DO-178B and COTS Tool Qualification

Version 1.2
2012-08-09
Application Note AN-ION-1-6102

Author(s) Klüser, Jürgen
Restrictions Public Document
Abstract This documents discusses requirements and aspects for the qualification of tools used in the development and testing process of aerospace electronics.

Table of Contents

1.0 Overview

In the development of electronic, software embedded systems or components, various tools are applied to efficiently perform tasks in design, implementation, testing, and documentation. Yet, in the development of safety critical applications in areas as aviation, automotive, railway transportation and others the use of tools generally is supposed to cause faults that may result in malicious behavior of the application. A set of standards has been developed to guide the assessment of tools and tool usage in the development processes of safety critical applications.

In the aviation industry, the DO-178B is a central standard for the development of software. Additional documents like the FAA guideline N8110.91 need to be considered as well. Generally the actual set of regulations and documents relevant in a development program is given as part of the "list of applicable documents" in the introduction of each development specification. In this document, we cover just some examples for basic understanding.

This application note discusses certain aspects of the use of engineering tools in developing and testing distributed electronic network systems in the aviation field.

2.0 Is MS Excel® certified?

A pen and a piece of paper can be considered as tools. This is also true for widely used software tools such as Microsoft® Excel® and more specialized engineering tools such as compilers and data bus test tools like CANoe.

This document addresses the question of certification or qualification of tools and its meaning in the frame of industrial applications.

DO-178B, as an important document for software requirements, uses the term “qualification” instead of “certification”. Therefore, in the following the term “qualification” is used.

3.0 Requirements

3.1 DO-178B

DO-178B defines requirements for the qualification of tools. According to Section 12.2, tool qualification is only needed when processes described in DO-178B are eliminated, reduced, or automated by use of that tool without its output being verified as specified in DO-178B Section 6.

If a tool provides results that are taken as input for subsequent development steps of safety critical items, without any additional evidence of integrity verification, DO-178B, Section 12.2 requires that the tool must be qualified. There is no need to qualify the tool if the output of the tool is verified by other means.

Furthermore, DO-178B classifies software tools as one of two types:

- Software development tools: These are “tools whose output is part of airborne software and thus can introduce errors.”
- Software verification tools: These are "tools that cannot introduce errors, but may fail to detect them."

3.2 FAA guideline N8110.91

FAA guideline N8119.91 provides good examples of guidelines, clarifications and further requirements describing the regulations for a development program. Its stated purpose: “This notice clarifies the application of DO-178B in the area of tool qualification but does not change the intent of DO-178B in this area.”

In the following DER means Designated Engineering Representative, ACO means Aircraft Certification Office.

After discussing background information, it provides guidelines for determining whether a tool should be qualified:

(1) Whether a tool needs to be qualified is independent of the type of the tool (development or verification). There are three questions to ask to determine if a tool needs qualification. If the answer is “Yes” to all of the questions below, the tool should be qualified:

(a) Can the tool insert an error into the airborne software or fail to detect an existing error in the software within the scope of its intended usage?

(b) Will the tool’s output not be verified as specified in Section 6 of DO-178B?

(c) Are processes of DO-178B eliminated, reduced, or automated by the use of the tool? That is, will the output from the tool be used to either meet an objective or replace an objective of DO-178B, Annex A?

Furthermore, N8110.91 states the following:

(2) Once it has been determined that a tool does not require qualification, the remainder of DO-178B Section 12.2 is not applicable to that tool. In order to ensure timely response, the cognizant ACO engineer or DER (if authorized) should be involved early in the certification project’s tool qualification agreements.

(3) The Plan for Software Aspects of Certification (PSAC) should include a listing of all software tools and justification for why each tool does or does not require qualification.

If this leads to a decision that a tool needs qualification, N8110.91 provides a table of qualification criteria, depending on whether the tool is a development tool or a verification tool. One criterion for both of these types is the determinism of the tool. The document discusses the problems of tools with graphical user interfaces – they are by design not deterministic. Qualification may still be possible, but will require a very deep analysis.

4.0 Discussion

Software development is a very complex process with repetitive time-consuming tasks. Existing software tools from the COTS market as well as specifically developed software tools are used to help in this process. Without some of these tools, such as compilers, software development would not even be possible.

In theory, it is adequate to develop such tools according to the very same goals as applied to the airborne software itself. However, if object oriented methods are applied it must be assured that object oriented code will not include dead code and that the code can directly be linked to detail requirements. The upcoming DO-178C will help by providing guidelines to achieve these goals. Nonetheless, the use of COTS tools will still be problematic.

Discussions with engineers involved in the verification and certification process of aircraft systems (e.g. DER, ACO, office of airworthiness) led to the recommendation that the output of a tool should not be trusted at all, even not the output of qualified tools. In any case, the output should be reviewed or proven by other mechanisms. As outlined in 3.1 and 3.2, in that case there is no need to qualify the tool. This significantly reduces cost and effort and allows the use of COTS tools.

5.0 Examples

The following examples provide a base for discussion of some use cases. They are significantly simplified to illustrate their basic features. Besides an example of the commercially mass-produced tool Excel, other examples focus on engineering tools available from Vector.

5.1 Requirements tracking with Excel®

Consider the management of important data in a tool like Excel. This may concern test case management or requirements listing and management. In the latter case, the correctness of the data will directly influence the correctness of the airborne software. Even final testing may not detect lost requirements, since test cases are designed according to the requirements list.

Excel has a graphical user interface. The function or the content of a cell may be influenced by other Excel sheets because of VBA programming features that can override Excel cell application content or functions, even if interfering sheets are not directly linked. This violates the principle of determinism. Ensuring a consistent, completely well-defined environment is practically impossible.

Even if the manufacturer were willing to cooperate in qualification, this would not make any sense with Excel for practical and business reasons.

The solution to the problem however, is quite simple by means of “checker mechanisms”: The requirements list needs to be exported in a simple format that easily can be reviewed, as well as simply compared to master files. After a review of the completeness of the requirements in the exported format, the exported file can serve as a new master file. If modifications have been made, correctness and completeness can easily be verified by comparison of the export file with the master file and the modification item list. The verified and corrected export file after this again, becomes the new master file.

Approaching like this, Excel does not need qualification.

5.2 Communication database design

Engineering of functions communicating via data buses requires communication management. For some data buses, certain standard file formats are used; for others, proprietary solutions exist. Vector offers tools like the CANdb++ editor, network designers and more. For about 20 years now, the CANdb database format DBC is a de facto standard for the CAN data bus in a number of industries. This format can be managed by a tool such as the CANdb++ editor.

Such databases are used by designers of network systems, function developers, developers of LRU software, and test departments. Therefore, errors in the CANdb++ editor could have a critical impact on the airborne software,

which would not necessarily be detected by tests. According to section 3.1 of this paper such a tool belongs to the category of software development tools.

The storage format is an ASCII format. Conducting a review in conjunction with an independent comparison tool makes it possible to verify completeness and identify differences on base of master files (refer to 5.1).

When such methods are applied, the CANdb++ editor can be used without qualification.

5.3 CANoe and VT as debugger and HIL on the developer's desk

Complex electronic functions generally are realized as distributed systems being connected via data buses. When developing software for an electronic device (e.g. LRU) in such distributed environment, one particular difficulty occurs. Parts of the overall functionality may not yet be available, although their data communication is needed on a data bus for design purposes. The communication partner may not be available yet. In those cases a substitute communication partner is necessary.

Another situation is debugging. The location of a bug may not be known, when it occurs somewhere among distributed devices. This results in very high debugging effort.

In these and other cases, a simulation tool like CANoe helps. CANoe provides users a detailed insight to the data bus – from the frame level to the functional level. It provides a communication partner in early development phases, or it may substitute a simulated model for a communication partner to detect bugs in an environment where it is not feasible to acquire the necessary critical values from the physical devices. By supporting several different bus systems in parallel, it helps in developing functions via gateways. And by including additional I/O, either directly or with the help of test hardware such as the Vector VT system; it can be scaled to a comfortable small or medium HIL system – on the developer's desk.

Tool assistance as described above significantly boosts development efficiency. It eliminates or at least reduces debugging problems that would otherwise cause excessive effort. Its primary strength is to identify hidden problems in early development phases, which avoids unnecessary process cycle iterations.

When used as described here, the tool does not "eliminate, reduce, or automate" processes. The actual verification of the communication integrity will be performed by system tests at later process stages, when all devices are at hand. The tool simply improves design process efficiency and helps to gain time and cost savings. So qualification is not required.

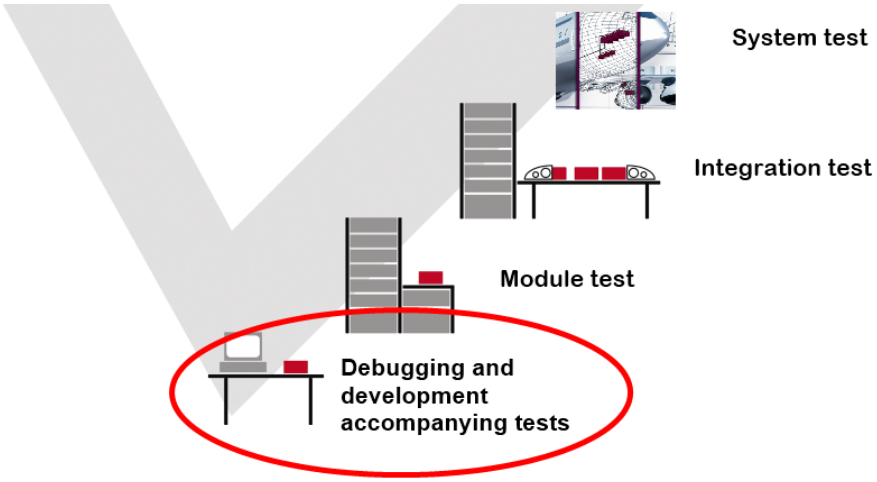

Figure 1: Tests during development in the V model

5.4 CANoe as a test tool

Test tools cannot “insert an error into the airborne software,” but they may “fail to detect an existing error”. So they need to be qualified if “the tool’s output is not verified”. According to 4.0, standard testing procedures should always verify results. For example, this can be done as shown in Figure 2.

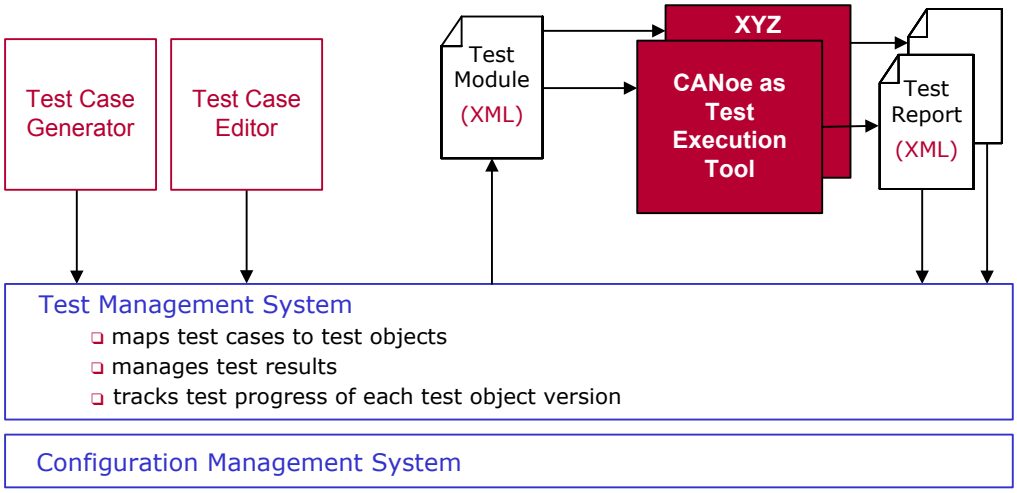

Figure 2: Testing with diversified test execution

Test management provides the test cases, which than are executed by different independently developed test execution tools. Finally, the resulting test reports need to be compared and approved.

This approach provides at least the same – but generally even more - confidence as the use of a single qualified tool. As a matter of fact, the described approach saves much cost compared to the qualification of a tool.

6.0 DO-178C

The new version DO-178C “Software Considerations in Airborne Systems and Equipment Certification” was released in December 2011. Written by many contributing experts it now considers modern software development aspects and provides guidance about their use. The categorization of tools into development tools and testing tools is replaced by a categorization with 3 criteria.

The first step always needs to be the determination if a tool has to be qualified. This is needed when processes of the DO-178C are eliminated, reduced, or automated by the use of the tool without its output being verified as specified in other sections of DO-178C. A tool is qualified only for use on a specific system. Using it in a different system requires a new qualification.

If a tool needs qualification the tool qualification level TQL needs to be determined. As guidance three criteria are defined:

Criteria 1: A tool whose output is part of the airborne software and thus could insert an error.
Criteria 2: A tool that automates verification processes and thus could fail to detect an error, and whose output is used to justify the elimination or reduction of:
1. Verification processes other than that automated by the tool, or
2. Development processes that could have an impact on the airborne software.
Criteria 3: A tool that, within the scope of its intended use, could fail to detect an error.

Five levels of tool qualification, TQL-1 to TQL-5, are identified based on the tool use and its potential impact in the software life cycle processes.

Software Level (DAL)	Criteria		
	1	2	3
A	TQL-1	TQL-4	TQL-5
B	TQL-2	TQL-4	TQL-5
C	TQL-3	TQL-5	TQL-5
D	TQL-4	TQL-5	TQL-5

The details about the tool qualification are described in DO-330 “Software Tool Qualification Considerations”.

The current aircraft programs use previous versions of these documents. Therefore there is not much practical experience with the new documents. First projects will show how they have to be used and which topics will need further interpretation.

7.0 Conclusion

Engineering tools today, are indispensable elements within the development and integration processes of complex systems. They provide a deep insight to the features of the design items and their mutual interoperation. They help to detect design errors at early development stages, and to debug the integrated system at the final phases of the development cycle. In order to achieve an adequate integrity level the tools being applied in the frame of the processes have to prove a high quality. For this purpose all tools being used for the design, integration and verifications of critical system devices have to be listed in the PSAC and must be approved by detailed agreements from the responsible ACO engineers or DER. However, this does not necessarily include the need for thorough tool qualification. The application of commercial-off-the-shelf tools always is admissible if their qualification is substituted by appropriate checker mechanisms as outlined in this paper. Going this way in general saves considerable project costs, while attaining the same or even better confidence and integrity levels than by expensive tool qualification.

8.0 Contacts

Germany
and all countries not named below:

Vector Informatik GmbH
Ingersheimer Str. 24
70499 Stuttgart
GERMANY
Phone: +49 711-80670-0
Fax: +49 711-80670-111
E-mail: info@de.vector.com

France, Belgium, Luxemburg:

Vector France S.A.S.
168, Boulevard Camélinat
92240 Malakoff
FRANCE
Phone: +33 1 42 31 40 00
Fax: +33 1 42 31 40 09
E-mail: information@fr.vector.com

Sweden, Denmark, Norway,
Finland, Iceland:

VecScan AB
Theres Svenssons Gata 9
41755 Göteborg
SWEDEN
Phone: +46 31 764 76 00
Fax: +46 31 764 76 19
E-mail: info@se.vector.com

United Kingdom, Ireland:

Vector GB Ltd.
Rhodium, Central Boulevard
Blythe Valley Park
Solihull, Birmingham
West Midlands B90 8AS
UNITED KINGDOM
Phone: +44 121 50681-50
Fax: +44 121 50681-69

E-mail: info@uk.vector.com

China:

Vector Automotive Technology
(Shanghai) Co., Ltd.
Sunyoung Center
Room 1701, No.398 Jiangsu Road
Changning District
Shanghai 200050
P.R. CHINA
Phone: +86 21 6432 53530
Fax: +86 21 6432 5308
E-mail: info@cn.vector.com

India:

Vector Informatik India Pvt. Ltd.
4/1/1/1, Sutar Icon, Sus Road,
Pashan, Pune - 411 021
INDIA

Phone: +91 20 2587 2023
Fax: +91 20 2587 2025

E-mail: info@in.vector.com

USA, Canada, Mexico:

Vector CANtech, Inc.
39500 Orchard Hill Place, Suite 550
Novi, MI 48375
USA

Phone: +1 248 449 9290
Fax: +1 248 449 9704
E-mail: info@us.vector.com

Japan:

Vector Japan Co. Ltd.
Tennozu Yusen Bldg. 16F
2-2-20 Higashi-shinagawa,
Shinagawa-ku,
Tokyo 140-0002
JAPAN
Phone: +81 3 5769 7800
Fax: +81 3 5769 6975
E-mail: info@jp.vector.com

Korea:

Vector Korea IT Inc.
5F, Gomoas bldg.
12 Hannam-daero 11-gil, Yongsan-gu
Seoul, 140-889
REPUBLIC OF KOREA

Phone: +82 2 807 0600
Fax: +82 2 807 0601
E-mail: info@kr.vector.com