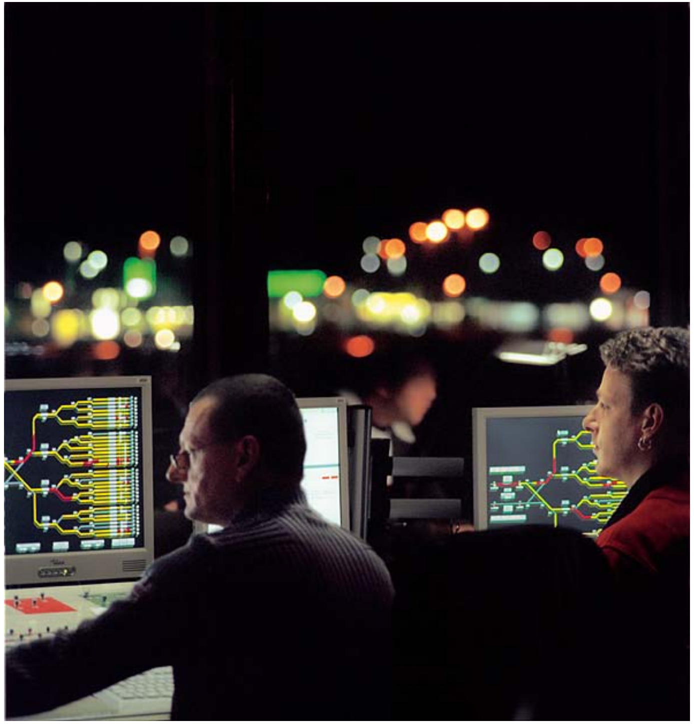
Relation of Model-Based Testing and Safety-Relevant Standards

Dr. Stephan Weißleder

Research Manager Testing

Department Quality of Embedded Systems (QUEST)

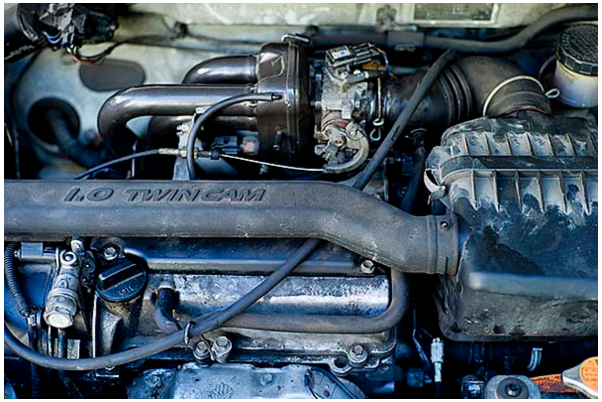
Fraunhofer-Institute FIRST



Fraunhofer
FIRST

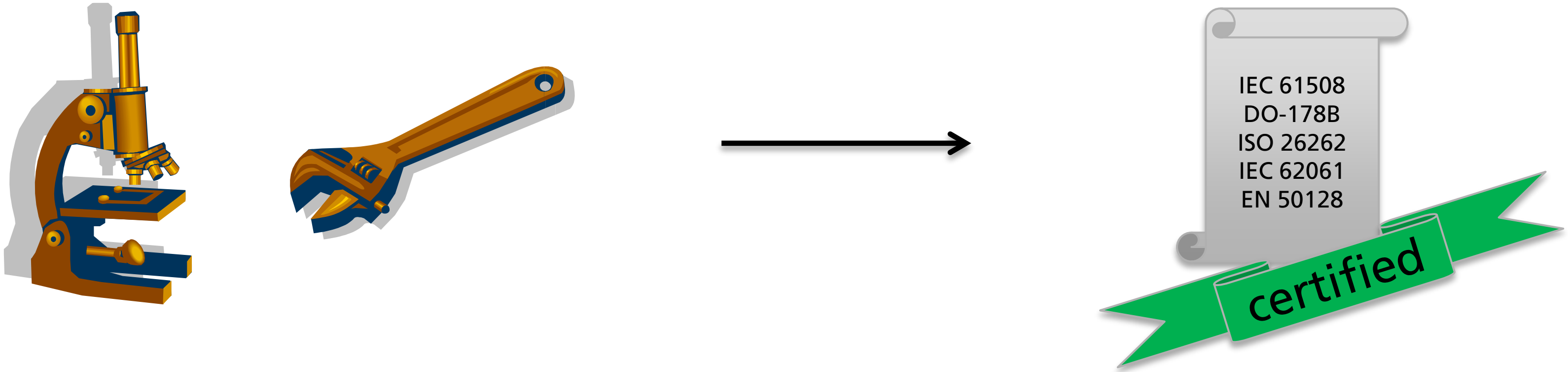
Fraunhofer-Institute FIRST – Department QUEST

Review and Assessment



Testing

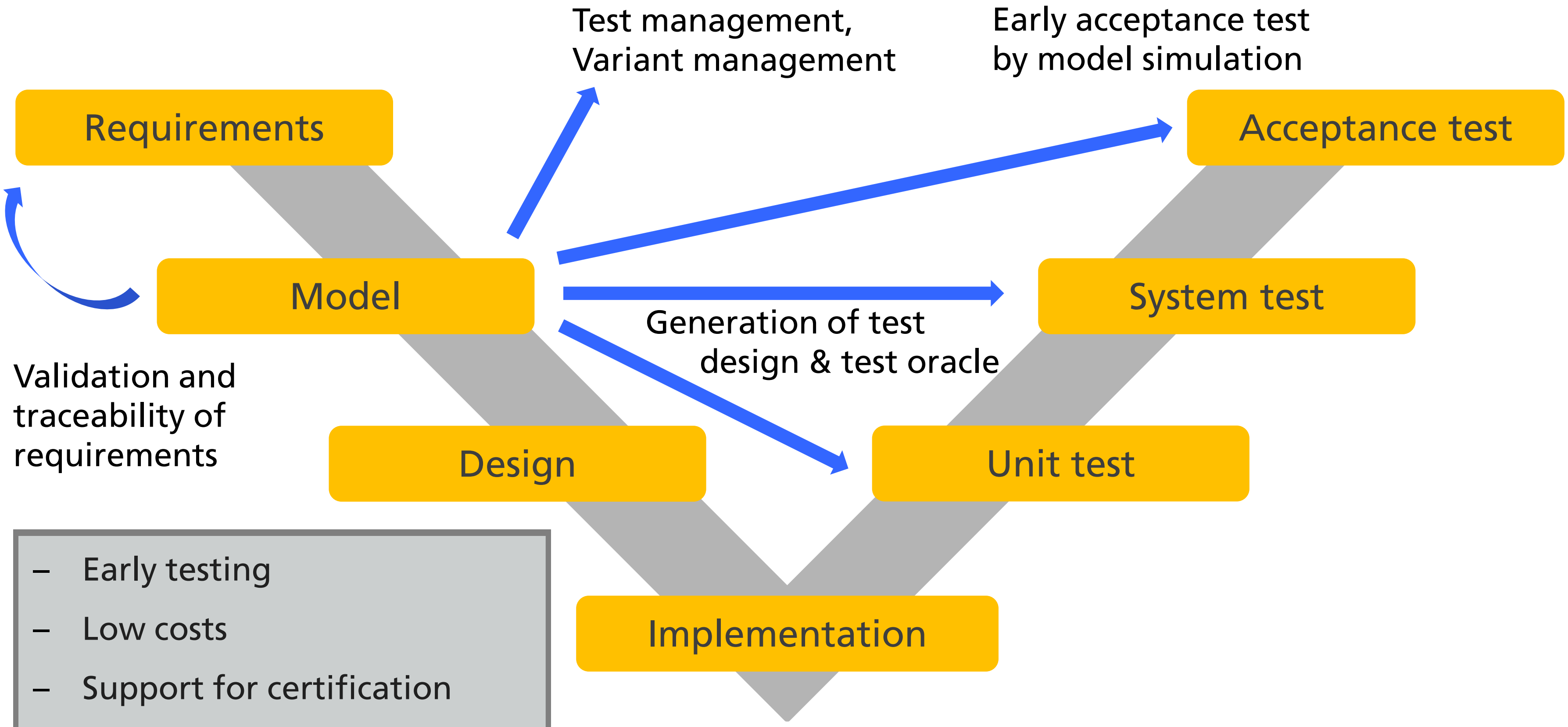
Verification



IEC 61508
DO-178B
ISO 26262
IEC 62061
EN 50128

Model-Based Testing

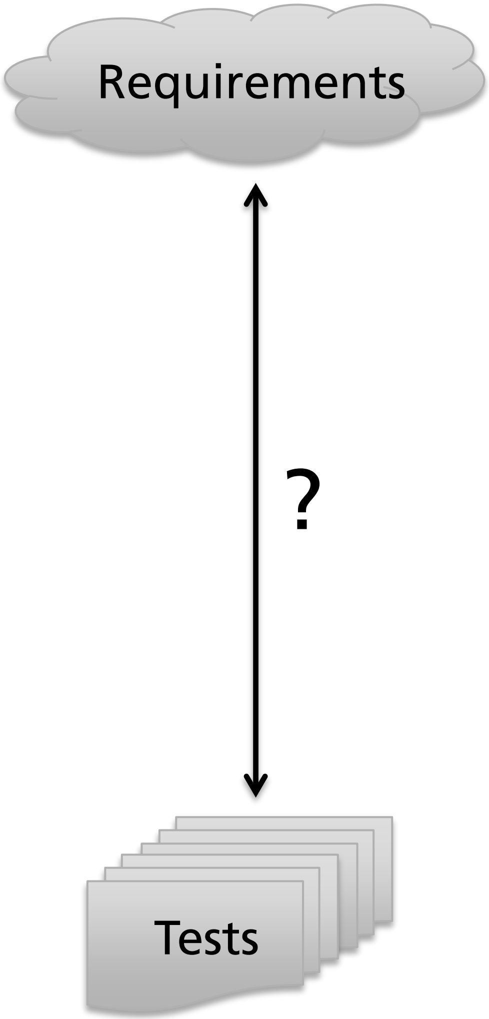
Model-Based Testing



- Early testing
- Low costs
- Support for certification
- Better test management
- High degree of automation
- Better test coverage

How Does MBT Support Safety-Relevant Standards?

How Does MBT Support Safety-Relevant Standards?

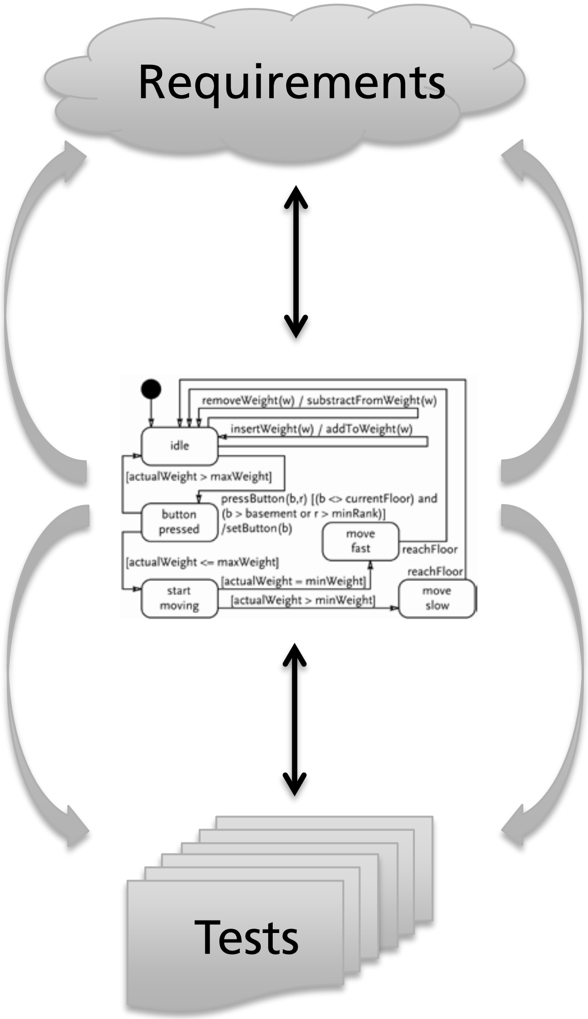


How Does MBT Support Safety-Relevant Standards?

Coverage

Relation of model coverage and requirements coverage?

Measure/Achieve code coverage, More detailed coverage information for system tests?



Requirements

removeWeight(w) / substractFromWeight(w)
insertWeight(w) / addToWeight(w)
idle
[actualWeight > maxWeight]
pressButton(b,r) [(b <> currentFloor) and (b > basement or r > minRank)] /setButton(b)
button pressed
move fast
reachFloor
[actualWeight <= maxWeight]
reachFloor
[actualWeight = minWeight]
start moving
[actualWeight > minWeight]
move slow

Tests

Traceability

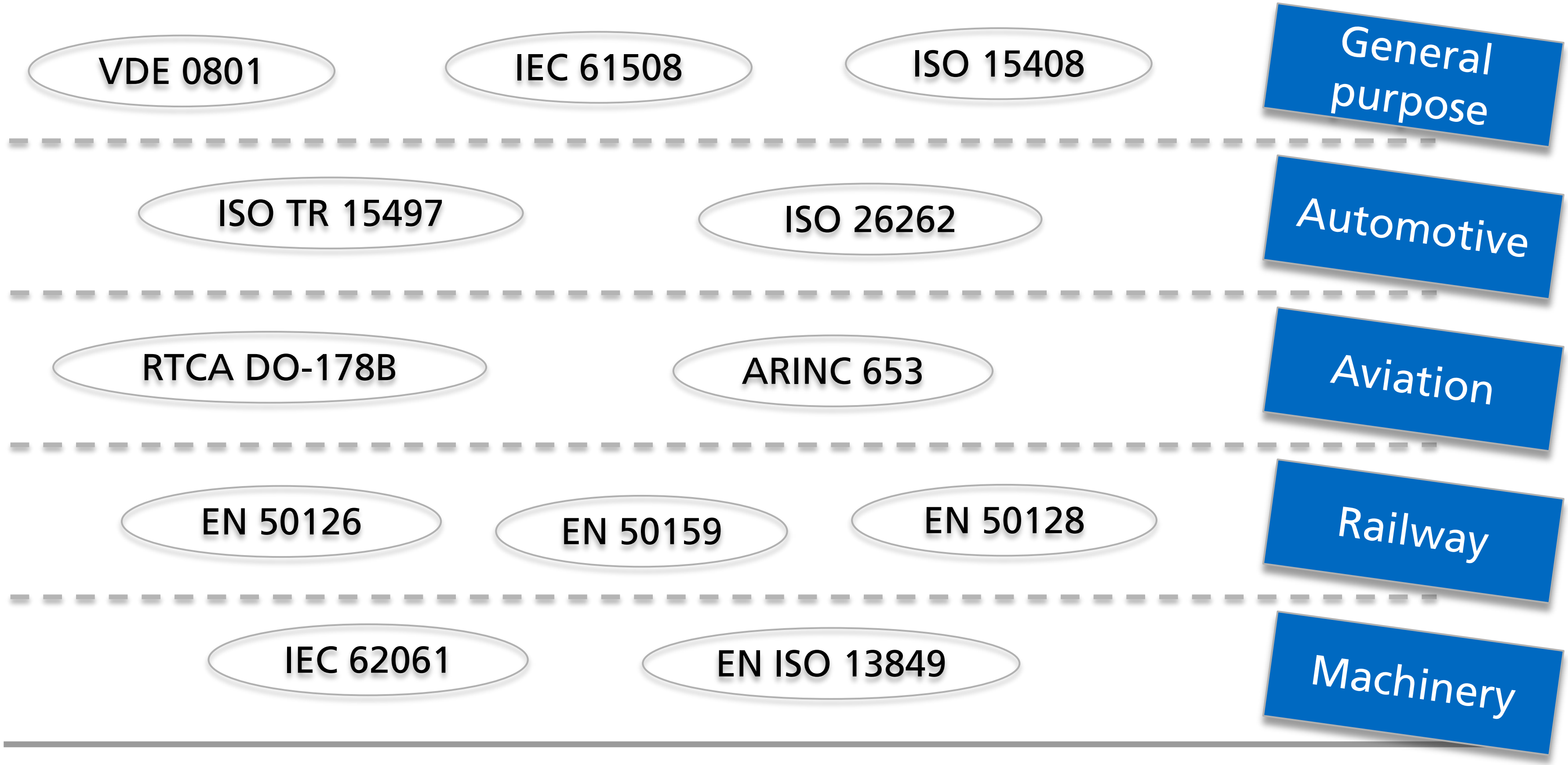
Validation of requirements, Better Traceability?

Higher Efficiency, Automatic Traceability

How do Safety-Relevant Standards Support MBT?

Standards

The good things about standards is there are so many to choose from.



VDE 0801 IEC 61508 ISO 15408 General purpose

ISO TR 15497 ISO 26262 Automotive

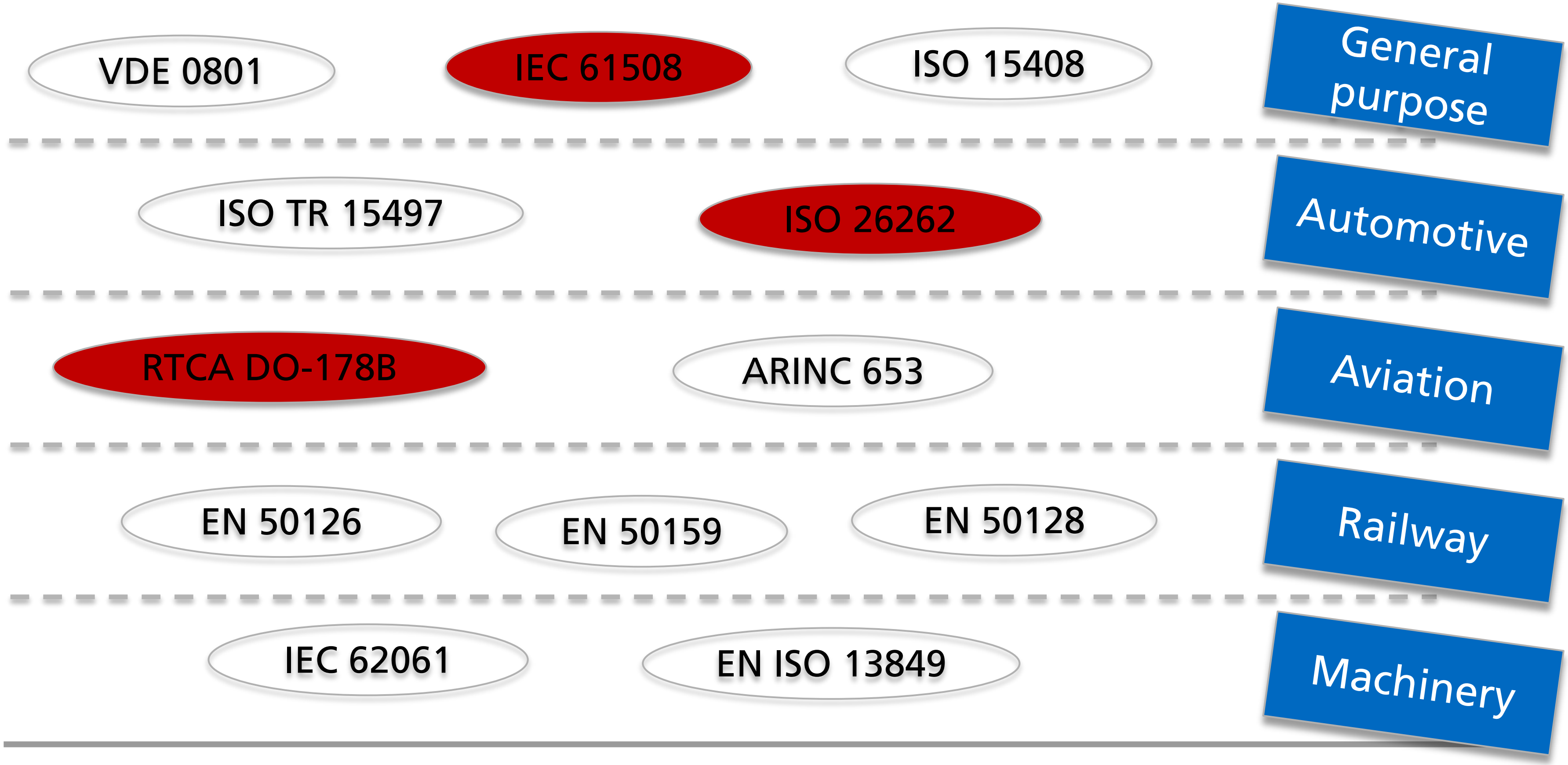
RTCA DO-178B ARINC 653 Aviation

EN 50126 EN 50159 EN 50128 Railway

IEC 62061 EN ISO 13849 Machinery

Standards

The good things about standards is there are so many to choose from.



VDE 0801 · IEC 61508 · ISO 15408 — General purpose

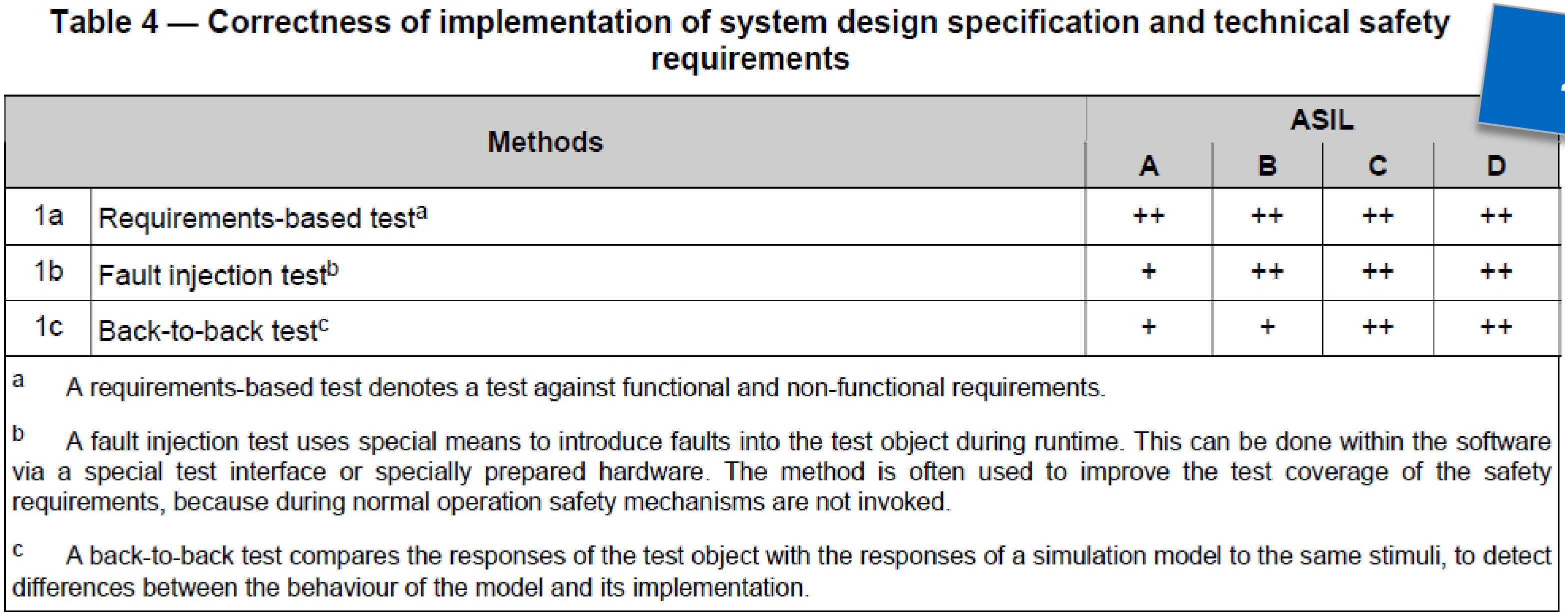
ISO TR 15497 · ISO 26262 — Automotive

RTCA DO-178B · ARINC 653 — Aviation

EN 50126 · EN 50159 · EN 50128 — Railway

IEC 62061 · EN ISO 13849 — Machinery

ISO 26262 (under publication)



26262-4

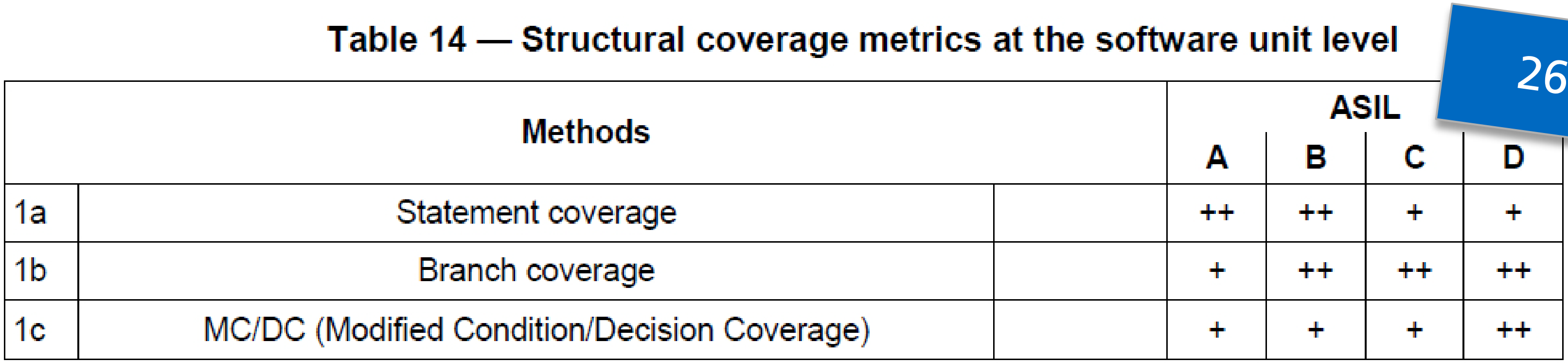
Table 4 — Correctness of implementation of system design specification and technical safety requirements

Methods		ASIL			
		A	B	C	D
1a	Requirements-based test[a]	++	++	++	++
1b	Fault injection test[b]	+	++	++	++
1c	Back-to-back test[c]	+	+	++	++

[a] A requirements-based test denotes a test against functional and non-functional requirements.

[b] A fault injection test uses special means to introduce faults into the test object during runtime. This can be done within the software via a special test interface or specially prepared hardware. The method is often used to improve the test coverage of the safety requirements, because during normal operation safety mechanisms are not invoked.

[c] A back-to-back test compares the responses of the test object with the responses of a simulation model to the same stimuli, to detect differences between the behaviour of the model and its implementation.



26262-6

Table 14 — Structural coverage metrics at the software unit level

Methods			ASIL			
			A	B	C	D
1a	Statement coverage		++	++	+	+
1b	Branch coverage		+	++	++	++
1c	MC/DC (Modified Condition/Decision Coverage)		+	+	+	++

ISO 26262 (under publication)

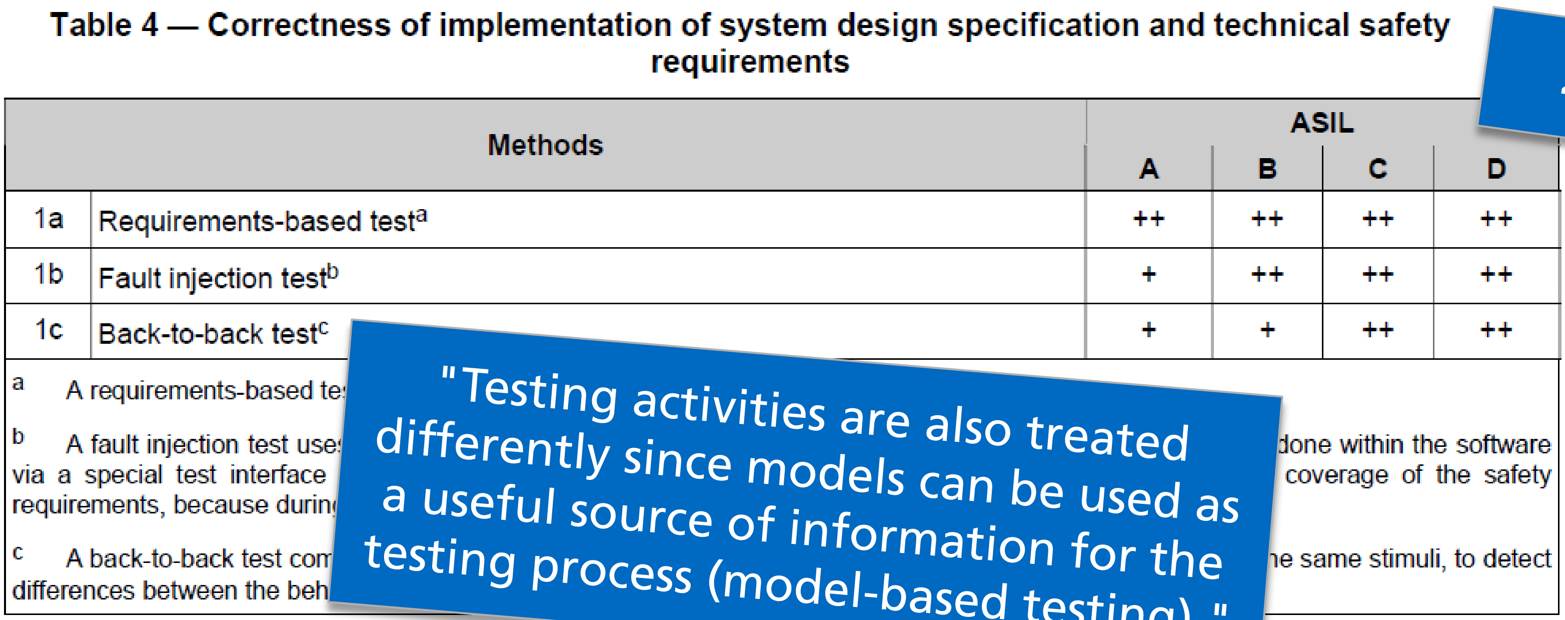


Table 4 — Correctness of implementation of system design specification and technical safety requirements

	Methods	ASIL A	ASIL B	ASIL C	ASIL D
1a	Requirements-based test[a]	++	++	++	++
1b	Fault injection test[b]	+	++	++	++
1c	Back-to-back test[c]	+	+	++	++

[a] A requirements-based te

[b] A fault injection test use done within the software via a special test interface coverage of the safety requirements, because durin

[c] A back-to-back test com he same stimuli, to detect differences between the beh

26262-4

"Testing activities are also treated differently since models can be used as a useful source of information for the testing process (model-based testing)."

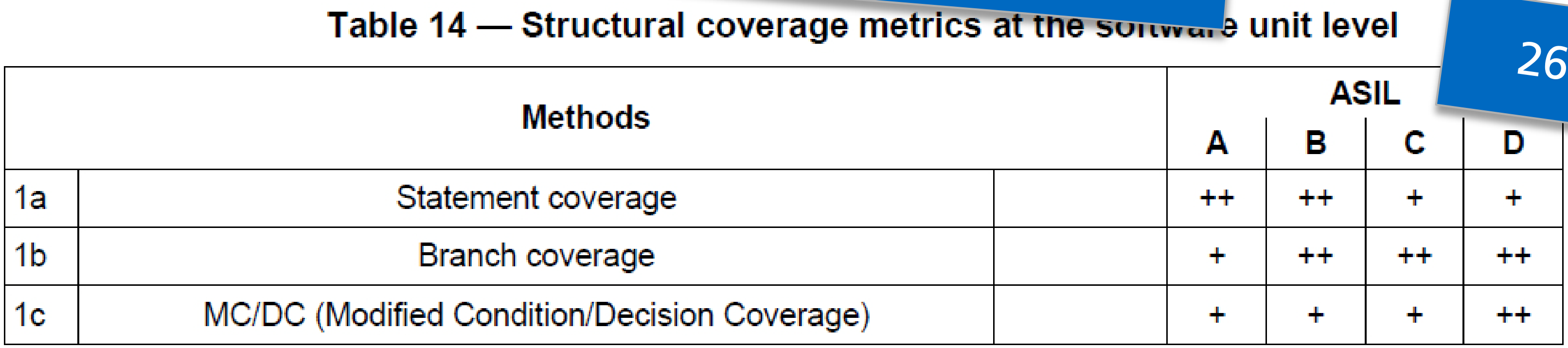


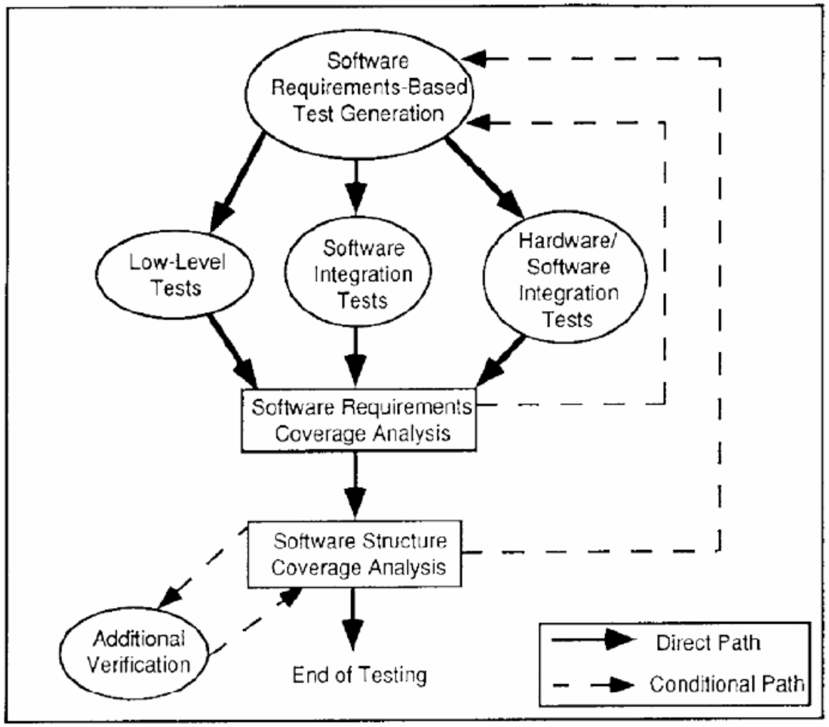
Table 14 — Structural coverage metrics at the software unit level

	Methods		ASIL A	ASIL B	ASIL C	ASIL D
1a	Statement coverage		++	++	+	+
1b	Branch coverage		+	++	++	++
1c	MC/DC (Modified Condition/Decision Coverage)		+	+	+	++

26262-6

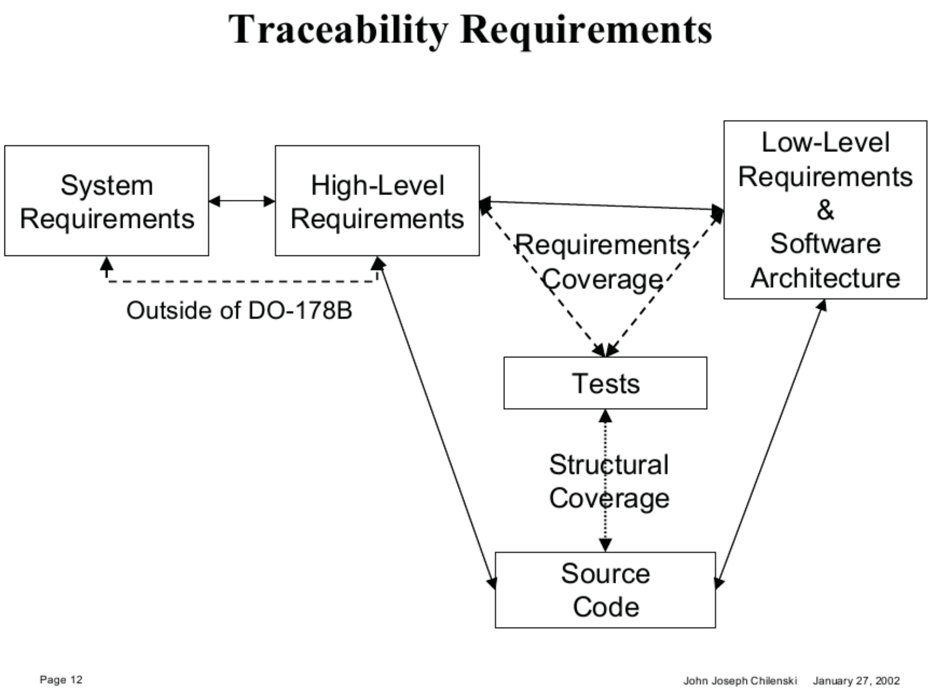
RTCA DO-178B – 1992

DAL E	DAL D	DAL C	DAL B	DAL A
No effect	Minor	Major	Hazardous	Catastrophic



Software Requirements-Based Test Generation
Low-Level Tests
Software Integration Tests
Hardware/ Software Integration Tests
Software Requirements Coverage Analysis
Software Structure Coverage Analysis
Additional Verification
End of Testing
Direct Path
Conditional Path

(Source: DO-178B Standard)



(Source: John Joseph Chilenski)

RTCA DO-178B – 1992

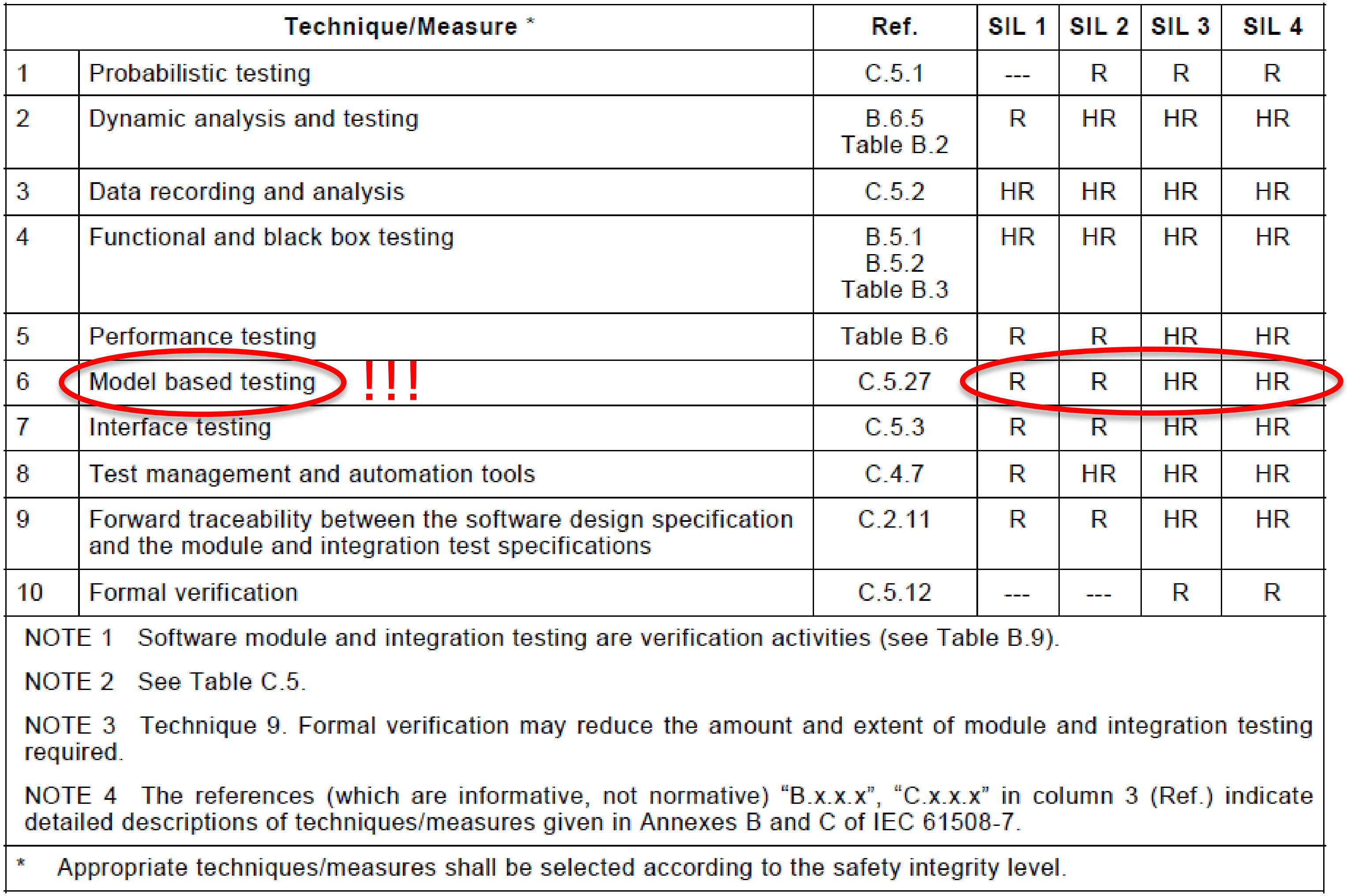
Objective	DAL			
	A	B	C	D
Test coverage of high-level requirements is achieved.	X	X	X	X
Test coverage of low-level requirements is achieved.	X	X	X	
Test coverage of software structure is achieved. (MC/DC)	X			
Test coverage of software structure is achieved. (Decision Coverage)	X	X		
Test coverage of software structure is achieved. (Statement Coverage)	X	X	X	
Test coverage of software structure is achieved. (Data coupling / control coupling)	X	X	X	

RTCA DO-178B – 1992

Objective	DAL			
	A	B	C	D
Test coverage of high-level requirements is achieved.	X	X	X	X
Test coverage of l[illegible]el requirements is achieved.	X	X	X	
Test coverage of s[illegible] (MC/DC)	[illegible]	[illegible]		
Test coverage of software structure is achieved. (Decision Coverage)	X	X		
Test coverage of software structure is achieved. (Statement Coverage)	X	X	X	
Test coverage of software structure is achieved. (Data coupling / control coupling)	X	X	X	

No support for model-based testing.
Things are getting better in DO-178C ?

IEC 61508 – 2010



Technique/Measure *		Ref.	SIL 1	SIL 2	SIL 3	SIL 4
1	Probabilistic testing	C.5.1	---	R	R	R
2	Dynamic analysis and testing	B.6.5 Table B.2	R	HR	HR	HR
3	Data recording and analysis	C.5.2	HR	HR	HR	HR
4	Functional and black box testing	B.5.1 B.5.2 Table B.3	HR	HR	HR	HR
5	Performance testing	Table B.6	R	R	HR	HR
6	Model based testing !!!	C.5.27	R	R	HR	HR
7	Interface testing	C.5.3	R	R	HR	HR
8	Test management and automation tools	C.4.7	R	HR	HR	HR
9	Forward traceability between the software design specification and the module and integration test specifications	C.2.11	R	R	HR	HR
10	Formal verification	C.5.12	---	---	R	R

NOTE 1 Software module and integration testing are verification activities (see Table B.9).

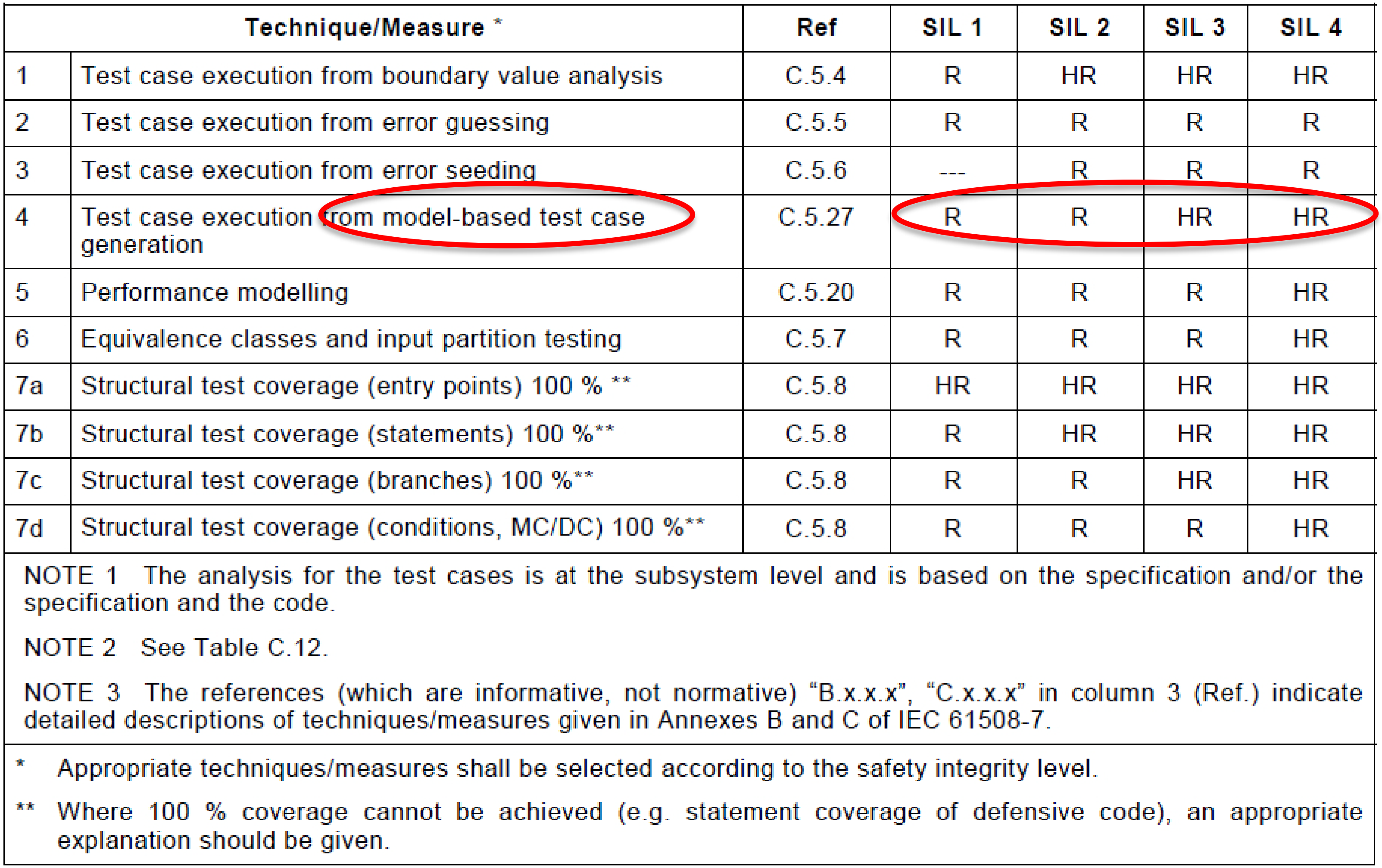
NOTE 2 See Table C.5.

NOTE 3 Technique 9. Formal verification may reduce the amount and extent of module and integration testing required.

NOTE 4 The references (which are informative, not normative) "B.x.x.x", "C.x.x.x" in column 3 (Ref.) indicate detailed descriptions of techniques/measures given in Annexes B and C of IEC 61508-7.

* Appropriate techniques/measures shall be selected according to the safety integrity level.

IEC 61508 – 2010



Technique/Measure *		Ref	SIL 1	SIL 2	SIL 3	SIL 4
1	Test case execution from boundary value analysis	C.5.4	R	HR	HR	HR
2	Test case execution from error guessing	C.5.5	R	R	R	R
3	Test case execution from error seeding	C.5.6	---	R	R	R
4	Test case execution from model-based test case generation	C.5.27	R	R	HR	HR
5	Performance modelling	C.5.20	R	R	R	HR
6	Equivalence classes and input partition testing	C.5.7	R	R	R	HR
7a	Structural test coverage (entry points) 100 % **	C.5.8	HR	HR	HR	HR
7b	Structural test coverage (statements) 100 %**	C.5.8	R	HR	HR	HR
7c	Structural test coverage (branches) 100 %**	C.5.8	R	R	HR	HR
7d	Structural test coverage (conditions, MC/DC) 100 %**	C.5.8	R	R	R	HR

NOTE 1 The analysis for the test cases is at the subsystem level and is based on the specification and/or the specification and the code.

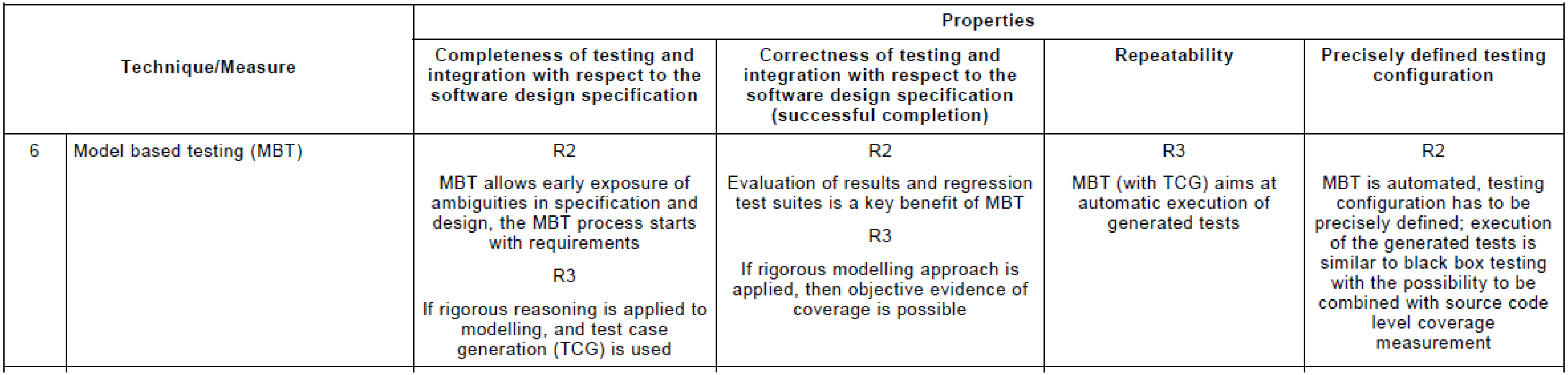
NOTE 2 See Table C.12.

NOTE 3 The references (which are informative, not normative) "B.x.x.x", "C.x.x.x" in column 3 (Ref.) indicate detailed descriptions of techniques/measures given in Annexes B and C of IEC 61508-7.

* Appropriate techniques/measures shall be selected according to the safety integrity level.

** Where 100 % coverage cannot be achieved (e.g. statement coverage of defensive code), an appropriate explanation should be given.

IEC 61508 – 2010

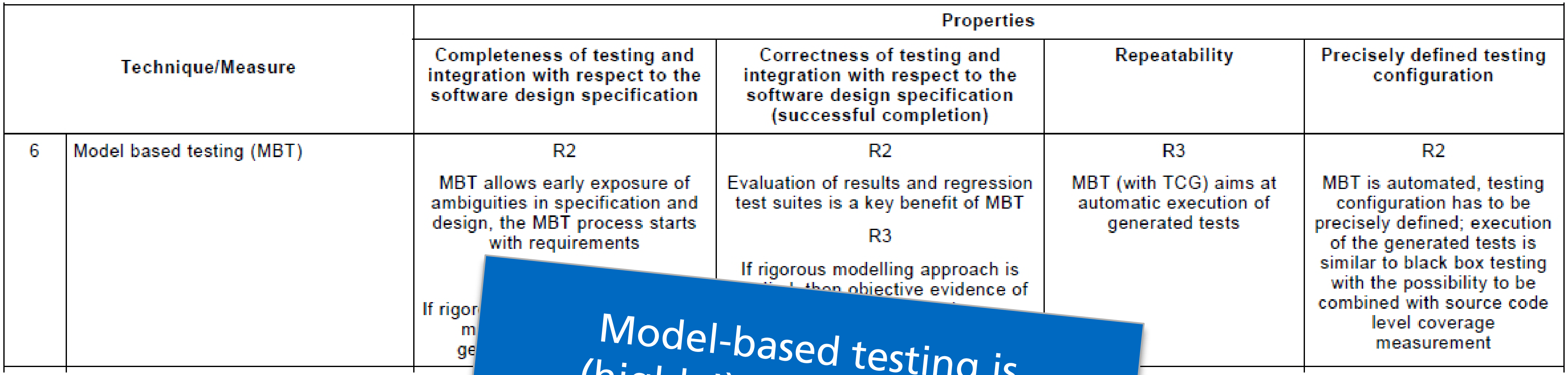


Technique/Measure		Properties			
		Completeness of testing and integration with respect to the software design specification	Correctness of testing and integration with respect to the software design specification (successful completion)	Repeatability	Precisely defined testing configuration
6	Model based testing (MBT)	R2 MBT allows early exposure of ambiguities in specification and design, the MBT process starts with requirements R3 If rigorous reasoning is applied to modelling, and test case generation (TCG) is used	R2 Evaluation of results and regression test suites is a key benefit of MBT R3 If rigorous modelling approach is applied, then objective evidence of coverage is possible	R3 MBT (with TCG) aims at automatic execution of generated tests	R2 MBT is automated, testing configuration has to be precisely defined; execution of the generated tests is similar to black box testing with the possibility to be combined with source code level coverage measurement

Advantages:

-Early requirements validation
-Automatic test case generation
-Combination of test case generation and code coverage measurement

IEC 61508 – 2010



	Technique/Measure	Properties			
		Completeness of testing and integration with respect to the software design specification	Correctness of testing and integration with respect to the software design specification (successful completion)	Repeatability	Precisely defined testing configuration
6	Model based testing (MBT)	R2 MBT allows early exposure of ambiguities in specification and design, the MBT process starts with requirements If rigor m ge	R2 Evaluation of results and regression test suites is a key benefit of MBT R3 If rigorous modelling approach is applied, then objective evidence of	R3 MBT (with TCG) aims at automatic execution of generated tests	R2 MBT is automated, testing configuration has to be precisely defined; execution of the generated tests is similar to black box testing with the possibility to be combined with source code level coverage measurement

Model-based testing is (highly!) recommended.

Advantages:

-Early requirements validation
-Automatic test case generation
-Combination of test case generation and code coverage measurement

Our mission is to bring model-based testing to industrial practice.

Dr. Stephan Weißleder
stephan.weissleder@first.fraunhofer.de
+49 (0)30 6392 1876