SAE International®

Software Tool Qualification According to ISO 26262

2011-01-1005
Published
04/12/2011

Mirko Conrad, Guido Sandmann and Patrick Munier
The MathWorks, Inc.

doi:10.4271/2011-01-1005

ABSTRACT

International standards that define requirements for the development of safety-related systems typically also define required confidence levels for the software tools used to develop those systems. The standards define—to a greater or lesser extent—procedures to classify, validate, certify, or qualify tools. To date, there is no common approach for tool validation, certification, and qualification across safety standards. Different standards attach different levels of importance to tool validation, certification, and qualification, and suggest different approaches to gain confidence in the tools used. With ISO 26262 "Road Vehicles - Functional Safety" on the horizon, automotive software practitioners will need to understand and implement the new software tool classification and qualification requirements laid out in this standard. ISO 26262 is the adaptation of IEC 61508 to comply with needs specific to the application sector of electric / electronic systems (E/E systems) within road vehicles. This adaptation applies to all activities during the safety lifecycle of systems composed of electrical, electronic, and software elements that provide safety-related functions.
Clause 11 of ISO 26262-8 provides guidance on software tool classification and qualification. The clause applies, if the safety lifecycle incorporates using a software tool, such that (1) activities or tasks required by ISO 26262 rely on the correct functioning of that tool, and (2) relevant outputs of that tool are not fully examined or verified. This paper describes the tool classification and qualification approach of ISO/FDIS 26262 and summarizes the authors' firsthand experiences with implementing this approach for development and verification tools.

ISO/FDIS 26262 TOOL QUALIFICATION APPROACH

This section provides a brief overview on the tool qualification approach as outlined in the final draft standard.

International standards that define requirements for the development of safety-related systems typically also define the required level of confidence for the software tools used to develop these systems. To varying degrees, these standards define procedures to classify, validate, certify, or qualify tools. To date, there is no common approach for tool validation, certification, and qualification that can be applied to all safety standards. Different standards attach different levels of importance to these objectives and suggest different approaches to gain confidence in the tools used [CMR10].

ISO/FDIS 26262 "Road Vehicles - Functional Safety" [ISO/DIS 26262] is the adaptation of IEC 61508 [IEC 61508] to comply with needs specific to the application sector of electric / electronic systems (E/E systems) within road vehicles. This adaptation applies to all activities during the safety lifecycle of systems composed of electrical, electronic, and software elements that provide safety-related functions.

As per ISO/FDIS 26262-8, 11, a software tool (or a software tool chain) used in the safety lifecycle, in a way that (1) activities or tasks required by ISO 26262 rely on the correct functioning of that tool, and (2) relevant outputs of that tool are not fully examined or verified, need to be assessed, classified, and potentially qualified. ISO/FDIS 26262-8 provides criteria to determine the required level

of confidence in a software tool and means to qualify the tool in order to create evidence that the tool is suitable for use in the development of safety-related software.[1]

The approach can be divided into a tool classification (evaluation of the software tool by analysis) step and a tool qualification step.

Tool Classification

First, the intended usage of the tool (use cases) needs to be documented, analyzed, and evaluated to ascertain

- The possibility that a malfunction in the software tool can introduce or fail to detect errors in the safety-related system being developed. The result is expressed using one out of two Tool Impact classes (TI1 or TI2).
- The confidence in measures to prevent or detect malfunctioning and corresponding erroneous output. The result is expressed using one out of three Tool Error Detection classes (TD1, TD2 or TD3).

As a result of this analysis, a required Tool Confidence Level is determined. The tool confidence level is classified using one out of three Tool Confidence Levels TCL1, TCL2, or TCL3.

Tool Qualification

Tools with the lowest possible TCL (i.e., TCL 1) do not require subsequent tool qualification. For all other TCLs, formalized tool qualification is necessary. The selection of appropriate tool qualification methods depends on the required TCL and on the Automotive Safety Integrity Level (ASIL) of the safety-related software to be developed using the software tool.

WORK PRODUCTS

The ISO/FDIS 26262 tool qualification process requires the creation of two tool qualification work products, a Criteria Evaluation Report documenting the tool classification and a Qualification Report documenting the tool qualification.

Software Tool Criteria Evaluation Report

The software tool criteria evaluation, i.e. tool classification, is crucial for the ISO 26262 tool qualification approach. The analysis must be carried out for all tools used in the software life cycle to determine whether or not a formal tool qualification is necessary.

Tool Use Cases (UC)

The evaluation results depend on how the tool is being used during the development of the safety-related system. Therefore, the intended usage of the tool needs to be documented (ISO/FDIS 26262-8, 11.4.5.1).

Tool Impact (TI)

First, it shall be evaluated whether a malfunction in the software tool can introduce or fail to detect errors in the system being developed.

If it can be argued that malfunctions in the tool cannot introduce errors in the system or prevent those errors from being detected, *Tool Impact* class *TI1* shall be chosen. Otherwise, the tool impact class is *TI2* (ISO/FDIS 26262-8, 11.4.5.2a).

Tool Error Detection (TD)

Second, the use cases for the software tool shall be analyzed to determine the confidence in measures that

- Prevent the software tool from malfunctioning and producing erroneous output or
- Detect that the software tool has malfunctioned and produced erroneous output.

---

[1] The tool qualification approach in ISO/FDIS 26262-8:2010 differs from the one used in the earlier DIS version of the standard. See [Sau09, CMR10] for information on tool qualification according to ISO/DIS 26262-8:2009.

Both tool-internal measures (e.g., monitoring) and tool-external measures implemented as part of the software lifecycle (e.g., guidelines, tests, and reviews) to prevent or detect errors should be considered in the analysis.

The degree of confidence in the prevention and detection measures determines the *Tool Error Detection* class. Class *TD1* shall be selected if there is a high degree of confidence, *TD2* if there is a medium degree of confidence, and *TD3* in all other cases (ISO/FDIS 26262-8, 11.4.5.2.b).[2]

Tool Confidence Level (TCL)

When TI and TD have been identified, the *tool confidence level* can be determined following the schematic provided in the left hand side of Fig. 1 (ISO/FDIS 26262-8, 11.4.5.5). When multiple use cases for a tool exist, there can be multiple TCLs. To determine the required tool qualification measures, the maximum TCL required (TCL_{REQ}) to support these use cases needs to be established.

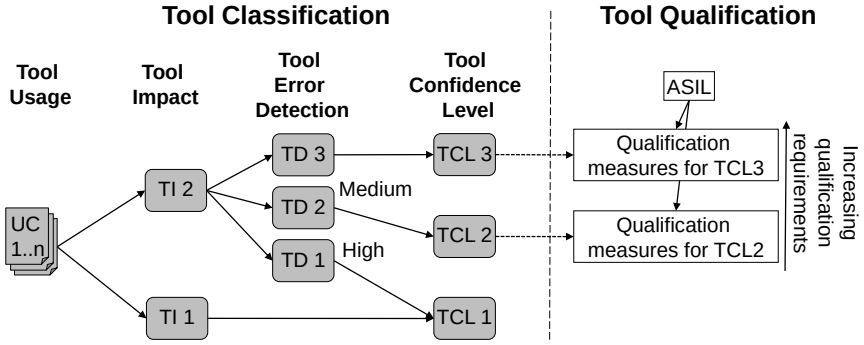

Figure 1 - ISO/FDIS 26262 Tool Classification Scheme.

The tool classification process and its results are documented in the *Software Tool Criteria Evaluation Report*.

Software Tool Qualification Report

A tool classified as TCL1 does not require specific *tool qualification methods* to be carried out. For software tools classified at TCL2 or TCL3, at least one dedicated *tool qualification method* has to be applied (see right hand side of Fig. 1). As defined in ISO/FDIS 26262-8, 11.4.6.1, the four permitted methods are (1a) *Increased confidence from use*, (1b) *Evaluation of the tool development process*, (1c) *Validation of the software tool*, and (1d) *Development in accordance with a safety standard*.

The selection of a method is guided by the ASIL classification of the application to be developed and the required TCL resulting from the tool classification. The specific recommendations for all TCL-ASIL combinations are listed in ISO/FDIS 26262-8, tables 4 and 5. They are summarized in Table 1 of this paper. Some of the methods may be easier to implement for commercial-off-the-shelf (COTS) tools, whereas others might be preferable for proprietary solutions developed and maintained by OEMs or suppliers. The selected tool qualification methods must be documented in the *Software Tool Qualification Report*.

The software tool qualification report also documents the actual tool qualification; that is, it provides evidence that the tool qualification methods were carried out as planned. Usage constraints and malfunctions identified during the qualification, if any, need to be documented.

[2] ISO/DIS 26262-8:2009 distinguished between 4 tool confidence levels. However, this was perceived as too complicated for practical use.

Table 1 – Recommendations for Tool Qualification Methods according to ISO 26262.
(+: recommended; ++: highly recommended)

(1a) Increased confidence from use

	ASIL A	ASIL B	ASIL C	ASIL D
TCL 2	++	++	++	+
TCL 3	++	++	+	+

(1b) Evaluation of the development process

	ASIL A	ASIL B	ASIL C	ASIL D
TCL 2	++	++	++	+
TCL 3	++	++	+	+

(1c) Validation of the software tool

	ASIL A	ASIL B	ASIL C	ASIL D
TCL 2	+	+	+	++
TCL 3	+	+	++	++

(1d) Development in compliance with a safety standard

	ASIL A	ASIL B	ASIL C	ASIL D
TCL 2	+	+	+	++
TCL 3	+	+	++	++

Note that the ISO/FDIS 26262 tool classification scheme and the tool qualification methods are agnostic of tool categories. That is, unlike other standards such as DO-178B, there is no distinction between development and verification tools. However, the ISO/FDIS 26262 tool qualification approach takes the interdependencies between tools into account. If a software tool T2 is used to verify the output of another software tool T1, the interdependency between these tools need to be considered when evaluating T2 (i.e. the downstream tool).

EXPERIENCES WITH QUALIFYING COTS TOOLS ACCORDING TO ISO/DIS 26262

MathWorks automotive industry customers have expressed a need to comply with the upcoming ISO 26262 standard [MW09] and for tools qualified as per ISO 26262 in particular. In order to support this customer need, the authors instantiated and implemented the generic ISO 26262 tool qualification approach.

In this section, the authors report their firsthand experiences with the qualification of COTS software tools according to ISO/FDIS 26262. These experiences were gained during the qualification of Embedded Coder™ software for code generation and Polyspace Client™ for C/C++ and Polyspace Server™ for C/C++ software for code verification from MathWorks. The initial qualification activities started in 2009 and were adapted to the FDIS version of ISO 26262 in 2010/2011.

IMPLEMENTATION OF THE ISO 26262 TOOL QUALIFICATION APPROACH

ISO 26262 allows different levels of qualification, including a self-qualification by the tool user (first party qualification). However, users of COTS tools expect the tool vendor to provide a tool qualification package that can be instantiated with minimal effort. Since no ISO 26262 tool qualification examples or best practices were available in 2009, the authors had to break new ground. To ensure the credibility of the approach and the impartiality of the results, MathWorks decided to collaborate with a recognized, accredited certification body for developing the tool qualification approach. MathWorks proposed a tool qualification approach and submitted it to TÜV SÜD for independent assessment and approval. TÜV SÜD was chosen because of its experience with software tool certifications and qualifications according to various standards.

TÜV SÜD had previously certified the Polyspace and Real-Time Workshop Embedded Coder products as suitable for use in developing safety-related software according to IEC 61508 and derivative standards. When the ISO 26262 tool qualification activities were launched in 2009, one goal was to use existing approaches and artifacts that were developed for certification packages for IEC

61508 and qualification kits for DO-178B whenever feasible. The decision to work with TÜV SÜD on ISO 26262 was bolstered by its earlier work on these initiatives.

ISO 26262 calls for a project-specific classification and qualification of software tools where applicable. However, for COTS tool vendors, qualification only makes sense if it can be leveraged by several customers and on multiple projects. To reconcile these competing requirements, MathWorks used a generic qualification approach based on one or more common, typical tool use cases and a reference workflow to be utilized by the tool user when developing or verifying safety-related software. In terms of ISO 26262, the reference workflow describes error prevention and error detection measures applied in conjunction with using the tool itself.

The tools were classified assuming that they are being used as specified in the typical use case(s) and tool usage is being supported by the error prevention and detection methods described in the reference workflows. Best practices for tool classification developed by automotive companies as described in [Mai09] were incorporated. The maximum required TCL resulting from the tool classification was used to determine the necessary tool qualification methods. For both products, a combination of the tool qualification methods (1b) *Evaluation of the tool development process* and (1c) *Validation of the software tool* were used. The tool qualification artifacts were created by the tool vendor and submitted to TÜV SÜD for review and approval. TÜV SÜD confirmed the adequateness of the artifacts in the certification report.

Automotive software practitioners can directly leverage the qualification by referencing the certificate and the certificate report, if they are using the tool within the constraints of the use case(s) and the reference workflow. To further support tool users, MathWorks created a tool qualification package (TQP) that contains templates for all tool qualification artifacts. The user needs to review the templates for applicability to the application under consideration, and to instantiate the information. The tool qualification package—as well as evidence documenting the independent assessment by TÜV SÜD of the tool qualification measures carried out by MathWorks—are made available as part of the IEC Certification Kit for ISO 26262 and IEC 61508. If the user applies the product differently, the tool classification needs to be carried out according to the actual use case(s). However, if the required TCL derived from the actual use cases is equal to or lower than the TCL that resulted from the typical use cases, the tool qualification can still be used.

The tool qualification approach described above can be characterized as a prequalification of the software tool by the COTS tool vendor, which can be applied and tailored by the tool user.

RELATED WORK

Details of the qualification approach discussed above are given in [CMR10] and [MW09]. The reference workflow is discussed in [Con09] and [CS09]. [CMR10] also discusses issues and open questions related to the tool qualification approach outlined in ISO/DIS 26262. [Sau09] and [Sau10] provide general discussions of ISO 26262 including tool qualification aspects. [Mai09] describes an industry example for ISO 26262 COTS tool classification. Trade magazine articles on recent tool qualification activities include [KKG10] and [BB10].

SUMMARY

With the advent of ISO 26262, automotive software practitioners need to understand how to implement the tool qualification requirements of this standard in practice. The authors reviewed the ISO/FDIS 26262 tool qualification approach and reported on their experiences with one of the first (if not the first) tool qualifications of commercially available production code generation and verification tools according to this emerging standard.

The Embedded Coder, Polyspace Client for C/C+, and Polyspace Server for C/C++ products for code generation and verification have been prequalified by the tool vendor in collaboration with an accredited certification body. To leverage the prequalification in ISO 26262 projects, tool users can instantiate and tailor a tool qualification package provided by MathWorks. Using such prequalified tools in a Model-Based Design tool chain significantly reduces the effort and costs incurred by the user for tool qualification.

The authors believe that the definition of reference workflows containing suitable verification and validation measures to be used in combination with the qualified tools provides practitioners with the necessary guidance to successfully apply Model-Based Design and advanced code verification tools in projects that need to comply with the requirements of ISO/FDIS 26262.

REFERENCES

1. [BB10] A. Bärwald, M. Beine: Sichere Codegenerierung. Automotive 1-2 2010, 30-33

2. [CMR10] M. Conrad, P. Munier, F. Rauch: Qualifying Software Tools According to ISO 26262. Proc. Model-based Development of Embedded Systems (MBEES10), Schloß Dagstuhl, Germany, Feb. 2010
3. [Con09]: M. Conrad: Testing-based translation validation of generated code in the context of IEC 61508. Formal Methods in System Design, 2009. DOI 10.1007/s10703-009-0082-0
4. [CS09]: Conrad, G. Sandmann: A Verification and Validation Workflow for IEC 61508 Applications. SAE Technical Paper #2009-01-0271, SAE World Congress 2009
5. [DO-178B]: RTCA/DO-178B. Software Considerations in Airborne Systems and Equipment Certification. 1992
6. [HBM10]: H. Hauff, A. Bärwald, J. Mottok: Sichere Werkzeuge für sichere Systeme! – Qualifizierung und Zertifizierung von Software-Entwicklungswerkzeugen. Automotive 1-2 2010, pp. 34-39
7. [IEC 61508]: IEC 61508:1998. Int. Standard Functional safety of electrical/ electronic/ programmable electronic safety-related systems. 1998-2000.
8. [ISO/DIS 26262]: ISO/DIS 26262. Draft In. Standard Road vehicles — Functional safety. 2009.
9. [KKG10] J. Klarmann, S. Kriso, M. Gebhardt: Qualification of development tools as per ISO 26262. REAL TIMES, 1/2010, pp. 28-20
10. [Mai09]: M. Maihöfer: Umgang mit Entwicklungswerkzeugen in Software-Entwicklungsprozessen der Automobilindustrie - ISO DIS 26262, Band 8, Kapitel 11: Inhalt, Bewertung, Auswirkung und Umsetzung (in German). EOROFORUM Konferenz 'Funktionale Sicherheit nach ISO/DIS 26262', Stuttgart, Germany, September 2009
11. [MBD]: Model-Based Design web page. The MathWorks Inc., www.mathworks.com/applications/controldesign/description
12. [KZ09]: A. Kornecki, J. Zalewski: Certification of software for real-time safety-critical systems: state of the art. Innovations in Systems and Software Engineering (2009) 5:149–161
13. [PM99]: Y. Papadopoulos, J. A. McDermid: The Potential for a Generic Approach to Certification of Safety-Critical Systems in the Transportation Sector. Journal of Reliability Engineering and System Safety 63 (1999) 47-66
14. [RTW-EC]: Real-Time Workshop® Embedded Coder™ product page. The MathWorks Inc., www.mathworks.com/products/rtwembedded
15. [Sau09]: J. Sauler: Die ISO 26262 für Automotive kommt! Elektronikpraxis TV (in German), 2009
    Part 1: www.youtube.com/watch?v=wqbNrgRcEVo
    Part 2: www.youtube.com/watch?v=vWkdIRINb8o
16. [Sau10]: J. Sauler: Alle Fakten zur neuen Sicherheits-Norm für die Autoindustrie ISO 26262 (In German), Interview, Elektronik Praxis, 3.2.2010
17. [MW08]: The MathWorks Real-Time Workshop Embedded Coder Certified By TÜV SÜD Automotive GmbH. Press Release, The MathWorks, Inc., 2008
    www.mathworks.com/company/pressroom/articles/article31189.html
18. [MW09]: The MathWorks Real-Time Workshop Embedded Coder and Polyspace Products Qualified According To ISO 26262. Press release, The MathWorks, Inc., 2009
    www.mathworks.com/company/pressroom/articles/article39270.html

CONTACT INFORMATION

Dr. Mirko Conrad, Development Manager, Simulink Certification and Standards, MathWorks

Mirko Conrad is a development manager at MathWorks in Natick, MA, where he leads the Simulink Certification and Standards team. He has previous automotive experience as a senior research scientist and project manager at Daimler-Benz / DaimlerChrysler. Mirko holds a Ph.D. in engineering (Dr.-Ing.) and an M.Sc. in computer studies (Dipl.-Inform.) from Technical University in Berlin, Germany. He is also a visiting lecturer at Humboldt University in Berlin. His publication record includes more than 60 papers on automotive software engineering, Model-Based Design, and safety-related software. He is a member of managing team of the Special Interest Group for Automotive Software Engineering in the German Computer Society (GI-ASE) and a former member of the ISO 26262 sub-working group on software.

E-mail: Mirko.Conrad@ mathworks.com

Guido Sandmann, Automotive Marketing Manager, EMEA, MathWorks GmbH

Guido Sandmann works for MathWorks as Automotive Marketing Manager responsible for the European region. In this technical marketing role he is responsible for message creation concerning MathWorks products and solutions dedicated to the automotive industry. He works closely with MathWorks customer facing organizations as well as with customers directly, discussing their requirements and desired improvements to the company's product portfolio.
Before joining MathWorks, he worked for OSC – Embedded Systems, a company with expertise in verification and testing based on formal methods, and for dSPACE.
Guido Sandmann holds a Diploma degree from the University of Oldenburg as Computer Scientist.

E-mail: Guido.Sandmann@mathworks.de