

Software Tool Qualification According to ISO 26262

An Experience Report

Mirko Conrad

The MathWorks, Inc.

Natick, MA, USA

mirko.conrad@mathworks.com

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 these systems. With ISO 26262 on the horizon, the new software tool qualification requirements laid out in this standard need to be understood and implemented by practitioners in the automotive industry.

This paper summarizes the tool qualification approach of ISO/DIS 26262 and discusses the author’s first-hand experiences with qualifying development and verification tools according to this emerging standard.

Keywords - ISO 26262; tool qualification; Embedded Coder; Polyspace; IEC Certification Kit

I. ISO/DIS 26262 TOOL QUALIFICATION APPROACH

This section provides a brief overview about the tool qualification approach as outlined in the ISO 26262 draft international standard (ISO/DIS 26262).

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 these systems. These standards define - to a greater or lesser extent – procedures to validate, certify, or qualify tools.

Up to now, there is no common approach for tool validation / certification / qualification across safety standards. Different standards attach different levels of importance to tool validation / certification / qualification and suggest different approaches to gain confidence in the tools used [CMR10].

ISO/DIS 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 safety-related systems comprised of electrical, electronic, and software elements that provide safety-related functions [ISO/DIS 26262-1].

As per ISO/DIS 26262, the objective of tool qualification is to provide evidence that a software tool (or a software tool chain) is suitable for use in the development of safety-related

software. ISO/DIS 26262-8, 11 [ISO/DIS 26262-8] provides guidance on *software tool qualification*¹.

To provide the required evidence, all software tools need to be assessed in a project-specific manner. First, the use cases for a tool need to be documented. Based on the documented use cases, it shall be evaluated if and how a malfunction in the software tool or an erroneous output produced by the tool can violate a safety requirement. Then, the probability of preventing or detecting these malfunctions and erroneous outputs of the tool needs to be evaluated. As a result of this analysis, a required *tool confidence level* (TCL) is determined.

Only tools with the lowest possible TCL, i.e. TCL 1, do not require any additional tool qualification. For all other TCLs, i.e. TCL 2 ... 4, 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. Permissible tool qualification methods, listed in tables 2, 3, and 4 of ISO/DIS 26262-8, comprise:

- a) Increased confidence from use
- b) Evaluation of the development process
- c) Validation of the software tool
- d) Development in compliance with a safety standard

Tool qualification can be carried out for individual tools as well as for tool chains or sets of tools.

The ISO/DIS 26262 tool qualification process requires the creation of the following *tool qualification work products* (ISO/DIS 26262-8, 11.5; see Figure 2 for a summary):

- Software Tool Qualification Plan
- Software Tool Documentation
- Software Tool Classification Analysis
- Software Tool Qualification Report

The following subsections provide details of the ISO/DIS 26262 tool qualification approach. [Sau09] touches upon this topic as well.

A. Software Tool Qualification Plan (STQP)

The software tool qualification plan is a planning document created in an early phase of the development of the safety-related system.

¹ Although ISO/DIS 26262 is a derivative standard of IEC 61508, tool certification / qualification approaches in these two standards differ significantly [CMR10].

Besides stating the applicant, and the application under consideration, it specifies the tool and tool version to be qualified, the intended configuration and operational environment. In this sense, the STQP shares conceptual similarities with tool qualification plans used in DO-178B [DO-178B] projects.

The tool qualification plan also lists the intended tool use cases which will be analyzed in the course of the tool classification process.

The STQP also is supposed to list the tool qualification methods and available means to detect malfunctions or erroneous output of the tool. Because this information is not yet available in the planning phase, the STQP typically cannot be finalized at this point. As a consequence, a first, incomplete STQP version containing the planning information need to be augmented in a second iteration with the tool classification and qualification summary information.

B. Software Tool Documentation (STD)

The software tool documentation comprises different information that the applicant may need when using the tool. It comprises information such as tool overview, available tool documentation set, operational environment and constraints, installation instructions, known issues.

The information contained in the STD can be used to check whether the use cases, configurations and operational environment listed in the STQP are indeed supported by the

tool. The STD has similarities to the description of tool operational requirements as per DO-178B.

C. Software Tool Classification Analysis (STCA)

The software tool classification is crucial for the ISO 26262 tool qualification approach. The STCA needs to be carried out for all tools used in the software life cycle. The outcome of the analysis determines whether or not a formal tool qualification is necessary.

The STCA depends on the particular tool use cases used during the development of the application under consideration. The analysis starts with analyzing the tool use cases specified in the software tool qualification plan. This analysis shall (1) evaluate whether a malfunction in the tool or erroneous output produced by the tool can result in the violation of a safety requirement and (2) determine the likelihood of preventing or detecting these errors. 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 tool classification facilitates the determination of the necessary *tool confidence level* (TCL) and follows the schematics provided in Fig. 1. The TCL in conjunction with the Automotive Safety Integrity Level (ASIL) of the safety-related software developed using the software tool, determines the selection of the appropriate tool qualification methods.

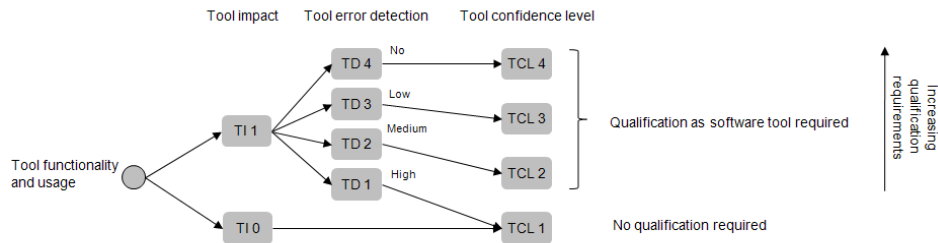


Figure 1. ISO/DIS 26262 Tool Classification Scheme

1) Tool Impact (TI)

First, the intended use cases for the tool shall be analyzed to determine if a safety requirement can be violated if the software tool is malfunctioning or producing erroneous output. If a violation of a safety requirement is not possible, *tool impact* TI0 shall be chosen. Otherwise, the tool impact is TI1 (ISO/DIS 26262-8, 11.4.3.2.a).

2) Tool Error Detection (TD)

Second, the intended software tool use cases shall be analyzed to determine the probability of preventing or detecting that the software tool is malfunctioning or producing erroneous output. The degree of confidence, that a malfunction or an erroneous output from the tool can be prevented or detected, determines *the tool error detection* TD as outlined in Table 1 (ISO/DIS 26262-8, 11.4.3.2.b).

TABLE I. TOOL ERROR DETECTION LEVELS

Degree of confidence	Tool error detection
high	TD1
medium	TD2
low	TD3
no systematic verification measures in subsequent development phases; malfunctions or erroneous outputs can only be detected randomly	TD4

3) *Tool Confidence Level (TCL)*

If TI and TD have been chosen, the *tool confidence level* (TCL) can be determined following the schematics provided in Figure 1 (ISO 26262-8, 11.4.3.4).

Having multiple use cases for a tool can potentially result in multiple TCLs. To determine the required tool qualification measures, the maximum TCL required (TCL_{REQ}) to support these use cases needs to be established. TCL_{REQ} needs to be documented in the STQP.

4) *Tool Qualification Methods (TCM)*

A tool classified at TCL1 does not require specific *tool qualification methods* to be carried out.

For software tools classified at any of the other tool confidence levels, at least one dedicated *tool qualification method* has to be applied. The four permitted methods are

‘Increased confidence from use’, ‘Evaluation of the development process’, ‘Validation of the software tool’, and ‘Development in compliance with a safety standard’.

The selection of a method is guided by the Automotive Safety Integrity Level (ASIL) classification of the application to be developed, and the required TCL resulting from the STQA. The specific recommendations for all TCL-ASIL combinations are listed in ISO/DIS 26262-8, tables 2, 3, and 4 and summarized in Table 2. As an example, to qualify a tool classified at TCL 3 up to ASIL D, methods (1b), (1c), (1d), or a suitable combination of these would be highly recommended. However, some of the methods may be easier to implement in case of COTS tools whereas others might be preferable for proprietary solutions developed and maintained by OEMs or suppliers.

TABLE II. RECOMMENDATIONS FOR TOOL QUALIFICATION METHODS ACCORDING TO ISO 26262

(1a) Increased confidence from use

	ASIL A	ASIL B	ASIL C	ASIL D
TCL 2	++	++	++	++
TCL 3	++	++	++	+
TCL 4	++	++	+	O

(1b) Evaluation of the development process

	ASIL A	ASIL B	ASIL C	ASIL D
TCL 2	++	++	++	++
TCL 3	++	++	++	++
TCL 4	++	++	++	+

(1c) Validation of the software tool

	ASIL A	ASIL B	ASIL C	ASIL D
TCL 2	+	+	+	+
TCL 3	+	+	+	++
TCL 4	+	+	++	++

(1d) Development in compliance with a safety standard

	ASIL A	ASIL B	ASIL C	ASIL D
TCL 2	+	+	+	+
TCL 3	+	+	+	++
TCL 4	+	+	++	++

(o ... no recommendation for / against usage; + ... recommended; ++ ... highly recommended)

The selected tool qualification method(s) need(s) to be documented in the STQP.

N.B. the ISO/DIS 26262 tool classification scheme as well as the tool qualification methods are agnostic of tool categories. I.e., in contrast to other standards such as DO-178B, there is no distinction between development and verification tools. It’s also noteworthy that the degree of recommendation for (1d) ‘Increased confidence use’ decreases for the higher ASILs. This is different from the IEC 61508 base standard, where this method is highly recommended for all integrity levels.

D. *Software Tool Qualification Report (STQR)*

The software tool qualification report documents the actual tool qualification. Usage constraints and malfunctions

identified during the qualification, if any, need to be documented in the STQR.

Fig. 2 summarizes the tool qualification activities and corresponding work products.

II. EXPERIENCES WITH QUALIFYING COTS TOOLS ACCORDING TO ISO/DIS 26262: A PRACTITIONER’S PERSPECTIVE

In this section, the author reports his first hand experience with the qualification of commercial-off-the-shelf (COTS) software tools according to ISO/DIS 26262. These experiences were gained during the qualification of the Real-Time Workshop® Embedded Coder™ code generator and the Polyspace® Client/Server for C/C+ code verifiers from MathWorks. The initial qualification activities occurred in 2009 and were continued in 2010.

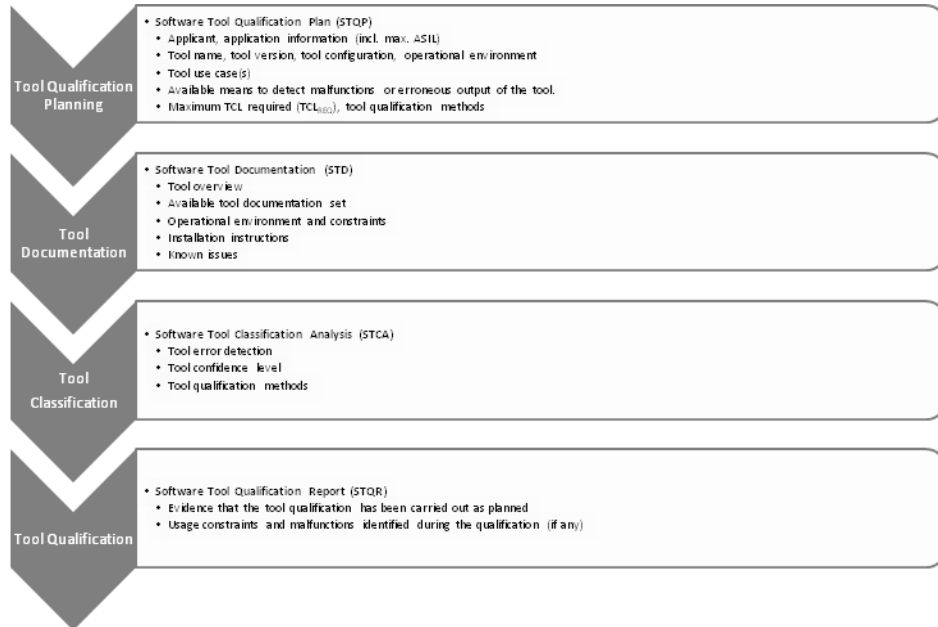


Figure 2. ISO/DIS 26262 Tool Qualification and Work Products (Overview)

MathWorks automotive industry customers have expressed their need for compliance with the upcoming ISO 26262 standard [TMW09] and for tools qualified as per ISO 26262 in particular. In order to support this customer need, the generic ISO 26262 tool qualification approach had to be instantiated and implemented.

A. Implementation of the ISO 26262 Tool Qualification Approach

ISO 26262 allows different levels of qualification, including a self-qualification by the tool user (1st party qualification). However, users of COTS tools expect the tool vendor to provide a tool qualification package that can be instantiated with limited effort.

Since no ISO 26262 tool qualification examples or best practices were available in 2009, MathWorks had to break new ground. To increase the credibility of the approach, MathWorks decided to co-operate with a recognized, accredited certification body when developing the tool qualification approach. So, MathWorks submitted the tool qualification approach to TÜV SÜD Automotive GmbH for assessment and approval. TÜV SÜD was chosen due to their reputation for software tool certifications / qualifications according to various standards.

MathWorks supports customers developing high-integrity systems in several industries and according to different standards. Tool certification packages for IEC 61508 and qualification kits for DO-178B did already exist when the ISO 26262 tool qualification activities were launched in 2009. Therefore, it was desirable to utilize the existing certification / qualification approaches and artifacts developed for these other standards whenever feasible.

The Polyspace and Real-Time Workshop Embedded Coder products have already been certified by TÜV as suitable for use to develop safety-related software according to IEC 61508 and derivative standards. Therefore, it was self-evident to extend this TÜV certification to ISO 26262.

ISO 26262 calls for a project-specific evaluation and - should the occasion arise - qualification of software tools. However, for COTS tool vendors, qualification makes only sense if it can be leveraged by several customers and projects.

To bridge this gap, a generic qualification approach was developed based on one or more common, typical tool use cases and a reference workflow to be utilized by the tool user when using the tool for developing or verifying safety-related software. In terms of ISO 26262, the reference workflow describes error prevention and error detection measures used in conjunction with using the tool itself.

The tool was classified assuming that the tool is 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 workflow. The maximum required TCL resulting from the tool classification was used to determine the necessary tool qualification methods.

The tool qualification artifacts were created by the tool vendor and submitted to TÜV SÜD for review and approval. TÜV SÜD stated the adequacy of the artifacts in the certificate / certification report.

A user can directly leverage the qualification if the tool is being used within the perimeter of the use case(s) and the reference workflow by referencing the certificate and the certificate report. To further support tool uses, MathWorks created a tool qualification package (TQP) containing templates for all tool qualification artifacts described in

section 1.1 – 1.4. The user needs to review the templates for applicability to the application under consideration, and to instantiate the information. Certificate, certification report and tool qualification package are available as part of the IEC Certification Kit for IEC 61508 and ISO 26262.

If the user deviates from the use case(s) and the reference workflow, 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 or lower than the TCL that resulted from the typical use cases, the tool qualification can still be leveraged.

The tool qualification approach described above can be characterized as a pre-qualification of the software tool by the COTS tool vendor that can be utilized and tailored by the tool user.

A possible way to leverage existing certification approaches was to add the ISO 26262 tool qualification on top of the existing IEC 61508 in-context tool certifications.

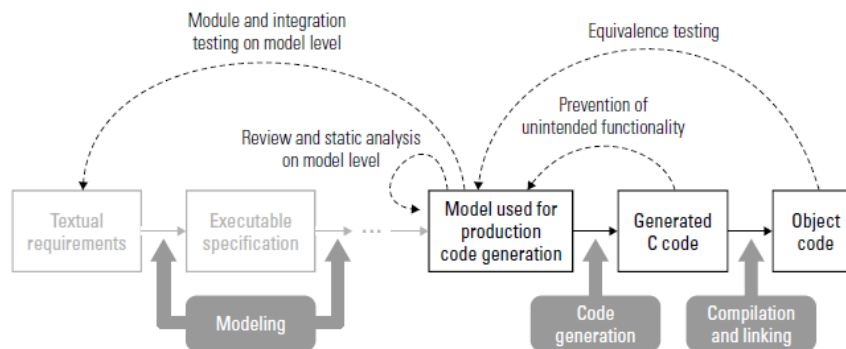


Figure 3. Reference Workflow for Application Specific Verification and Validation of Models and Generated Code

The verification and validation measures described in this reference workflow help to detect or prevent potential malfunctions or erroneous outputs of the code generator. This description was used to determine the tool error detection (TD) for the code generator.

According to the certification report, applying the entire workflow for Application-Specific Verification and Validation of Models and Generated Code provides a high degree of confidence that potential malfunctions or erroneous output of the code generator can be detected or prevented, i.e. the tool error detection is TD1 resulting in a tool confidence level of TCL1. Tool qualification for the code generator can be claimed without carrying out additional tool qualification methods.

To give the user more flexibility when integrating Real-Time Workshop Embedded Coder into their development processes, MathWorks also aimed at supporting use cases that only utilize a suitable subset of the reference workflow². Assuming that the selected subset provides a medium degree of confidence to detect or prevent potential malfunctions or

These existing certifications are based on specific workflows (*reference workflows*) to be utilized by the applicant when using the tool for developing or verifying software for IEC 61508 applications. In the context of ISO 26262 tool qualification, these workflows can be re-used to describe and limit the intended tool use cases as well as to list available means to detect malfunctions or erroneous output of the tool.

In the following, we will illustrate this approach using the Real-Time Workshop Embedded Coder code generator as an example:

The reference workflow describes a workflow for application-specific verification and validation of models and generated code developed using the Simulink® modeling environment and the Real-Time Workshop Embedded Coder C code generator. The main constructive development activities as well as the corresponding verification and validation activities are summarized in Fig. 3. [Con09, CS09] provide more detailed discussions.

erroneous output of the code generator, the tool error detection is TD2 resulting in a tool confidence level of TCL2. In this case, tool qualification methods need to be selected according to Table. 2.

The PolySpace verifiers for C/C++ were classified at TCL2 as well.

In case of both products, a combination of the tool qualification methods (1b) Evaluation of the development process and (1c) Validation of the software tool were used.

The *assessment of the development process* was carried out by TÜV SÜD as part of the IEC 61508 tool certification procedure. The assessment criteria were enhanced to match the ISO 26262 requirements.

The *software tool validation* was carried out differently for the two tools. In case of PolySpace, for example, existing test artifacts that were created as part of the DO-178B tool qualification kit for PolySpace were reused.

For both tools, the tool classification reports, artifacts to demonstrate compliance with the documented development processes and evidence for the validation of the tools were submitted to TÜV SÜD for review. TÜV SÜD assessed the artifacts and stated their suitability to claim tool qualification in the certification reports for Real-Time Workshop Embedded Coder and PolySpace.

² The applicant needs to document the chosen workflow subset in a conformance demonstration template that ships with the tool qualification package.

To further support users of Real-Time Workshop Embedded Coder when claiming tool qualification, a *Tool Qualification Package* was created. The package contains templates for the tool qualification work products identified by ISO/DIS 26262-8 as well as evidence documenting the independent assessment by TÜV SÜD of the tool qualification measures carried out by MathWorks. In the tool qualification package, the tool qualification work products are integrated into one single document to account for the overlap and dependencies between the different work products (see section 4 for a detailed discussion of this issue).

B. Discussion

The author sees the following issues with the tool qualification approach outlined in ISO/DIS 26262:

No formal qualification credits: Similar to IEC 61508, ISO/DIS 26262 has a generic requirement for tool qualification, but doesn't offer formal certification credits in exchange for tool qualification. This way there is little incentive for applicants to certify tools if it can be avoided. As such, it is feared that the tool classification will be dressed up to reduce or avoid tool qualification methods.

No distinction between verification and development tools: Unlike DO-178B and IEC 61508 Ed. 2.0, ISO/DIS 26262 does not differentiate between tools that can introduce errors (aka development tools) and tools that can only fail to detect errors (aka verification tools). Imposing the same tool qualification requirements for both classes of tools does not account for the different criticalities of these tool categories. Providing a generic reduction of the TCL for verification tools could be one means of addressing this issue. If this is not addressed properly in new versions of the standard, the author is concerned that the tool categories are treated differently when assigning a TD level. However, this would be less transparent than a clear statement in the standard itself.

Re-using the same arguments for several tools: A single means to detect or prevent errors in a tool could be used in the argumentation to lower the tool confidence level for several tools. This seems to result in a lower confidence for the overall tool chain, when compared with using different means to lower the confidence levels of different tools. However, ISO/DIS 26262 does not seem to provide any guidance on how to deal with these cases of 'double accounting'. It would be helpful to have some kind of 'TD/TCL arithmetic' that allows the determination of tool confidence and tool error detection levels when combining tools or tool chains that have been classified already.

Difficulty in providing a reasonable tool classification without considering the entire tool chain: The above-mentioned issue also leads to the problem that proper tool classifications and qualifications are difficult to achieve without considering the entire tool chain. This raises questions of how feasible the qualification of individual tools is in practice.

Overlap between STQP and STCA: There seems to be overlap or at least a strong interdependence between parts of the STQP and the STCA. The STCA seems to be

prerequisite to complete the sections of the STQP that are concerned with the tool qualification methods and the required Tool Confidence Level. On the other hand, the documentation of the tool use cases and the means for detecting malfunctions or erroneous output seem to be input for the tool classification. In the final standard, the author would like to see a clearer separation of concerns between the two documents and a suggested order in which they should be created.

No established tool qualification best practices: The reported ISO 26262 tool qualification activities were carried out at a time where no reference qualifications were available. The author believes, that the definition of suitable verification and validation measures to be used in combination with a qualified tool is a means to provide practitioners with the necessary guidance to successfully utilize these tools in projects that need to comply with the requirements of ISO 26262. Until common best practices have been established, the chosen qualification approach could be used as a reference for other tool qualifications.

III. SUMMARY AND CONCLUSIONS

With the advent of ISO 26262 automotive practitioners need to figure out how implement the tool qualification requirements of this standard in practice.

The paper reported on experiences with one of the first (if not the first) ISO/DIS 26262 tool qualifications of commercially available production code generation and verification tools.

The successful ISO/DIS 26262-8 tool qualifications of MathWorks Real-Time Workshop Embedded Coder, PolySpace Client for C/C++ and PolySpace Server for C/C++ demonstrated the feasibility of applying this functional safety standard to both development and verification tools.

The definition of suitable verification and validation measures to be used in combination with the qualified tools is believed to provide practitioners with the necessary guidance to apply Model-Based Design and advanced code verification tools in projects that need to comply with the requirements of ISO/DIS 26262.

REFERENCES

- [CMR10] M. Conrad, P. Munier, F. Rauch: Qualifying Software Tools According to ISO 26262. Proc. of MBEES 2010, Dagstuhl, Germany, 2010
- [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
- [CS09] M. Conrad, G. Sandmann: A Verification and Validation Workflow for IEC 61508 Applications. SAE Techn. Paper #2009-01-0271, SAE World Congress 2009
- [CSM10] M. Conrad, J. Sauler, P. Munier: Experience Report: Two-stage Qualification of Software Tools. Proc. 2. EUROFORUM ISO 26262 Conference, Stuttgart, Germany, 2010
- [DO-178B] RTCA/DO-178B. Software Considerations in Airborne Systems and Equipment Certification. 1992
- [IEC 61508] IEC 61508-3:1998. Int. Standard Functional safety of electrical/ electronic/ programmable electronic safety-related systems. 1998-2000.

- [ISO/DIS 26262] ISO/DIS 26262:2009. Draft International Standard Road vehicles — Functional safety. 2009.
- [ISO/DIS 26262-1] ISO/DIS 26262-1:2009. Draft International Standard Road vehicles — Functional safety - Part 1: Vocabulary. 2009.
- [ISO/DIS 26262-8] ISO/DIS 26262-8:2009. Draft International Standard Road vehicles — Functional safety - Part 8: Supporting processes. 2009.
- [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
- [MBD] Model-Based Design web page. The MathWorks Inc., www.mathworks.com/model-based-design
- [KZ09] A. Kornecki, J. Zalewski: Certification of software for real-time safety-critical systems: state of the art. Innovations Syst Softw Eng (2009) 5:149–161
- [RTW-EC] Real-Time Workshop® Embedded Coder™ product page. The MathWorks Inc., www.mathworks.com/products/rtwembedded
- [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
- [Sau10] J. Sauler: Alle Fakten zur neuen Sicherheits-Norm für die Autoindustrie ISO 26262 (In German), Interview, Elektronik Praxis, 3.2.2010
- [TMW08] 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
- [TMW09] 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