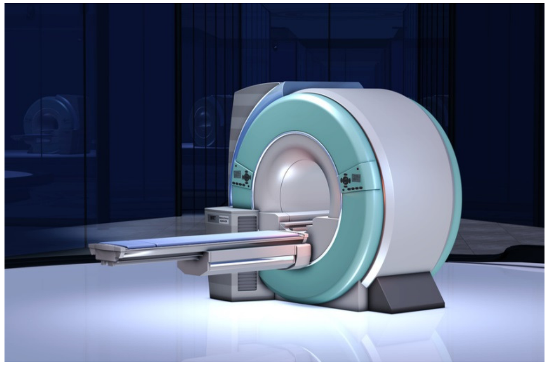
Compiler in sicherheitsrelevanten Projekten
Was bedeutet das für die Testumgebung?

HEICON Global Engineering GmbH
Kreuzweg 22, 88477 Schwendi

Internet: www.heicon-ulm.de
Blog: http://blog.heicon-ulm.de



HEICON

HEICON Global Engineering

HEICON is a specialized engineering company which provides consulting- and development support with a focus on software-based embedded systems.
The efficient implementation of methods and processes is the area of our engagement.

Founding: 2018 Headquarter: Schwendi near Ulm Membership: Bodensee AIRea fascination aerospace ASQF Arbeitskreis Software-Qualität und -Fortbildung e.V.
Employees: 2 Legal form: GmbH

Revenue Distribution:

HEICON
Global Engineering

Tool Qualification – What is it and when do you need it?

Compiler – A definition

Tool Qualification – Requirements in DO178 and ISO26262

Typical tool qualification measures

Minimizing the risk of undetected compiler errors

Conclusion

Tool Qualification – What is it?

A tool qualification is the proof, by using appropriate measures, that a selected tool works correctly. The selected tool replaces a process which is defined in the functional safety standard.

Tool Qualification – When do you need it?

4-Eye-Principle

- Essential process in a functional safety development
- In the tools area, this means, that you can not „just“ assume that a tool works correctly
- A tool qualification is required if you use the tool to automate a process which is defined in a functional safety standard (= 4-Eye-Principle violation)

2 possible failure categories

- **Failure** in a safety critical system are not recognized (Example: Verification tools which checks the coding rules, Unittest tools)
- **Failures** are introduced into the safety critical system (Example: Compiler)

Tool Qualification – What is it and when do you need it?

Compiler – A definition

Tool Qualification – Requirements in DO178 and ISO26262

Typical tool qualification measures

Minimizing the risk of undetected compiler errors

Conclusion

Compiler – A definition

- ✓ The central "tool“ in the software development
- ✓ Forms the link between the human-readable
  high-level source code (e.g., C and C ++) and
  the machine code, interpretable for the hardware processor.

Tool Qualification – What is it and when do you need it?

Compiler – A definition

Tool Qualification – Requirements in DO178 and ISO26262

Typical tool qualification measures

Minimizing the risk of undetected compiler errors

Conclusion

Tool Qualification – Requirements in DO178 and ISO26262

DO 178C and DO 330: Determination of Tool Qualification Level

Compiler

Software Level	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

Criteria 1: A tool whose output is part of the resulting software and thus could insert an error.

Criteria 2: A tool that automates verification process(es) and thus could fail to detect an error, and whose output is used to justify the elimination or reduction of a verification or development process aspect.

Criteria 3: A tool that, within the scope of its intended use, could fail to detect an error.

Tool Qualification – Requirements in DO178 and ISO26262

TQL-5 Tools

Creation of tool operational requirements and proof the fulfillment of it by tests

TQL-1 bis 4 Tools

Proof that the tool was developed according DAL A to D!

Compiler

Tool Qualification – Requirements in DO178 and ISO26262

ISO 26262: Determination of the „tool confidence level“ (TCL)

		Tool error detection		
		TD1	TD2	TD3
Tool impact	TI1	TCL1	TCL1	TCL1
	TI2	TCL1	TCL2	TCL3

Compiler

Table 3, ISO 26262-8

TI1: There is an argument available, that a malfunctioning of the tool does not introduce or fail to detect a failure in a safety critical item

TI2: Shall be selected in all other cases

TD1: High degree of confidence that a malfunction and its corresponding erroneous output will be prevented or detected

TD2: Medium degree of confidence that a malfunction and its corresponding erroneous output will be prevented or detected

TD3: In all other cases

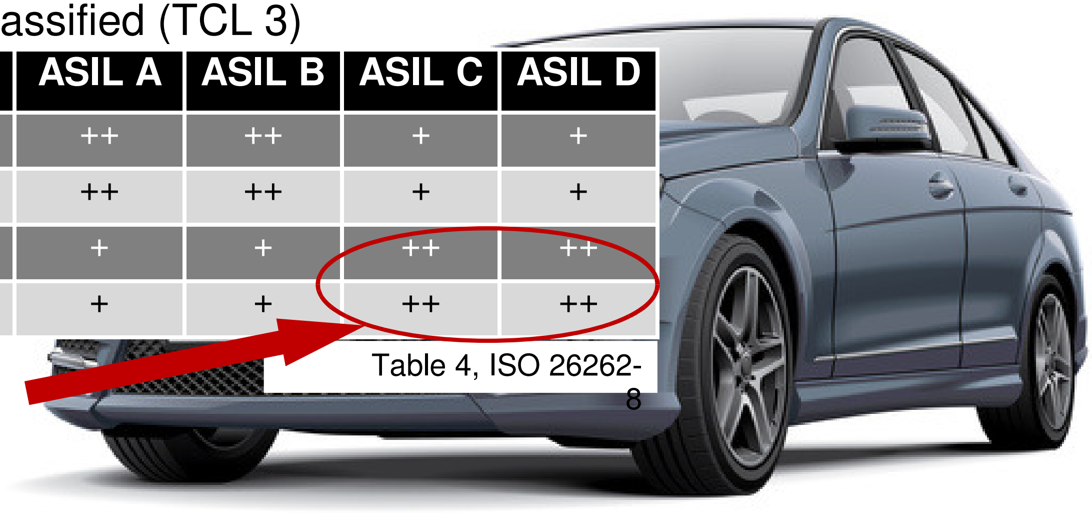
Tool Qualification – Requirements in DO178 and ISO26262

ISO 26262: Qualification of software tools classified (TCL 3)

Methoden	ASIL A	ASIL B	ASIL C	ASIL D
1. Increased confidence from use	++	++	+	+
2. Evaluation of the tool development process	++	++	+	+
3. Validation of the software tool	+	+	++	++
4. Development in accordance with a safety standard	+	+	++	++

Table 4, ISO 26262-8

Compiler



Tool Qualification – What is it and when do you need it?

Compiler – A definition

Tool Qualification – Requirements in DO178 and ISO26262

Typical tool qualification measures

Minimizing the risk of undetected compiler errors

Conclusion

Typical tool qualification measures

Measure
1. Increased confidence from use
2. Evaluation of the tool (compiler) development process
3. Validation of the software tool (compiler)
4. Development of the Tool (compiler) according to an Functional Safety Standard

- ✓ The most powerful method No. 4 is practically not useable for compilers as they are built by commercial manufacturers. The cost for a compiler that is developed according to safety standards is such high that selling it with a profit is hardly possible in the relative small safety market

- ✓ Method No. 3, the validation of the software tool, i.e. compiler faces the challenge to cover all possible cases of high level code constructs and its corresponding translation to the machine code

- ✓ Method No. 2 has not yet been really used in the safety market. The reason might be, that this method can not deliver sufficiently good results for compilers in particular.

- ✓ Method 1 is applicable for compiler verification in all safety critical markets. I am not aware of any safety critical project that would use a compiler that is completely new on the market. In all functional safety projects, compilers are used which are on the market for many years and have already been used in similar projects.

Tool Qualification – What is it and when do you need it?

Compiler – A definition

Tool Qualification – Requirements in DO178 and ISO26262

Typical tool qualification measures

Minimizing the risk of undetected compiler errors

Conclusion

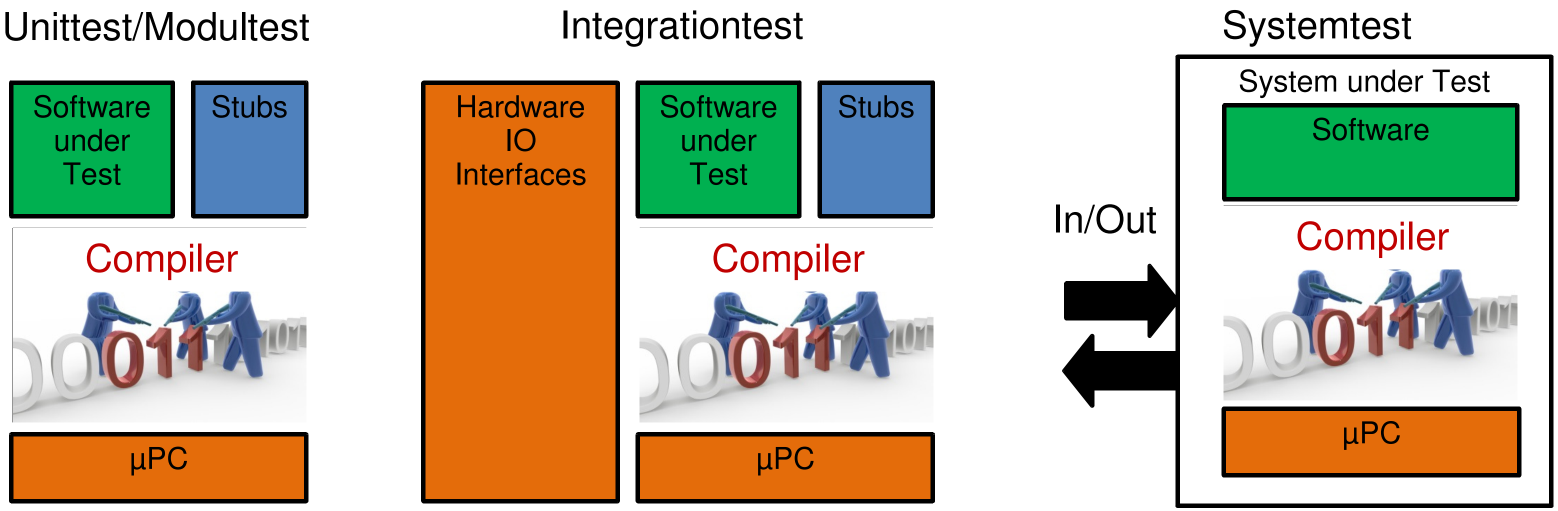
Minimizing the risk of undetected compiler errors

- ✓ According to the DO 178B/C and DO330 a compiler validation is not sufficient in the aerospace

- ✓ So a different approach was developed in the aerospace industry in order to achieve confidence in the use of compiler

- ✓ For safety projects with a high criticality level (SIL3/4 / ASIL C/D) this approach is a very good alternative way also for projects outside the aerospace.

Minimizing the risk of undetected compiler errors



1. For each test (Unit- Integration- or Systemtest), exactly the same compiler settings are used, which are also used for the final operative source code.

1. A structural source code coverage (in case of DAL A MC/DC) of 95% and more has to be achieved by testing

1. 100% Req. based Test Coverage is achieved (All Low-Level Req. / High-Level Req.) are tested with normal and robustness test cases

1. For particularly safety critical projects (DAL A and B), a traceability from the object code to the high-level language source code (C and C ++) is required.

Tool Qualification – What is it and when do you need it?

Compiler – A definition

Tool Qualification – Requirements in DO178 and ISO26262

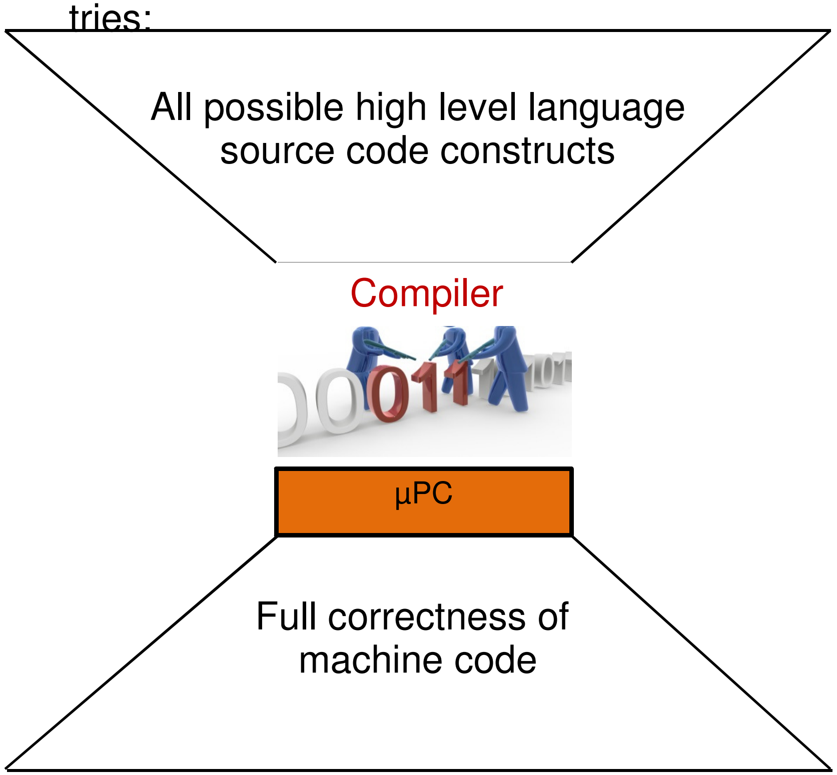
Typical tool qualification measures

Minimizing the risk of undetected compiler errors

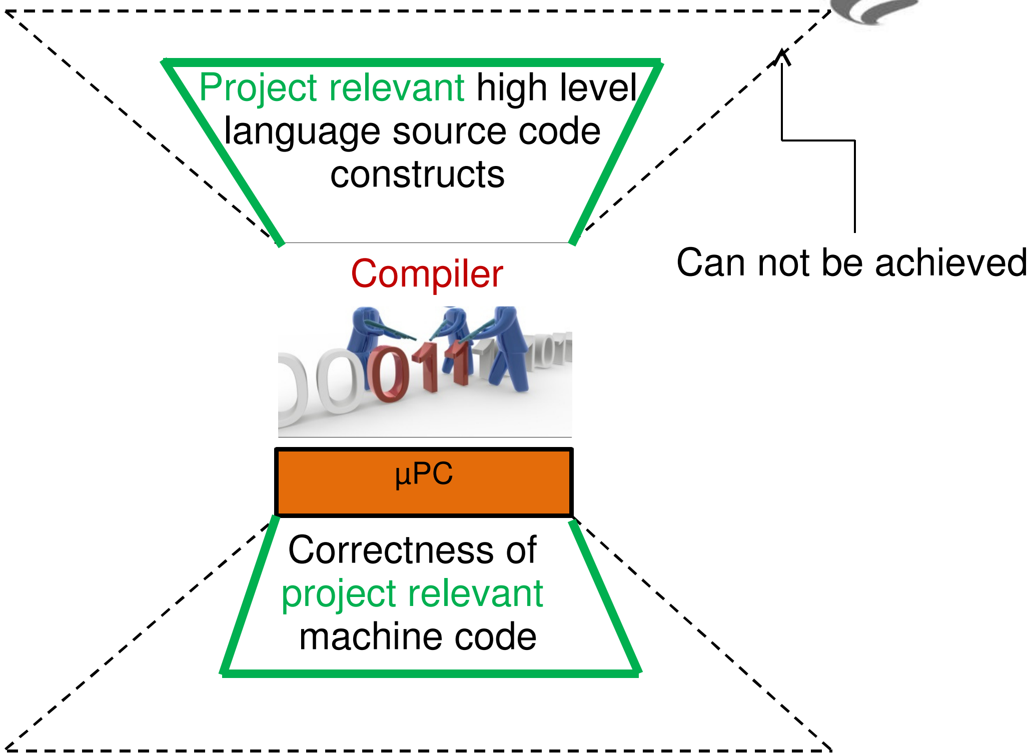
Conclusion

Minimizing the risk of undetected compiler errors

Validation package for Compiler tries:


All possible high level language source code constructs
Compiler
µPC
Full correctness of machine code

Aerospace approach


Project relevant high level language source code constructs
Compiler
µPC
Correctness of project relevant machine code
Can not be achieved

Conclusion

Its impossible to prove the correctness of the full functionality of a compiler, but for a single project, the compiler functionality used is limited and therefore (almost) completely testable. This is exactly what is achieved with applying the recommended measures.

1. In SIL 3 / 4 – ASIL C / D projects you have to achieve anyway 100% structural coverage
2. In SIL 3 / 4 – ASIL C / D projects you have to achieve a full requirements coverage by tests

→ The shown process does not cause significant more test effort
→ It’s the best method to be sure that the compiler does not have any failure – for the source code used in the safety project

Pre-Requisits:

→ Professional Test Environments on all levels
→ Professional test strategy

Conclusion

Further information:

- White paper on compiler qualification: https://solidsands.nl/wp-content/uploads/2018/01/QualificationBenefits.pdf
- Formally verified compilation: https://www.absint.com/compcert/index.htm
- HEICON blog: English: http://blog.heicon-ulm.de/blogs-in-english-2/
  HEICON blog: German: http://blog.heicon-ulm.de/kategorien/

Contact - Publications

Contact:

HEICON – Global Engineering GmbH
Martin Heininger Dipl.-Ing(FH)
Kreuzweg 22
D-88477 Schwendi
Tel.: +49 7353 - 98 17 81
Mobil: +49 176 - 24 73 99 60
martin.heininger@heicon-ulm.de
http://www.heicon-ulm.de

Publications:

Testing power electronics according ISO26262, ATZ 04/15

Monthly: Blog article about Functional Safety Topics: http://blog.heicon-ulm.de