Functional Safety Implications for Development Infrastructures

Dr. Erwin Petry
KUGLER MAAG CIE GmbH
Leibnizstraße 11 · 70806 Kornwestheim · Germany
Mobile: +49 173 67 87 337 · Tel: +49 7154-1796-222 · Fax: +49 7154-1796-480
Email: erwin.petry@kuglermaag.com · Internet: www.kuglermaag.com

June 24, 2010
Eclipse Embedded Day 2010, Stuttgart, Germany, June 24, 2010

Contents

• What is functional safety?
• What does the upcoming standard on functional safety in the automotive domain ISO/DIS 26262 require regarding software tools?
• What is the software tool qualification method according to ISO/DIS 26262?
• How is software tool qualification performed by manufacturers and by users?
Functional Safety
Definitions ISO/DIS 26262-1

Safety: Absence of unreasonable risk

Risk: Combination of the probability of occurrence of harm and the severity of that harm

Functional safety: Absence of unreasonable risk due to hazards caused by malfunctioning behavior of E/E systems

E/E system: System that consists of electrical and/or electronic elements, including programmable electronic elements

Functional Safety Standards
Common principles

• Functional safety is an attribute of products/systems
• Standards describe the state-of-the-art technology for achieving functional safety of products: E.g. IEC 61508, ISO/DIS 26262
• Such standards are used as guidance for product developments, in product liability lawsuits, for marketing purposes, …
• Contain process- and product-related requirements and recommendations
• Process-related requirements include requirements related to development and test environments as well as related to software tools used
• Argument: Risk reduction through a controlled development process and its tools
• Standards require/recommend methods to be applied. Examples: Semi-formal notations for software design, static code analysis, statement coverage
• In practice most methods need to be implemented using software tools
• No specifically named software environments or tools from a specific manufacturer recommended
ISO/DIS 26262 Tool Requirements

Miscellaneous requirements

- Adequate resources shall be provided, incl. tools, databases, templates
- Software tools for software development shall be selected and their use planned; including guidelines for their application
- Shall be consistent across the software lifecycle and compatible with system and hardware lifecycles
- In case of modifications to previously suited software tools: Impact analysis
- Requirements/recommendations for software implementation: Related to dynamic objects or variables, related to unconditional jumps, … -> Need to be supported by the environment/language and/or tools
- Examples but no requirement for integration and test environments: MiL, SiL, PiL, HiL, vehicle
- No specific requirement for certified or proven in use compiler
- No distinction between development and test tools
- **Software tools used must be suited for purpose.** Evidence by applying ISO/DIS 26262-8, clause 11, Qualification of software tools

ISO/DIS 26262 Overview

1. Vocabulary

2. Management of functional safety
   - 2.5 Overall safety management
   - 2.6 Safety management during item development
   - 2.7 Safety management after release for production

3. Concept phase
   - 3.5 Item definition
   - 3.6 Initiation of the safety lifecycle
   - 3.7 Hazard analysis and risk assessment
   - 3.8 Functional safety concept

4. Product development: system level
   - 4.5 Initiation of product development at the system level
   - 4.6 Specification of the technical safety requirements
   - 4.7 System design
   - 4.8 Environmental safety requirements
   - 4.9 Evaluation of violation of the safety goals
   - 4.10 HW integration & testing

5. Product development: hardware level
   - 5.5 Initiation of product development at the hardware level
   - 5.6 Specification of hardware safety requirements
   - 5.7 Hardware design
   - 5.8 Hardware arch. metrics
   - 5.9 Evaluation of violation of the safety goals
   - 5.10 HW integration & testing

6. Product development: software level
   - 6.5 Initiation of product development at the software level
   - 6.6 Spec. of SW safety requirements
   - 6.7 Software architectural design
   - 6.8 SW unit design/implementation
   - 6.9 Software unit testing
   - 6.10 Software integration & testing
   - 6.11 Verification of software safety requirements

7. Production and operation
   - 7.5 Production
   - 7.6 Operation, service (maintenance and repair), and decommissioning

8. Supporting processes
   - 8.5 Interfaces within distributed development s
   - 8.6 Specification & management of safety requirements
   - 8.7 Configuration management
   - 8.8 Change management
   - 8.9 Verification
   - 8.10 Documentation
   - 8.11 Qualification of SW tools
   - 8.12 Qualif. of SW components
   - 8.13 Qualif. Of HW components
   - 8.14 Proven in use argument

9. ASIL-oriented and safety-oriented analyses
   - 9.5 Requirements decomposition with respect to ASIL tailoring
   - 9.6 Criteria for coexistence of elements
   - 9.7 Analysis of dependent failures
   - 9.8 Safety analyses

10. Guideline (informative)

Tool Qualification
**ISO/DIS 26262-8 Supporting Processes**

Qualification of software tools

- **Software tools** used in the lifecycle of safety-related items or elements must be **suited** for its use.
- Suitability must be analyzed and evidence must be provided.
  - Perform an analysis of the use case in the workflow: Does the use of the tool have the potential to violate a safety requirement?
  - Judgement whether an error in the tool can still be detected so that nevertheless no safety requirement will be violated
- In case there is a hazard by the tool a **qualification of the tool** must be performed resp. evidence of qualification must be given.
- ISO/DIS 26262 defines different methods for the qualification depending on the hazard and the ASIL of the item or the element.
- TCL (Tool Confidence Level) is not a required attribute of a tool but an attribute of the use of a tool in the safety lifecycle.
- Tool qualification can largely be performed before item development, assuming a TCL.

**ISO/DIS 26262-8 Supporting Processes**

Qualification of software tools

- Classification of software tools according to two attributes:
  - **TI** (**Tool Impact**): Probability of violating a safety requirement by an error of the tool
  - **TD** (**Tool error Detection**): Probability of preventing or detecting a malfunction or erroneous output of the tool
- Both attributes TI and TD are determined and translated into a required **Tool Confidence Level (TCL)**.
ISO/DIS 26262-8 Supporting Processes
Qualification of software tools

- **TI (Tool Impact):** Probability of violating a safety requirement by an error of the tool
  - **TI0** shall be chosen when there is an argument that there is no such possibility
  - **TI1** shall be chosen in all other cases
- **TD (Tool error Detection):** Probability of preventing or detecting a malfunction or erroneous output of the tool
  - **TD1** shall be chosen if there is a high degree of confidence that a malfunction or an erroneous output from the software tool will be prevented or detected
  - **TD2** shall be chosen if there is a medium degree of confidence that a malfunction or an erroneous output from the software tool will be prevented or detected
  - **TD3** shall be chosen if there is a low degree of confidence that a malfunction or an erroneous output from the software tool will be prevented or detected
  - **TD4** shall be chosen in all other cases

---

**ISO/DIS 26262-8 Supporting Processes**
Qualification of software tools

- **TCL1:** Tools that cannot violate a safety requirement (TI0) or whose malfunction can be prevented or detected with a high degree of confidence (TI1 und TD1) need the lowest level of confidence TCL1. No qualification measures necessary
- **TCL2** is for tools with TI1 and TD2.
- **TCL3** is for tools with TI1 and TD3.
- **TCL4** is the highest level of confidence needed. It is for such tools that have the potential to violate a safety requirement and a low degree of confidence to detect an erroneous output by other means.
Tool Confidence Levels
Examples of what could typically be expected

- Compilers and code generators: TCL2 or TCL3
  - Heavily depends on the quality of subsequent tests. Even TCL4 possible.
- Simulation and analysis tools: TCL1 or TCL2
- Test automation: TCL2
- Configuration management system for the product itself: TCL2
- Most other tools: TCL1

- In a well organized workflow we would expect no tool to be TCL4

#### ISO/DIS 26262-8 Supporting Processes
Qualification of software tools

<table>
<thead>
<tr>
<th>TCL4</th>
<th>Methods</th>
<th>ASIL</th>
</tr>
</thead>
<tbody>
<tr>
<td>1a</td>
<td>Increased confidence from use</td>
<td>++</td>
</tr>
<tr>
<td>1b</td>
<td>Evaluation of the development process</td>
<td>++</td>
</tr>
<tr>
<td>1c</td>
<td>Validation of the software tool</td>
<td>+</td>
</tr>
<tr>
<td>1d</td>
<td>Development in compliance with a safety standard²</td>
<td>+</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>TCL3</th>
<th>Methods</th>
<th>ASIL</th>
</tr>
</thead>
<tbody>
<tr>
<td>1a</td>
<td>Increased confidence from use</td>
<td>++</td>
</tr>
<tr>
<td>1b</td>
<td>Evaluation of the development process</td>
<td>++</td>
</tr>
<tr>
<td>1c</td>
<td>Validation of the software tool</td>
<td>+</td>
</tr>
<tr>
<td>1d</td>
<td>Development in compliance with a safety standard²</td>
<td>+</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>TCL2</th>
<th>Methods</th>
<th>ASIL</th>
</tr>
</thead>
<tbody>
<tr>
<td>1a</td>
<td>Increased confidence from use</td>
<td>++</td>
</tr>
<tr>
<td>1b</td>
<td>Evaluation of the development process</td>
<td>++</td>
</tr>
<tr>
<td>1c</td>
<td>Validation of the software tool</td>
<td>+</td>
</tr>
<tr>
<td>1d</td>
<td>Development in compliance with a safety standard²</td>
<td>+</td>
</tr>
</tbody>
</table>
Software Tool Qualification Methods

1a Increased confidence from use (1/2)

- **Used** previously for the same purpose with comparable use-cases and with a comparable determined environment and with similar functional constraints
- **Specification** of the software tool unchanged
- **No violation of a safety requirement** allocated to a previously developed safety-related item or element occurred as a consequence of malfunctions or erroneous outputs of this software tool
  - To create such evidence, data about the occurrence of malfunctions or of erroneous output of the software tool, observed or detected during previous developments shall be accumulated in a systematic way and made available.

Software Tool Qualification Methods

1a Increased confidence from use (2/2)

- The requirements of the *proven in use* argument from clause 14 are not applicable.
  - I.e. e.g. no requirement for at least one year operating time and no limit for incident rate
- Analyze previous use:
  - Identify tool and version, details of period of use
  - Documentation of malfunctions
  - Measures taken to deal with known malfunctions, related to identified versions
- Confidence from use argument only valid for the considered **version**
  - May be valid only for a specific variant of use: Was the compiler used with or without code optimization option?
Software Tool Qualification Methods

1b Evaluation of the development process

- Development process shall comply with an appropriate standard
- Provide evidence by an assessment
  - E.g. Automotive SPICE, CMMI, ISO 15504

Software Tool Qualification Methods

1c Validation of the software tool

- Validation measures shall demonstrate that the software tool fulfils its specified requirements
  - E.g. by using a test suite with a determined functional and structural coverage
- Analyze eventually occurring erroneous outputs, including analysis of possible consequences and measures for avoidance and detection
- The reaction of the software tool to anomalous operating conditions shall be examined
  - E.g. use of prohibited use of configuration settings
- Examine robustness
- Validation can largely be automated using a validation suite
  - Ensure correctness and robustness of such functionality that will actually be used for the development of safety-related elements
Software Tool Qualification Methods
1d Development in compliance with a safety standard

- 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.

ISO/DIS 26262-8 Supporting Processes
Qualification of software tools

- Example: Workflow and TD classification

  ![Diagram](image)

- What is the probability that a fault in the compiler is detected by subsequent tests?
ISO/DIS 26262-8 Supporting Processes
Qualification of software tools

• **Procedure** for qualification
  • Precise identification of the candidate for qualification (version, parameters, …)
  • Analyze intended use of the tool in the lifecycle. Determine TI.
  • Estimate probability of the tool error detection. Determine TD.
  • Determine TCL
  • Determine maximum ASIL of the safety function or of the item
  • Determine method(s) for qualification (tables 2 through 4)
  • Apply method(s) for qualification („qualify“). Provide a report.
  • Confirm (review) the qualification

• **Output work products** of qualification
  • Qualification plan
  • Tool documentation
  • Tool classification analysis
  • Qualification report

---

ISO/DIS 26262-8 Supporting Processes
Qualification of software tools

• **Confirmation review** of the qualification recommended for ASIL B and required from ASIL C upwards (ISO/DIS 26262-2, 6.4.6.2, table 1)

<table>
<thead>
<tr>
<th>Confirmation review of the qualification of software tools (see ISO/26262-8, Clause 11)</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>- independent from the person performing the qualification of the software tool</td>
<td></td>
<td>10</td>
<td>11</td>
<td>11</td>
</tr>
</tbody>
</table>

• I0 = should be performed (recommendation)
• I1 = shall be performed (requirement)
• No requirement for independence of the reviewer
• Self qualification possible
Software Tool Qualification in Practice
By manufacturers (example)

1. Tool development including tool test and validation by the manufacturer (methods 1d and 1c)
2. Tool maintenance: Documentation of tool usage, bug reports, bug analysis, bug fixing and user information by the manufacturer (method 1a)
3. Evaluation of the development process and of the maintenance process for the tool by an independent inspection authority (KUGLER MAAG CIE, …) with qualification report and tool certificate (method 1b)
4. Review of the qualification report by different persons of the manufacturer and the inspection authority (Confirmation review according to part 2, 6.4.6.2)

- Qualification is valid for a specific version of the tool
- A TCL, an ASIL, use cases and environments of usage are assumed
- Validity of the manufacturer’s qualification needs to be evaluated for the particular use by the using organization/project

Software Tool Qualification in Practice
By tool users (example)

1. Assumed method 1d “development (of the tool) in compliance with a safety standard” is not possible
2. Evaluation of the tool’s development process is also nearly impossible (method 1b)
3. Candidates for qualification are only such tools/development environments for which continuous tool maintenance is effective
4. Systematically collect information about the tool usage (in-house and external)
   - Includes information about violations of safety requirements as a consequence of malfunctions of the tool
5. Self qualification or commissioning of qualification using method “increased confidence from use” for ASIL A, B
6. Self qualification or commissioning of qualification using methods “increased confidence from use” and “validation of the tool” for TCL3/ASIL D, resp. TCL4/ASIL C, D
7. Review of the qualification report by different persons of the tool user’s organization and/or the inspection authority (Confirmation review according to part 2, 6.4.6.2)
Software Tool Qualification in Practice

Example

- One such sheet for each software tool in the workflow

### Software Tool Qualification Report

<table>
<thead>
<tr>
<th>Tool name</th>
<th>Tool short description</th>
<th>Tool information (11.4.2.2)</th>
<th>Planning of qualification (11.4.2.1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Yes tool does ...</td>
<td>features, functions, properties, installation process, use manual, environment, behaviour under anomalous operating conditions, known malfunctions (or references)</td>
<td>unique identification, version, configuration, use-cases, environment, maximum ASIL</td>
</tr>
<tr>
<td>Author</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Date</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Tool use-cases (11.4.3.1)</th>
<th>Purpose</th>
<th>Environmental and functional constraints</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tool name</td>
<td></td>
<td>e.g. model, source code, object code, embedded software, executable test</td>
</tr>
<tr>
<td>Tool short description</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Tool impact</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Hazard</th>
<th>Measure indicating error detection</th>
<th>Effectiveness</th>
<th>TD</th>
</tr>
</thead>
<tbody>
<tr>
<td>reference above</td>
<td>What measure could detect a tool error? E.g. review, test</td>
<td>low/medium/high</td>
<td>TD1/TD2/TD3/TD4</td>
</tr>
<tr>
<td>Tool error detection analysis (11.4.3.2.a)</td>
<td>Hazard</td>
<td>Tool impact</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Tool error Detection</th>
<th>low/medium/high</th>
<th>TD1/TD2/TD3/TD4</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Tool confidence level (11.4.3.4)</th>
</tr>
</thead>
<tbody>
<tr>
<td>TCL1/TCL2/TCL3/TCL4</td>
</tr>
</tbody>
</table>

Qualification required?

- yes/no

### Remarks

- There is still very **little experience** with the qualification method according to ISO/DIS 26262
- Tool qualification is possible for an inspection authority without testing the tool itself
  - In case of TCL3/ASIL D or TCL4/ASIL C validation e.g. by the manufacturer
- There is not yet a consolidated opinion about the TCL classification of standard tools
  - A code generator was classified TCL1 because inserted errors would be found by the subsequent workflow. For TCL1 no specific methods for qualification are necessary.
  - Not defined what low, medium and high degree of confidence in tool error detection (TD) means
Summary

• In the automotive application domain the possibility to violate a safety requirement needs to be assessed for all software tools used in the workflow
• Typically, tool qualification needs to be performed for only the main software tools used in the workflow
• Criteria for tool qualification are relatively vague in ISO/DIS 26262
• Tool qualification itself is not difficult
• Tools can be qualified by inspection authorities, manufacturers and users

Thank you for your attention!

Should you have any questions please do not hesitate to contact us ...

KUGLER MAAG CIE GmbH
Leibnizstraße 11
D-70806 Kornwestheim
Internet: www.kuglermaag.com
Tel. Office : +49 7154 - 807 210
Email: Safety@kuglermaag.com

Dr. Erwin Petry
Email: Erwin.Petry@kuglermaag.com
Mobile: +49 (0) 173-678 7337
Our Book about Functional Safety
(in German)

Can be ordered here:
http://www.kuglermaag.de/webshop.html