized.

1. “Qualification”, what does it mean?

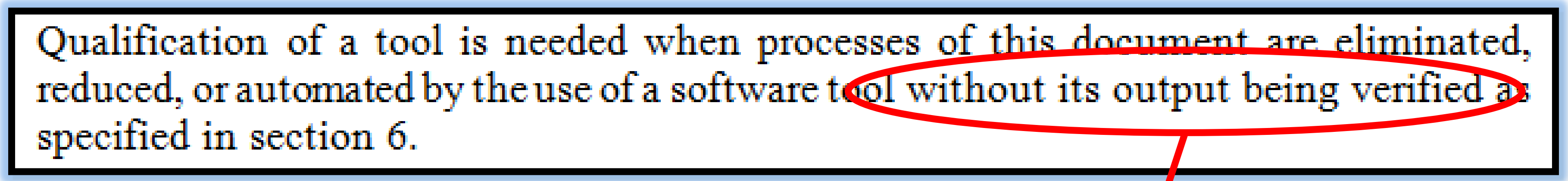
Tools in the software life cycle

- May we use a tool to replace any manual activity?
- Do we need to qualify it?
- Is there another benefit to qualify a tool?

1. “Qualification”, what does it mean?

Tools in the software life cycle

DO-178/ED-12:



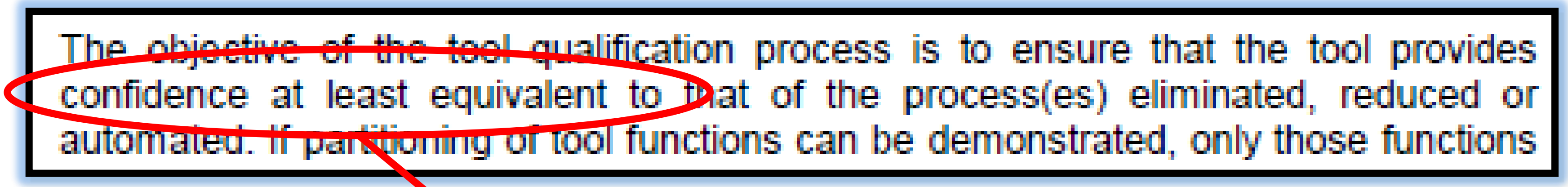
Qualification of a tool is needed when processes of this document are eliminated, reduced, or automated by the use of a software tool without its output being verified as specified in section 6.

When the tool outputs are verified, and all verification objectives satisfied, then the qualification is not required.

1. “Qualification”, what does it mean?

Tools in the software life cycle

DO-178/ED-12:



The objective of the tool qualification process is to ensure that the tool provides confidence at least equivalent to that of the process(es) eliminated, reduced or automated. If partitioning of tool functions can be demonstrated, only those functions

Confidence ⬄ Certification credit claimed
It will be obtained in regard of the tool qualification process!

1. “Qualification”, what does it mean?

Tools in the software life cycle

The qualification replaces a recurrent activity performed in the scope of the Software life cycle, with a non-recurrent activity performed at tool level.

2. A domain independent document

DO-330/ED-215 Document

RTCA/EUROCAE working group

Guidance for airborne and ground software

updated

Two main packaging decisions:

- Supplements for to document technology-specific or method-specific guidance
- A new document for tool qualification

2. A domain independent document

DO-330/ED-215 Document

- To consider that tools usage benefits to safety
- To encourage the use of tools by providing relevant qualification objectives
- To improve the tool quality by addressing their specificities
- To keep the objective-based approach
- To not prevent the use of technical and methods that best fit to tools
- To produce an easy to use document
- To produce a document that may be apply to other domains

2. A domain independent document

DO-330/ED-215 Document

DO-178C/ED-12C
and also
DO-278/ED-109
DO-254/
DO-200/
And why not other domains as
Space (ECSS)
Automotive (ISO26262) …

Need for qualification
Level of qualification
Reference the TQS

The European Organisation for Civil Aviation Equipment
L'Organisation Européenne pour l'Equipement de l'Aviation Civile

SOFTWARE TOOL QUALIFICATION CONSIDERATIONS

This document is the exclusive intellectual and commercial property of EUROCAE.
It is presently commercialised by EUROCAE.
This electronic copy is delivered to your company/organisation for internal use exclusively.
In no case it may be re-sold, or hired, lent or exchanged outside your company.

ED-215
January 2012

102 rue Etienne Dolet
92240 Malakoff, France
Web Site: www.eurocae.net

Tel: 33 1 40 92 79 30
Fax: 33 1 46 55 62 65
Email: eurocae@eurocae.net

DO-330/ED-215:
Tool Qualification *Document*
"Software Tools Qualification Considerations"

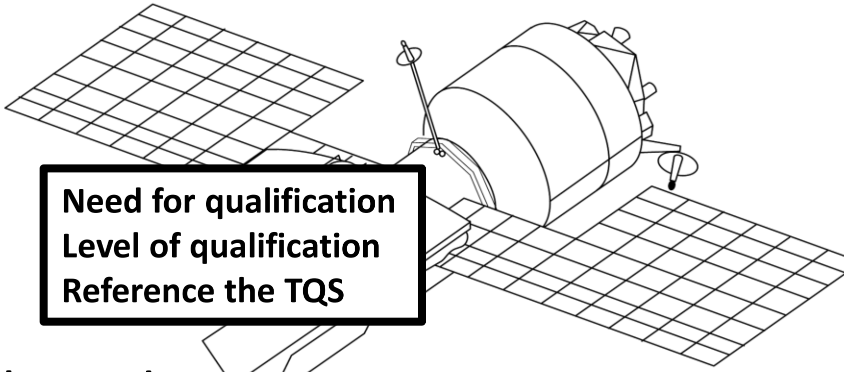
Domain Independant

"How" to qualify the tools
Objective-oriented
Tool processes
Leveling "TQL"
Clarifications

2. A domain independent document

Benefits:

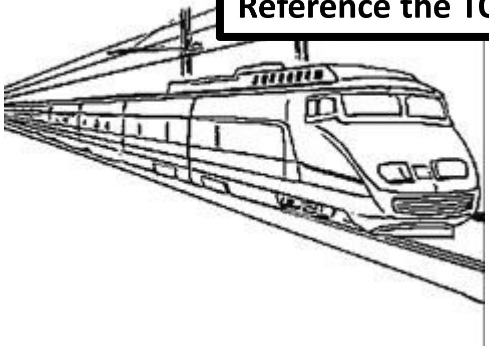
Need for qualification
Level of qualification
Reference the TQS



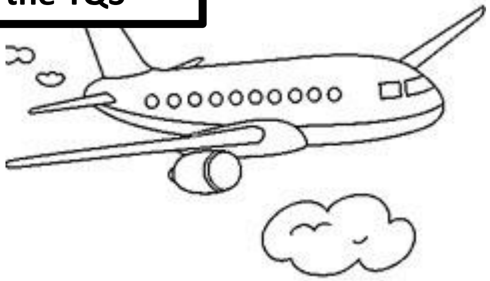
Need for qualification
Level of qualification
Reference the TQS

Each domain defines, based on its own constraints, the need and level of tool qualification

Need for qualification
Level of qualification
Reference the TQS

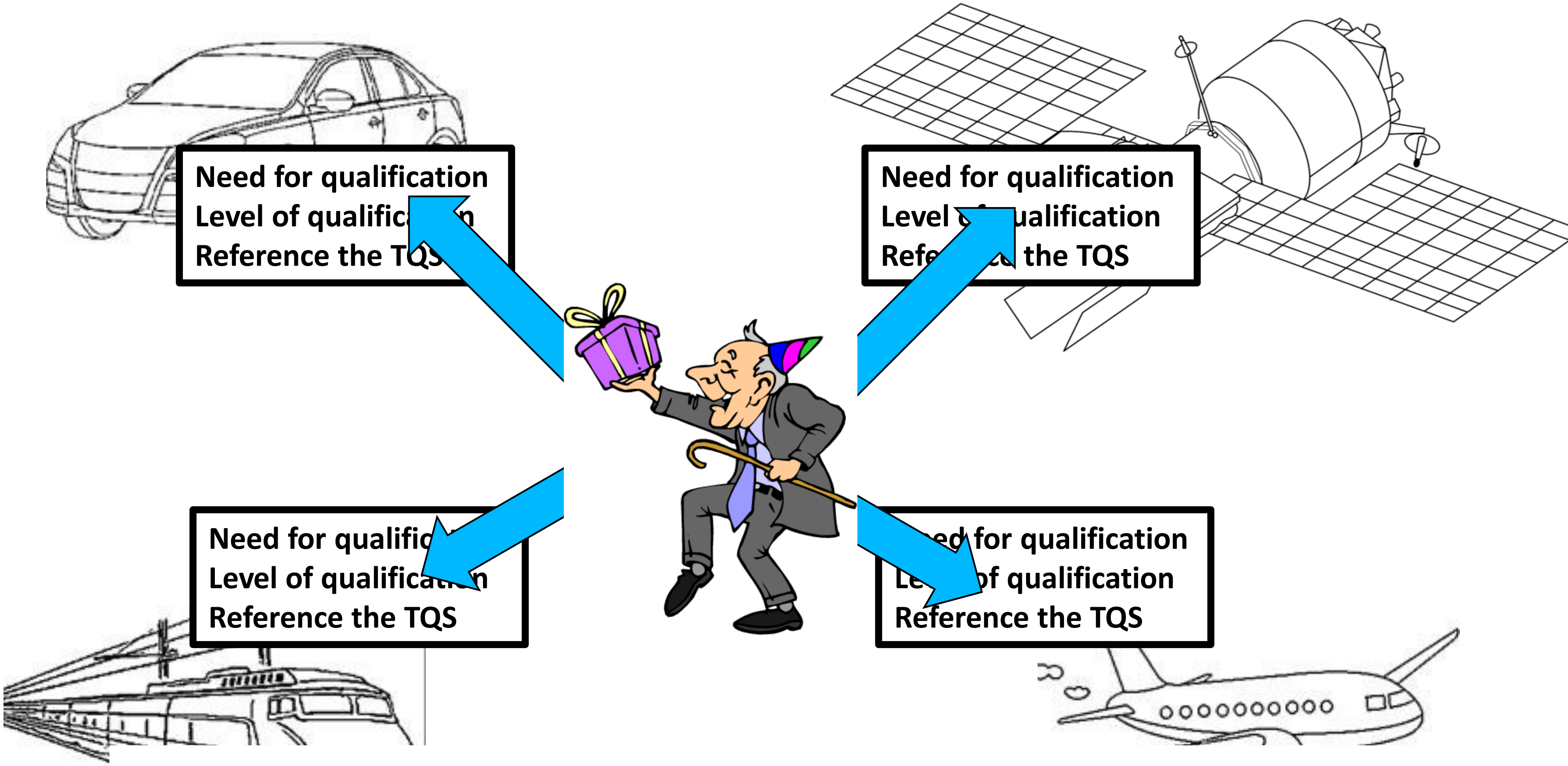


Need for qualification
Level of qualification
Reference the TQS



2. A domain independent document

Benefits:



Need for qualification
Level of qualification
Reference the TQS

Need for qualification
Level of qualification
Reference the TQS

Need for qualification
Level of qualification
Reference the TQS

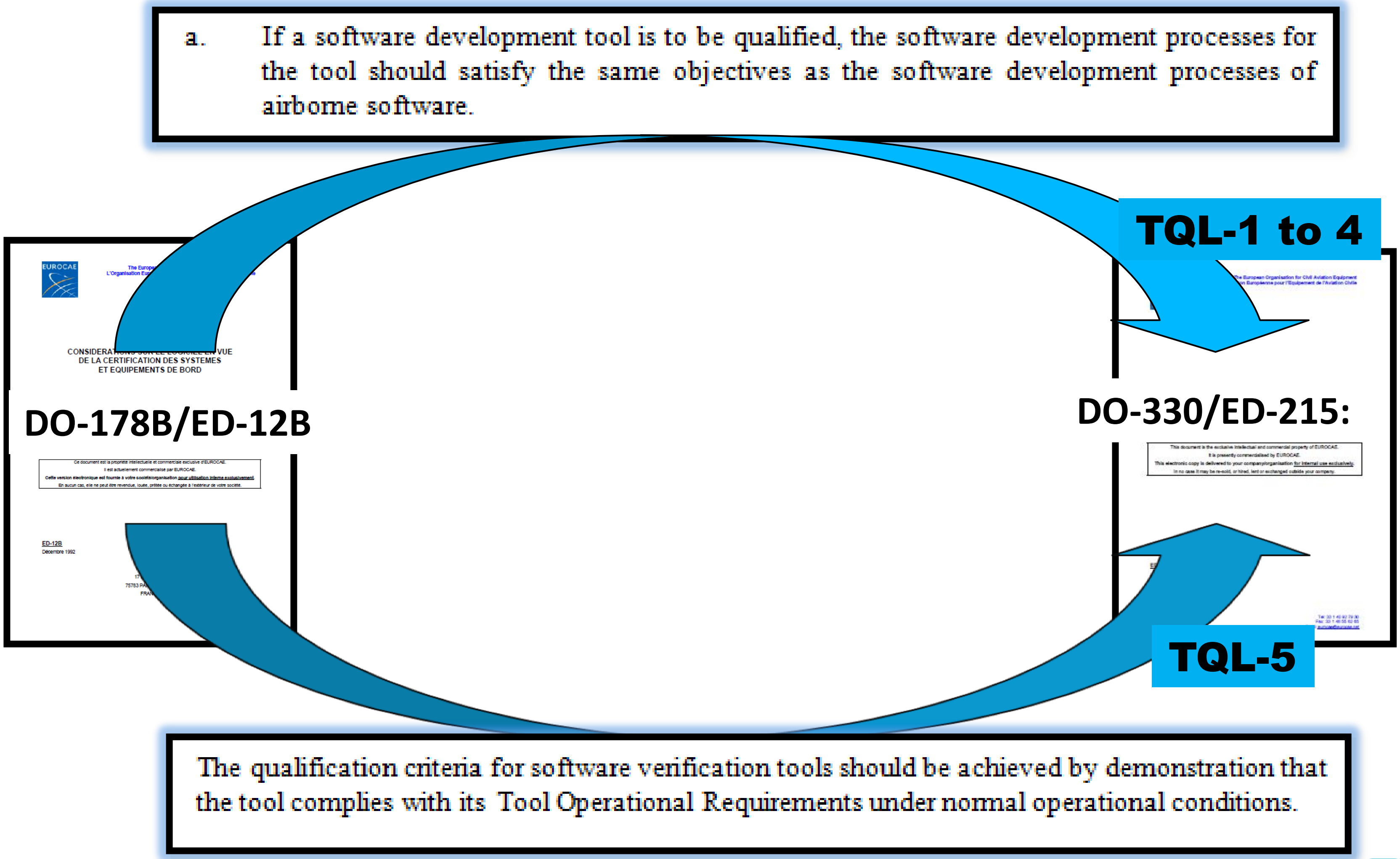
Need for qualification
Level of qualification
Reference the TQS

A tool vendor have a unique reference for qualification

As tool qualification is encouraged, will increase the tool offer and their quality

3. Tool Qualification Levels

5 levels from TQL-1 (most rigorous) to TQL-5 (less)



a. If a software development tool is to be qualified, the software development processes for the tool should satisfy the same objectives as the software development processes of airbome software.

The qualification criteria for software verification tools should be achieved by demonstration that the tool complies with its Tool Operational Requirements under nomal operational conditions.

3. Tool Qualification Levels

Principle: Tool User" and "Tool Developer"

Qualification performed in the scope of a specific project/context

Some errors may be detected only in the user context

=> Specific Objectives for the user context

=> Tool user is responsible of the qualification

This separation facilitates reuse, COTS and further qualification after changes

TQL-5 qualification requires only user activities

3. Tool Qualification Levels

Principle: “Tool User” and “Tool Developer”

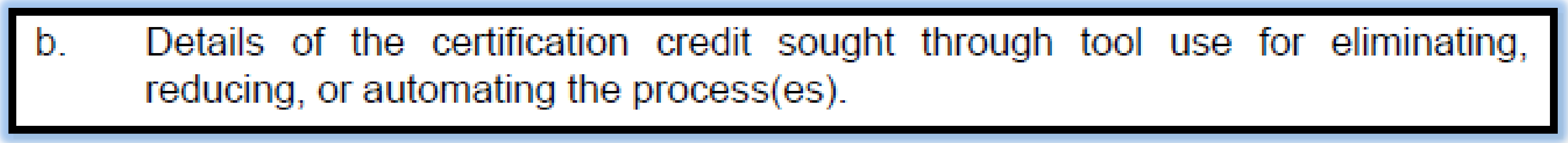
Two levels of requirements;

- **Tool Operational Requirements** describe the software life cycle needs (user context).
- Tool development processes produce **Tool Requirements** (one or several levels), from TOR (developer context)

3. Tool Qualification Levels

Principle: Adequacy of the tool?

Identification of the certification credit in a software document (not tool)



b. Details of the certification credit sought through tool use for eliminating, reducing, or automating the process(es).

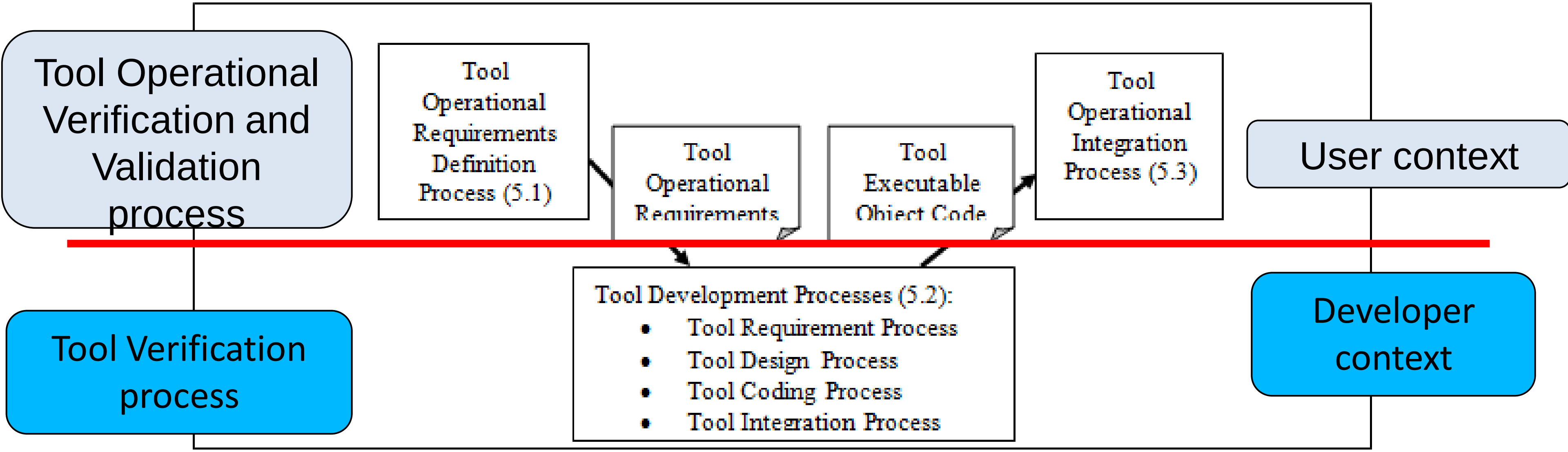
Not enough to verify the compliance to the requirements

"Validation" is necessary!

To ensure that the tool is compliant with the user needs (software life cycle) as described in the TOR …. or NOT!

3. Tool Qualification Levels

Principle: Tool User and Tool Developer processes



Tool Operational Verification and Validation process

Tool Operational Requirements Definition Process (5.1)

Tool Operational Requirements

Tool Executable Object Code

Tool Operational Integration Process (5.3)

User context

Tool Verification process

Tool Development Processes (5.2):

- Tool Requirement Process
- Tool Design Process
- Tool Coding Process
- Tool Integration Process

Developer context

3. Tool Qualification Levels

TQL-5 (⇔ *Verification tools*)

Scope:

Tools that cannot introduce errors, but may fail to detect them, and whose outputs are not verified (« for correctness »)

Certification credit: The « diagnostic » provided by the tool is correct

This diagnostic may be:

- Error detection or no errors
- Measurements
- Other life cycle data (whose cannot insert an error e.g. test data)

3. Tool Qualification Levels

TQL-5 (⇔ *Verification tools*)

Qualification: Compliance with "user needs"

- Tool Operational Requirements development and validation
- Requirements based tests in the operational environment
- To apply "integral processes" on these activities:
  - Quality Assurance
  - Configuration management
  - Change management: Limited to impact analysis of known problems

3. Tool Qualification Levels

TQL-5 (⇔ *Verification tools*)

Focus points

- Do not need to consider all the functions/features of the tool but only those used.
- The qualification is done from a user point of view. He writes the TOR and performs the verification in the operational context
- The TOR should be focused on the expected behavior of the tool (and not how it works)

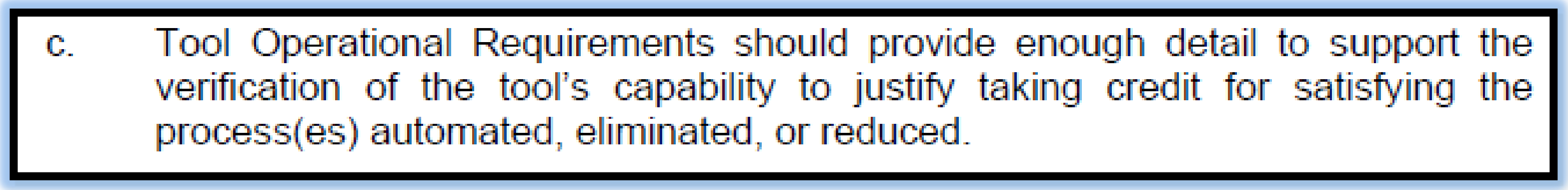
3. Tool Qualification Levels

TQL-5 (⬄ *Verification tools*)

Focus points

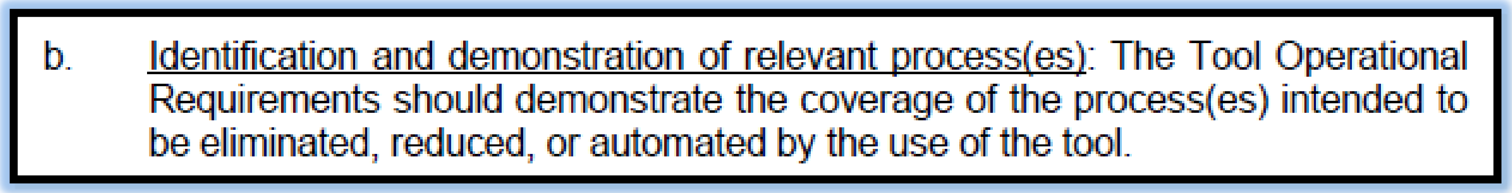
- TOR and Certification Credit

Guidance for development:



c. Tool Operational Requirements should provide enough detail to support the verification of the tool's capability to justify taking credit for satisfying the process(es) automated, eliminated, or reduced.

TOR validation:



b. Identification and demonstration of relevant process(es): The Tool Operational Requirements should demonstrate the coverage of the process(es) intended to be eliminated, reduced, or automated by the use of the tool.

3. Tool Qualification Levels

TQL-4 (⇔ "Super" *Verification tools*)

Example 1: Proof tool

The tool is used to automate some verification of Source Code.

=> TQL-5

In addition tests are alleviated (based on the confidence on the tool)

=> TQL-4

Example 2: Static code analyzer

The tool is used to automate some verification of Source Code review

=> TQL-5

Specific mechanisms for overflow are not introduced (based on the confidence on the tool)

=> TQL-4

3. Tool Qualification Levels

TQL-4 (⬄ “Super” *Verification tools*)

TQL4 versus TQL5:

More certification credit => More evidences to be provided

- TOR Verification
- Processes definition (plans)
- Tool requirements development and verification
- Tool Architecture description
- Tool compliance to « Tool Requirements » and requirement coverage analysis
- Change management

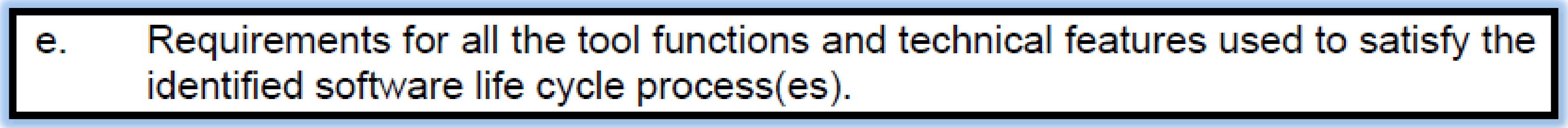
Similar as level D software
With TOR ⬄ System requirements

3. Tool Qualification Levels

TQL-4 (⬄ “Super” *Verification tools*)

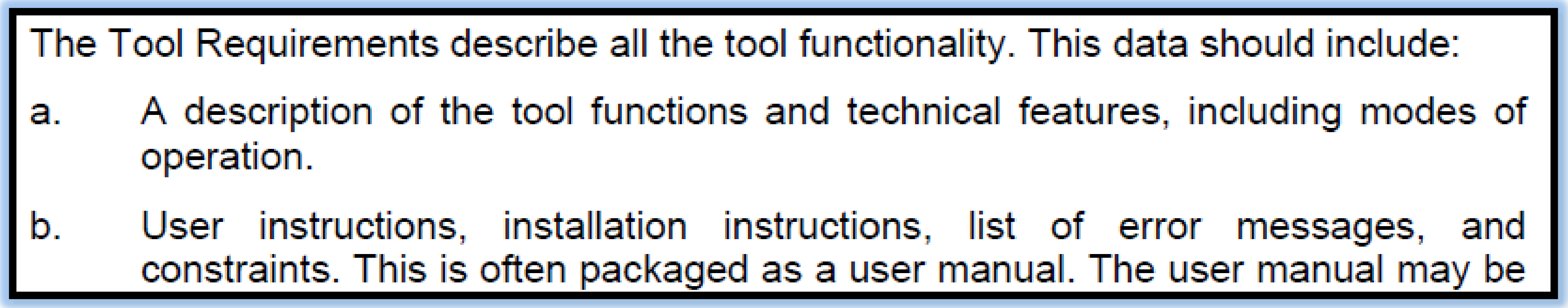
Complete description of the tool:

- TOR: may be limited the required functions



e. Requirements for all the tool functions and technical features used to satisfy the identified software life cycle process(es).

- Tool Requirements: All tool functions



The Tool Requirements describe all the tool functionality. This data should include:

a. A description of the tool functions and technical features, including modes of operation.

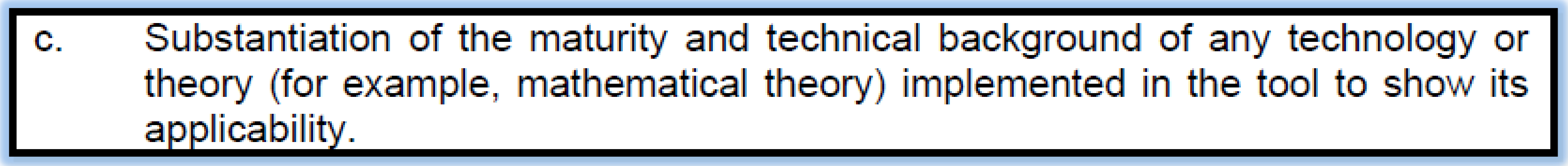
b. User instructions, installation instructions, list of error messages, and constraints. This is often packaged as a user manual. The user manual may be

3. Tool Qualification Levels

TQL-4 (⇔ “Super” *Verification tools*)

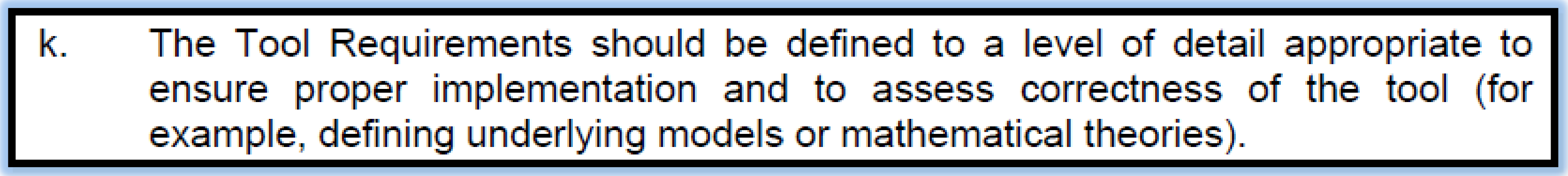
Initial intent: Formal Method

PSAC



c. Substantiation of the maturity and technical background of any technology or theory (for example, mathematical theory) implemented in the tool to show its applicability.

Tool Requirements:



k. The Tool Requirements should be defined to a level of detail appropriate to ensure proper implementation and to assess correctness of the tool (for example, defining underlying models or mathematical theories).

3. Tool Qualification Levels

TQL-1, 2 and 3 (⇔ *Code Generators*)

Scope:

Tools whose output is part of the airborne software and thus can introduce errors

Certification credit:

- Development process automated
- Tool Outputs verification eliminated or reduced

3. Tool Qualification Levels

TQL-1, 2 and 3 (⬄ *Code Generators*)

Qualification: Similar as the software itself!

- Qualification at TQL-1 to 3 requires evidences about all the development, verification, quality assurance and management processes of the tool
- Need to be aware and to apply qualification requirements since the launch of the development: Process definition and data recording.
- and often very far from those applied by tool vendors, so very difficult to qualify a COTS as a development tool

3. Tool Qualification Levels

COTS Tools

Scope: Commercial (component) Off The Shelves

Commercial, freeware, open-source

- Tools “not developed” specifically for a project
- A supplier is identified
- Data from supplier may be not available
- Tool not developed of a project needs (⇔TOR)

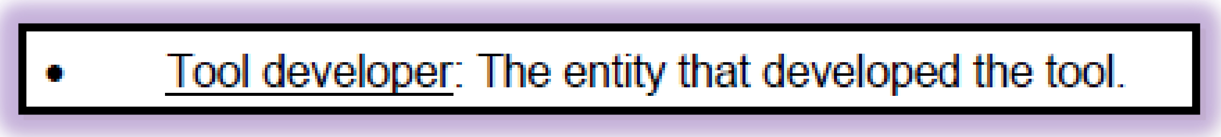
Certification credit: Same! Independent of tool origin!

3. Tool Qualification Levels

COTS Tools

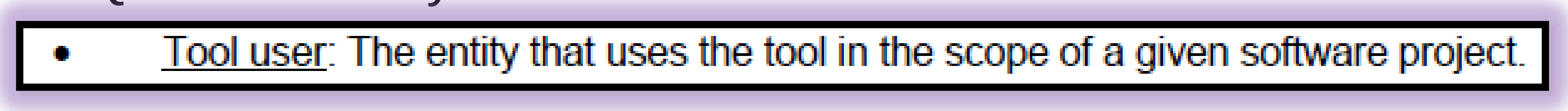
Qualification: A two steps approach (DO-330/ED-215§11.3)

- Pre-qualification by the Tool-Developer



- Tool developer: The entity that developed the tool.

- Qualification by the Tool User



- Tool user: The entity that uses the tool in the scope of a given software project.

3. Tool Qualification Levels

COTS Tools

Qualification: A two steps approach (DO-330/ED-215§11.3)

- Pre-qualification by the Tool-Developer
  - Pre-TQP, pre-TAS, pre-TCI
  - Pre-TOR
  - Development and verification performed accordingly to this pre-TOR.

3. Tool Qualification Levels

COTS Tools

Qualification: A two steps approach (DO-330/ED-215§11.3)

- Qualification by the Tool User
  - Finalization of TQP, TAS and TCI with user activities
  - Assess the pre-TOR and provide additional information
  - Assess qualification data and may provide additional data in case of deficiencies
  - Perform all « validation » activities

3. Tool Qualification Levels

COTS Tools

TQL-5 COTS tools:

May be qualified without any data from the tool vendor

Except known errors

4. AutoCode Generator qualification benefits

ACG Qualification: For which benefit?

- Are source code verification objectives achieved ?
- May tests be reduced ?
- What about structural coverage analysis ?

Not clearly defined in any standard

4. AutoCode Generator qualification benefits

After a lot of discussion, a FAQ was accepted (FAQ D.8) in the scope of DO-330/ED-215

- Identifies possible certification credit
- Use different scenario as possible software life cycle
- Does not pretend to be exhaustive
- Still subject to discussion

4. AutoCode Generator qualification benefits

Source Code verification?

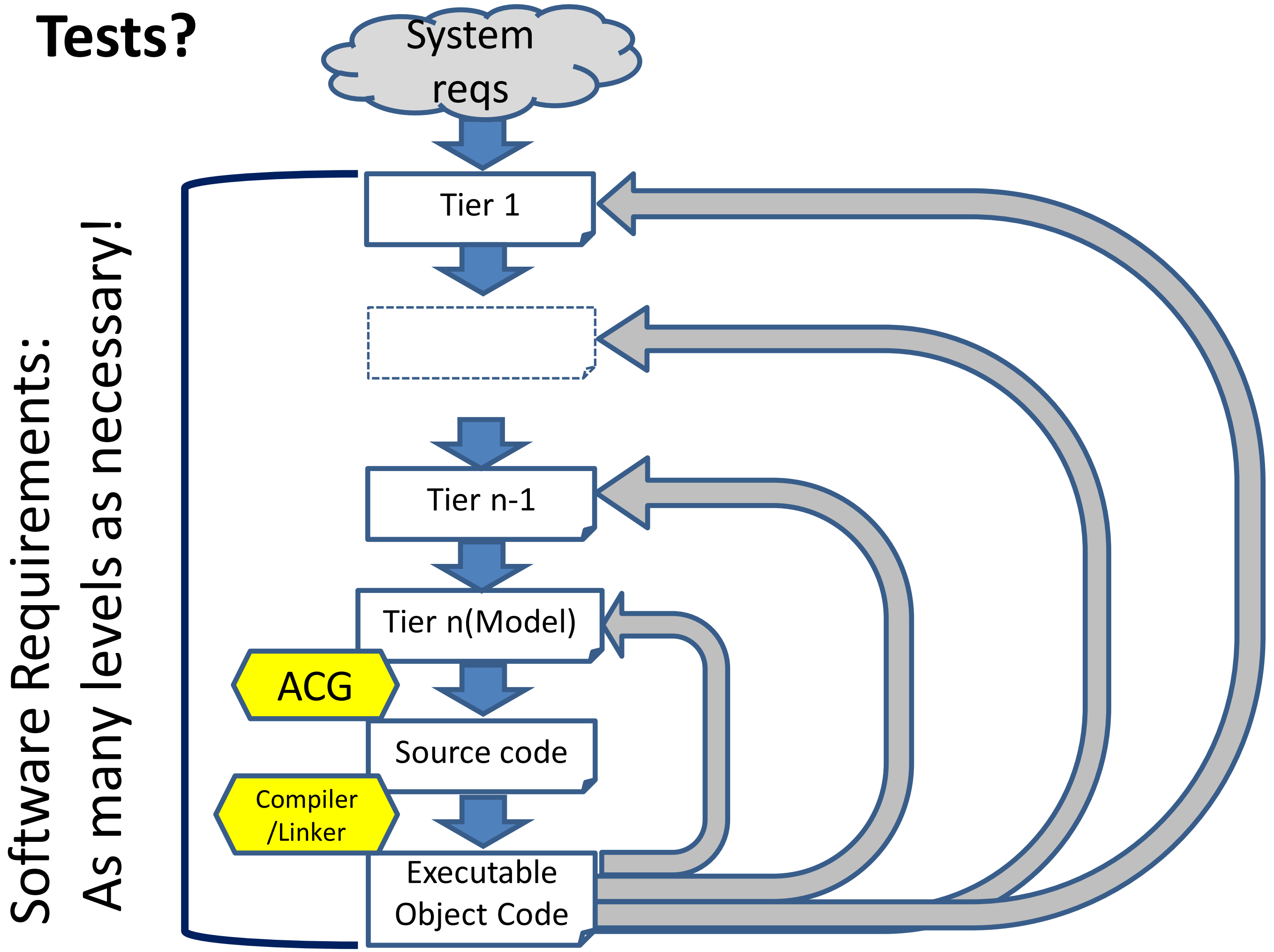
Source code is tool outputs and does need further verification

Based on
- the content and the accuracy of the Tool Requirements
- the completeness of tool verification process

But also additional verification should be performed
- Tool usage (parameters, environment ..)
- Generation of source code
- Impact of possible tool limitations and constraints
- Completeness and correctness of tool inputs

4. AutoCode Generator qualification benefits

Tests?

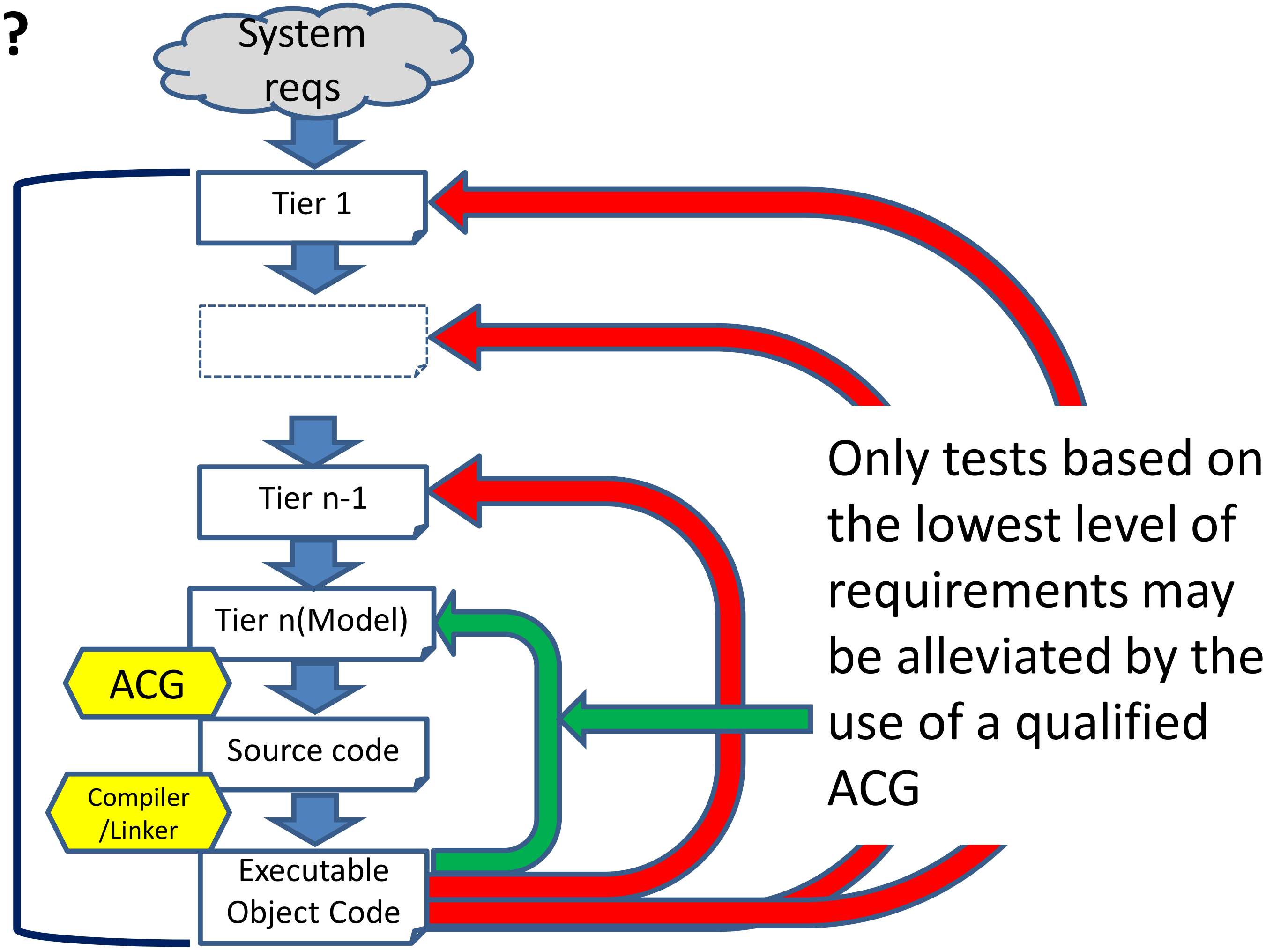


Tests: To demonstrate the compliance of EOC to all requirements (and all levels)

4. AutoCode Generator qualification benefits

Tests?

Software Requirements:
As many levels as necessary!



Only tests based on the lowest level of requirements may be alleviated by the use of a qualified ACG

4. AutoCode Generator qualification benefits

Tests?

Based on the thoroughness of the Tool Operational Verification and Validation process:

Equivalent to low-level requirements based tests, performed through equivalent classes of inputs

But, equivalent only if ...

Inputs: (1) all the allowed elements,

(2) an acceptable degree of combination

(3) the size and complexity limits.

Outputs: All possible statements that may be generated

EOC generation: Same compiler/linker, same setup.

4. AutoCode Generator qualification benefits

Tests?

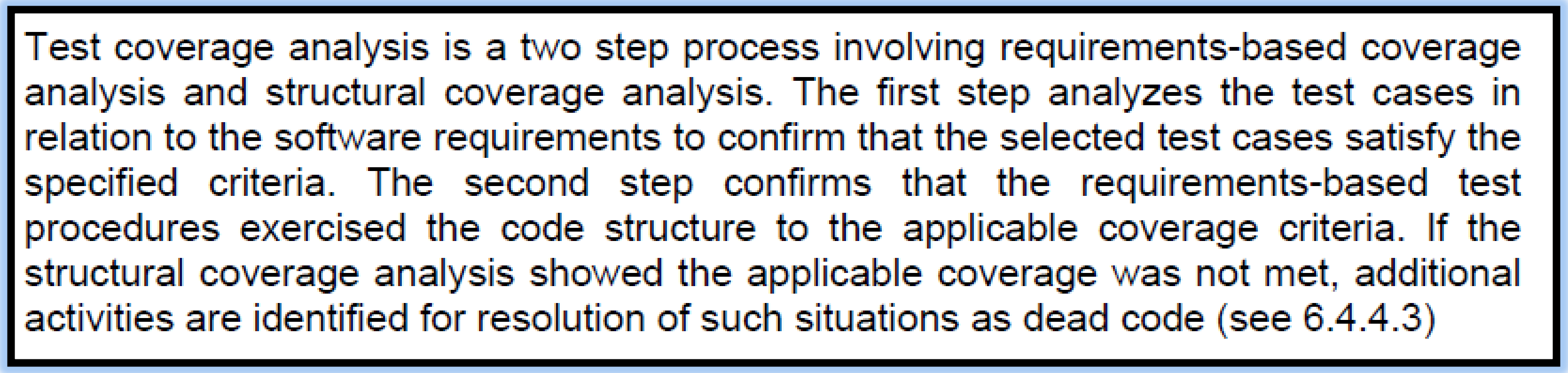
Activities

- To define a set of inputs that include all of the allowed elements, and a combination of that may be generated
- To execute the ACG on the set of inputs and generate source code
- To generate executable object code by using the compiler/linker with the selected options,
- To verify the compliance of the executable object code to the input files, through testing.

4. AutoCode Generator qualification benefits

Structural Coverage Analysis?

The purpose of the SCA is to verify the tests, not to verify the code, or the requirements …

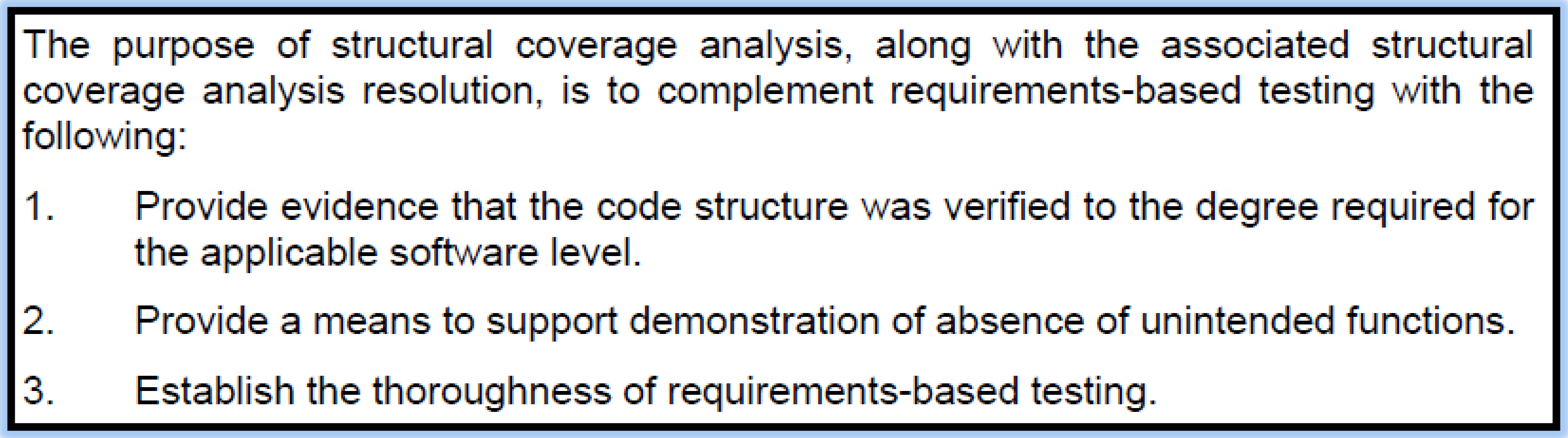


Test coverage analysis is a two step process involving requirements-based coverage analysis and structural coverage analysis. The first step analyzes the test cases in relation to the software requirements to confirm that the selected test cases satisfy the specified criteria. The second step confirms that the requirements-based test procedures exercised the code structure to the applicable coverage criteria. If the structural coverage analysis showed the applicable coverage was not met, additional activities are identified for resolution of such situations as dead code (see 6.4.4.3)

4. AutoCode Generator qualification benefits

Structural Coverage Analysis?

The purpose of the SCA is to verify the tests, not to verify the code, or the requirements …



The purpose of structural coverage analysis, along with the associated structural coverage analysis resolution, is to complement requirements-based testing with the following:

1. Provide evidence that the code structure was verified to the degree required for the applicable software level.
2. Provide a means to support demonstration of absence of unintended functions.
3. Establish the thoroughness of requirements-based testing.

4. AutoCode Generator qualification benefits

Structural Coverage Analysis?

Yes, may be claimed

Based on

- the content and the accuracy of the Tool Requirements (No unintended functions)
- the thoroughness of the Tool Operational Verification and Validation process

The representative set of inputs used in this activity is:

- Representative of the project requirements
- Sufficient to satisfy the requirements coverage objectives

Achievements

N° 1 : To address all kind of tools ☑

N° 2 : To improve the DO178 applicability to tools ☑

N° 3 : To make more feasible the “development tool” qualification ☒

N° 4 : To remain unchanged the “verification tool” qualification ☑

N° 5 : To address the tool particularities ☑

N° 6 : To ease the tool qualification data reuse ☑

N° 7 : To clarify the approach for COTS tool qualification ☑

N° 8 : To understand the tool qualification objectives without good knowledge of airborne software ☑

N° 9 : To be usable for other domains ☑

N° 10 : To fit with new software technologies ☑