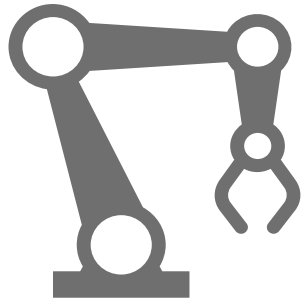
MATLAB EXPO 2019

Comment obtenir des crédits de certification avec Simulink

Daniel Martins

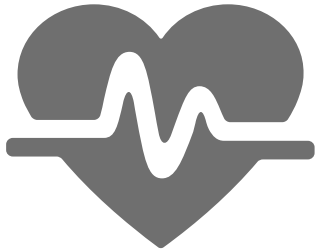
Standards landscape

DO-178C
DO-254

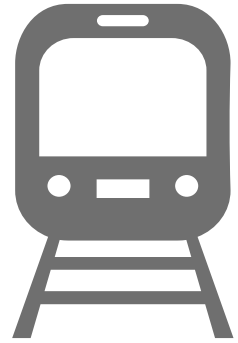


IEC 61508

ISO 26262

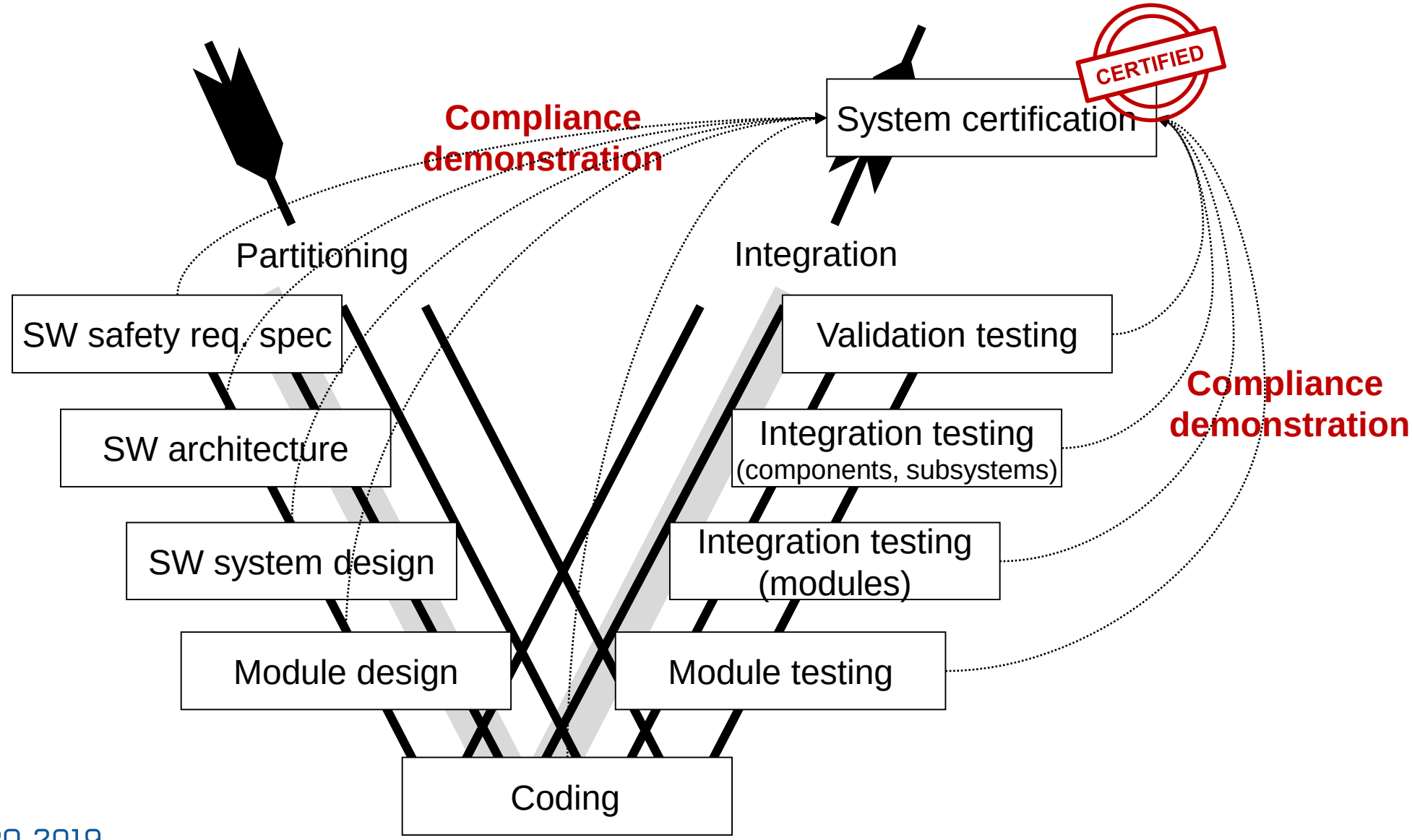


IEC 62304

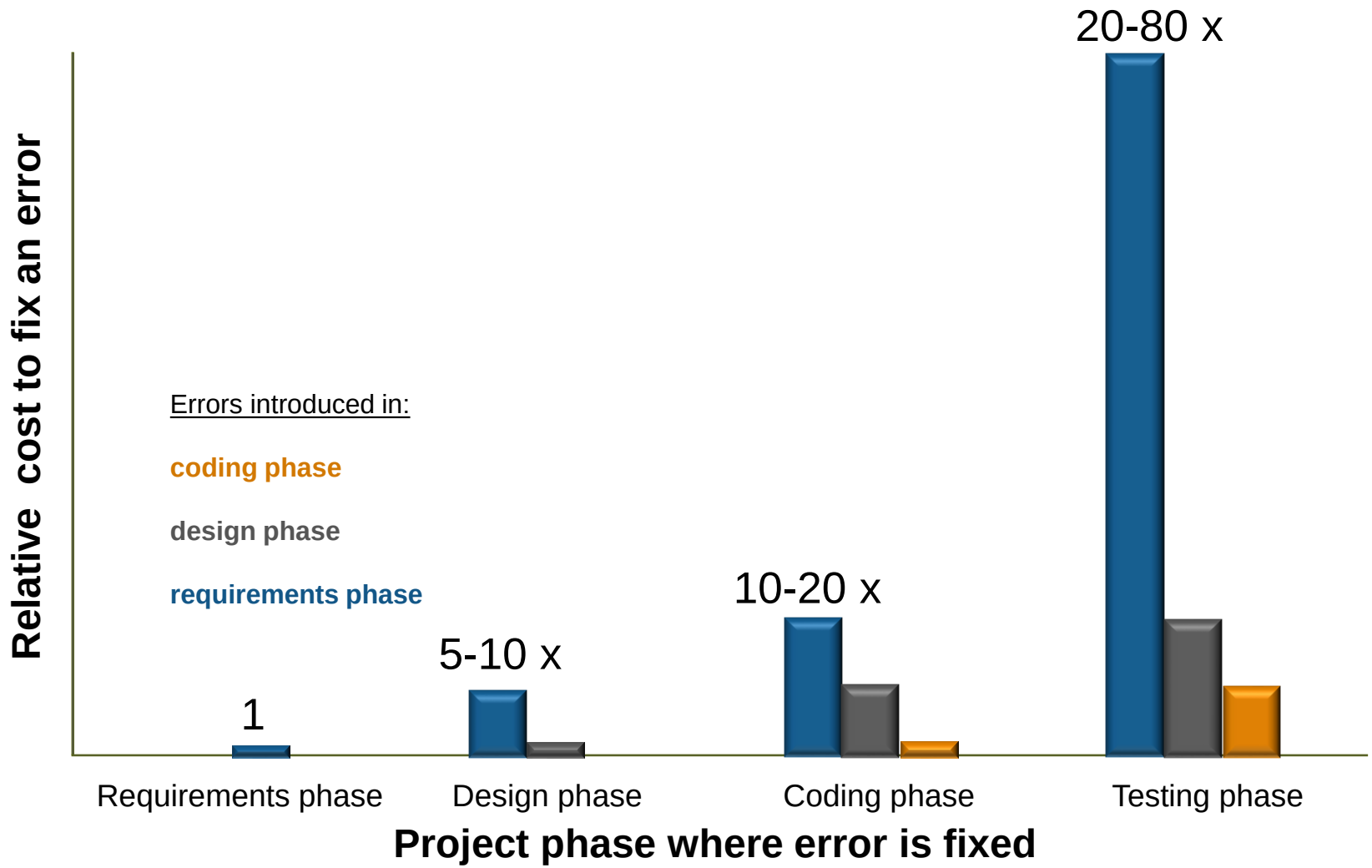


EN 50128

Development process for safety-related systems

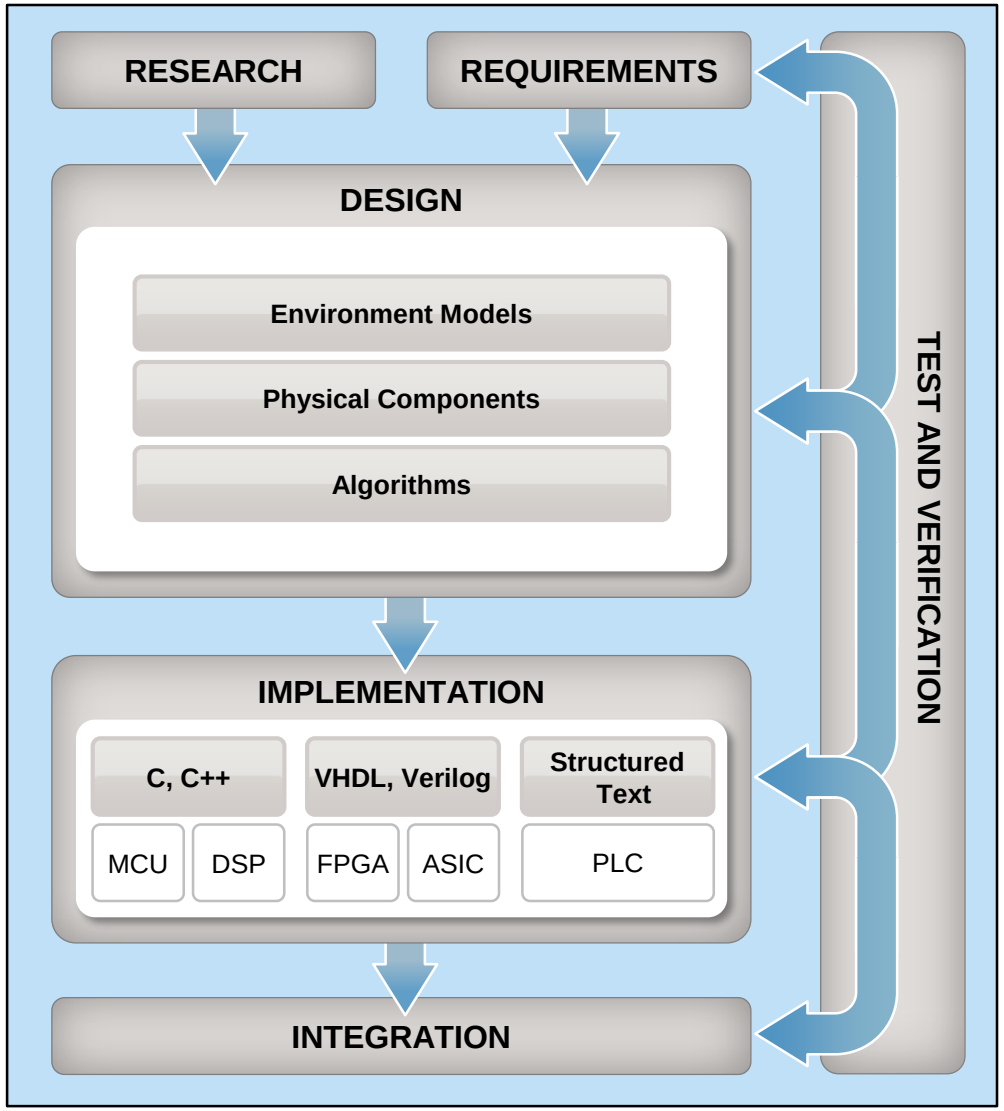


Requirement phase is key in the development process



Source: Return on Investment for Independent Verification & Validation, NASA, 2004.

Model-Based Design: work early on requirements



- Model multidomain systems
- Explore and optimize system behavior
- Collaborate across teams and continents

- Generate efficient code
- Explore and optimize implementation tradeoffs
- Model concurrent systems

- Automate testing
- Detect design errors
- Support certification and standards

Role of Model-Based Design within DO-178C

A Design Model prescribes software component internal data structures, data flow, and/or control flow. A Design Model includes low-level requirements and/or architecture. In particular, when a model expresses software design data, regardless of other content, it should be classified as a Design Model. This includes models used to produce code.

Role of Model-Based Design within ISO 26262

A model consists of function blocks with well-defined inputs and outputs. […]

The functional model can serve as a blueprint for the implementation of embedded software on the control unit through code generation.

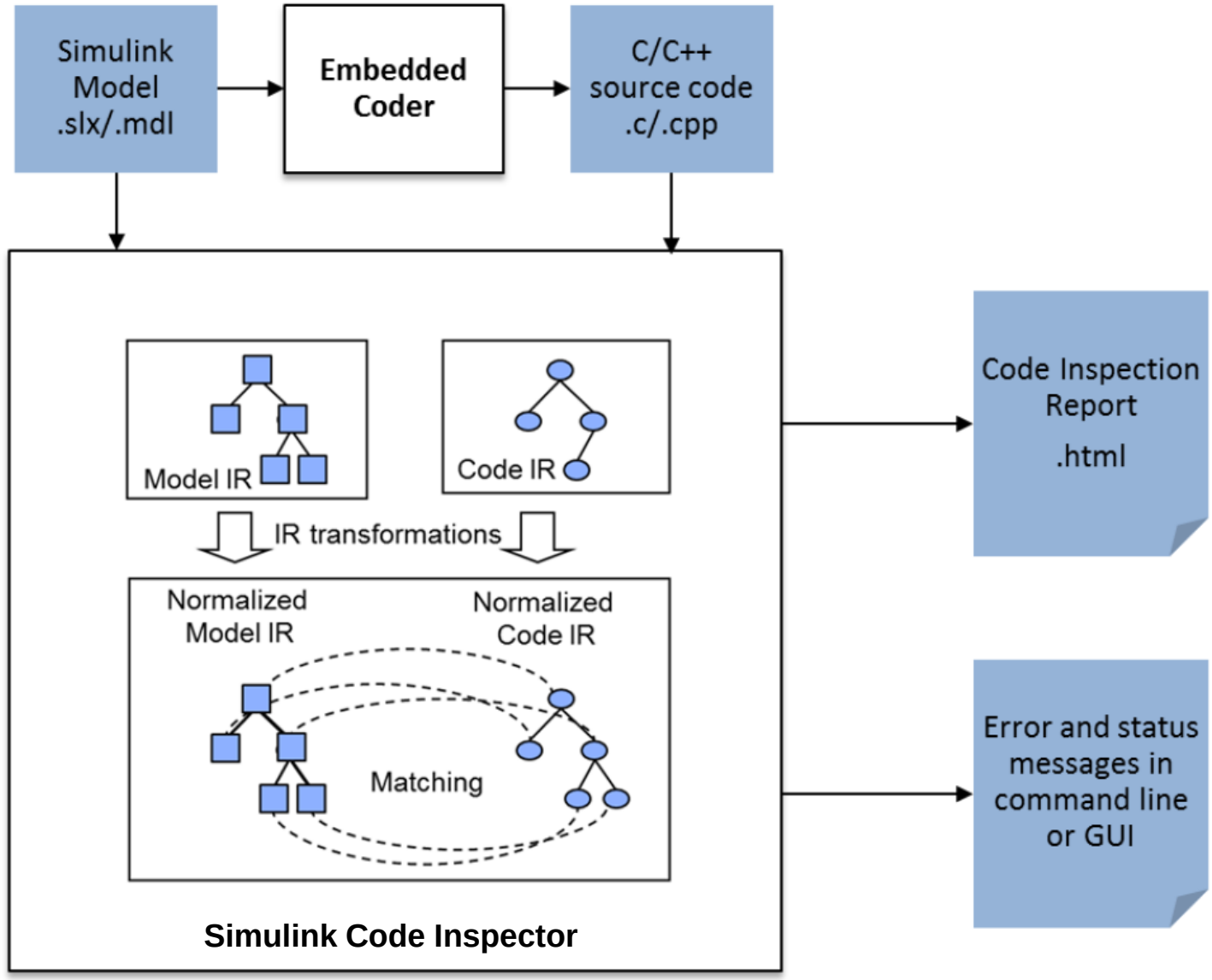
[…]. In comparison to code-based software development with a clear separation of phases, in model-based development a stronger coalescence of the phases “Software safety requirements”, “Software architectural design” and “Software unit design and implementation” can be noted […]. Verification activities can also be treated differently since models can be used as a useful source of information for the testing process (e.g. model-based testing), or can serve as the object to be verified. The seamless utilization of models facilitates highly consistent and efficient development.

Meeting DO-178C Objectives Table A5

Objective	Software Levels	Anticipated Certification Credit [Tool(s)]
(1) Source Code complies with low-level requirements	A, B, C	Full [Simulink Code Inspector]
(2) Source Code complies with software architecture	A, B, C	Full [Simulink Code Inspector]
(3) Source Code is verifiable	A, B	Full [Simulink Code Inspector + Polyspace Bug Finder]
(4) Source Code conforms to standards	A, B, C	Full [Polyspace Bug Finder]
(5) Source Code is traceable to low-level requirements	A, B, C	Full [Simulink Code Inspector]
(6) Source Code is accurate and consistent	A, B, C	Partial [Simulink Code Inspector, Polyspace verifier]

How does *Simulink Code Inspector* work?

IR: Intermediate Representation



Simulink Code Inspector Report

Web Browser - Simulink Code Inspector Report for GearControl.slx

Simulink Code Inspector Report for GearControl.slx

Location: file:///C:/Work/Work19a/TCUsandbox/work/codegen/slprj/slci/GearControl_report.html

Simulink Code Inspector Report for GearControl.slx

Overall Inspection Result : Passed
Utils Need Manual Review : No

Code Verification Results : Verified

Function Interface Verification Results : Verified

Function	Status	Details
GearControl_initialize	Verified	-
GearControl	Verified	-

Model To Code Verification Results : Verified

Status	Details	
Verified	Model objects with status Verified :	18
	Model objects with status Partially processed :	0
	Model objects with status Unable to process :	0
	Model objects with status Failed to verify :	0

Code To Model Verification Results : Verified

Function	Status	Details
GearControl_initialize	Verified	Function does not have any executable code

Web Browser - Simulink Code Inspector Report for GearControl.slx

Simulink Code Inspector Report for GearControl.slx

Location: file:///C:/Work/Work19a/TCUsandbox/work/codegen/slprj/slci/GearControl_report.html

Traceability Results : Traced

Model To Code Traceability Results : Traced

Status	Number of model objects
Traced	18
Partially processed	0
Unable to process	0
Failed to trace	0

Code To Model Traceability Results : Traced

Status	Number of code lines
Traced	52
Nonfunctional code	91
Not processed	2
Partially processed	0
Unable to process	0
Failed to trace	0

Not processed code:

File : GearControl.c

Code location	Code
16	#include "GearControl.h"
17	#include "GearControl_private.h"

Meeting IS026262 Objectives

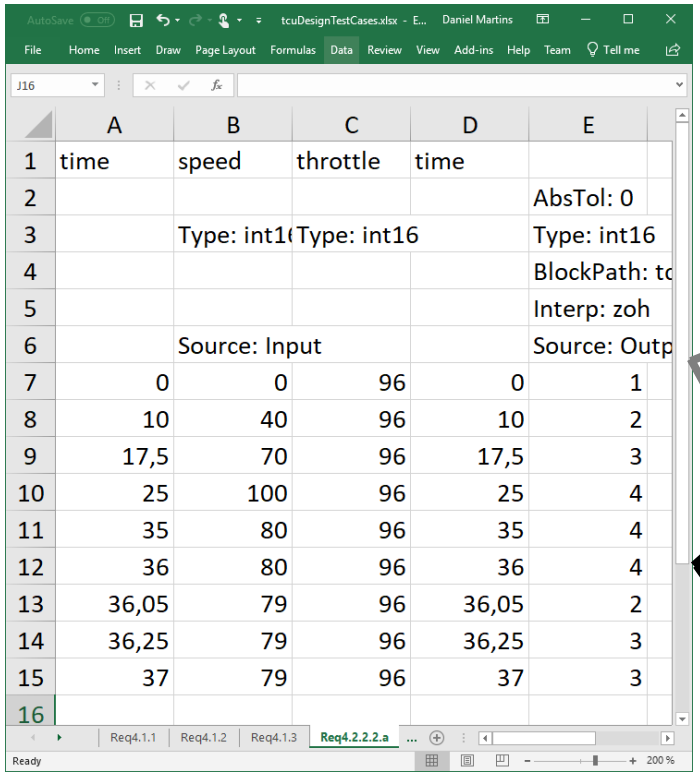
- Table 7: Methods for Software Unit Verification

	Methods	**ASIL**				**MBD Tools**
		A	B	C	D	
1n	Back-to-back comparison test between model and code, if applicable	+	+	++	++	Simulink Test Embedded Coder SIL/PIL

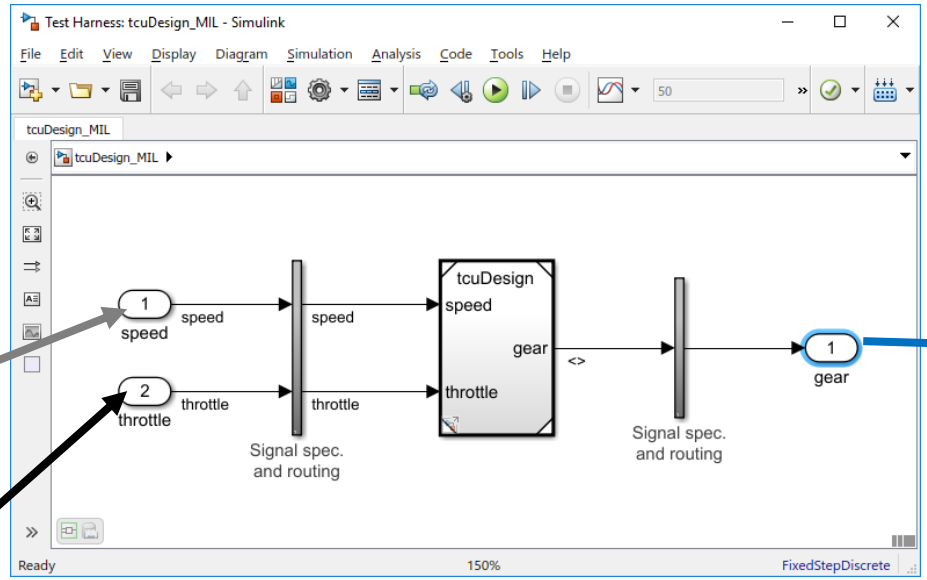
- Table 9: Structural Coverage Metrics at the Software Unit Level

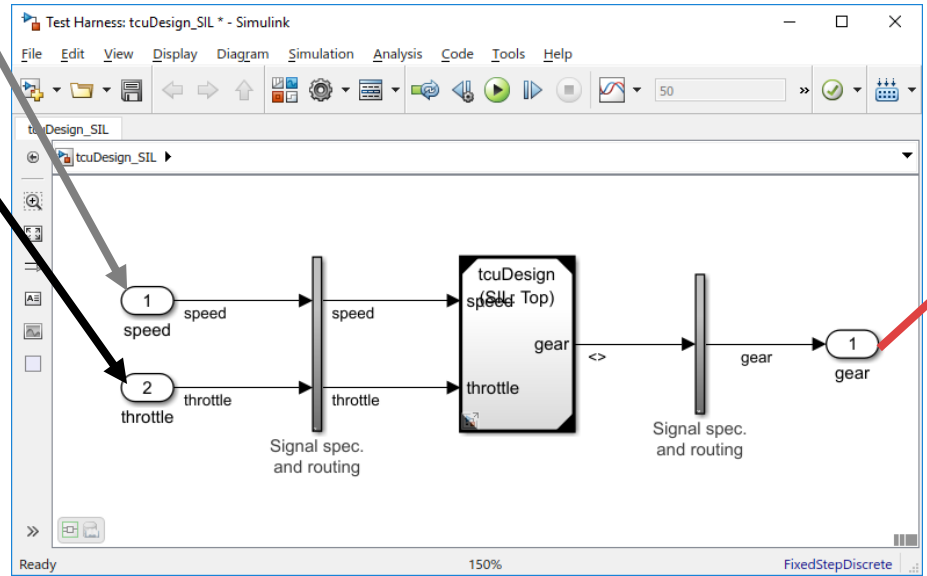
	Methods	**ASIL**				**MBD Tools**
		A	B	C	D	
1a	Statement coverage	++	++	+	+	Simulink Coverage
1b	Branch coverage	+	++	++	++	Simulink Coverage
1c	MC/DC Modified (Condition/Decision Coverage)	+	+	+	++	Simulink Coverage

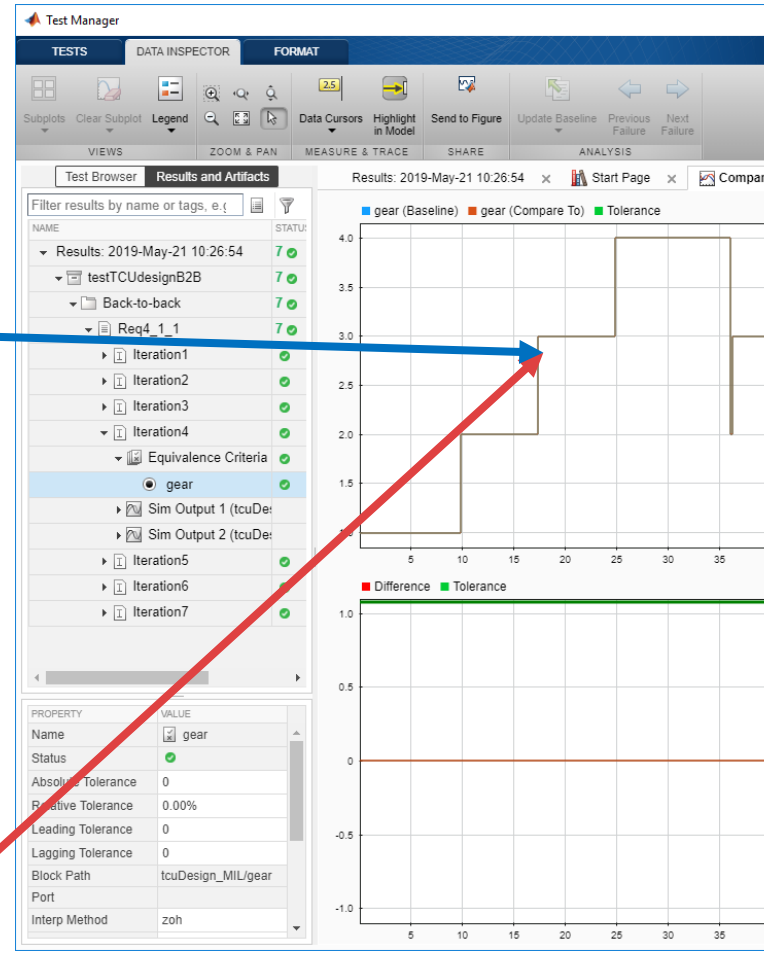
Back-to-back testing



	A	B	C	D	E
1	time	speed	throttle	time	
2					AbsTol: 0
3		Type: int1(	Type: int16		Type: int16
4					BlockPath: tc
5					Interp: zoh
6		Source: Input			Source: Outp
7	0	0	96	0	1
8	10	40	96	10	2
9	17,5	70	96	17,5	3
10	25	100	96	25	4
11	35	80	96	35	4
12	36	80	96	36	4
13	36,05	79	96	36,05	2
14	36,25	79	96	36,25	3
15	37	79	96	37	3
16					







Simulink Coverage report

Web Browser - SIL Coverage Report for ShiftThreshold

SIL Coverage Report for ShiftThreshold

Location: /C:/Users/dmartins/AppData/Local/Temp/tpbafd046c_450a_48e0_9938_dd2886b2935b/tpd6e7d95b_c6d4_4046_9268_c9eda6a4a330.

Summary

File Contents/Complexity		Test 1					
		Decision		Statement		Function	Function call
1 . ShiftThreshold.c	8	100%		100%		100%	100%
2 . . . ShiftThreshold	7	100%		100%		100%	100%
3 . . . ShiftThreshold_initialize	1	--		100%		100%	--

Details

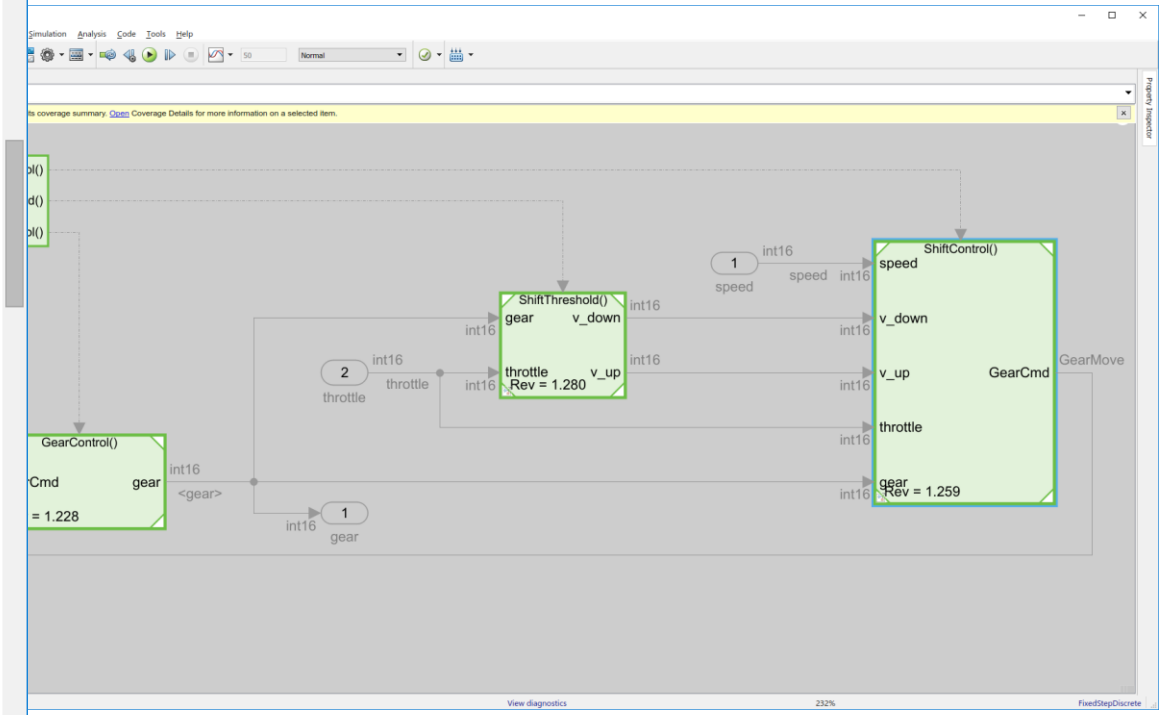
1. File ShiftThreshold.c

Function: ShiftThreshold (line 21)
ShiftThreshold_initialize (line 84)

Metric	Coverage
Cyclomatic Complexity	8
Decision (D1)	100% (8/8) decision outcomes
Statement	100% (20/20) covered statements
Function	100% (2/2) covered functions
Function call	100% (6/6) covered function calls

2. Function ShiftThreshold (line 21)

File: ShiftThreshold.c (code)
Model Object: ShiftThreshold
Covered expressions: (int32_T)*rtu_gear (line 30)
(int32_T)*rtu_gear (line 59)



What DO-178C says about tool qualification

Qualification of a tool is needed when processes of this document are eliminated, reduced, or automated by the use of a software tool without its output being verified .

The purpose of the tool qualification process is to ensure that the tool provides confidence at least equivalent to that of the process(es) eliminated, reduced, or automated.

What ISO26262-8 says about tool qualification

A software tool used in the development of a system or its software or hardware elements, can support or enable a tailoring of the safety-lifecycle […]. In such cases confidence is needed that the software tool effectively achieves the following goals:

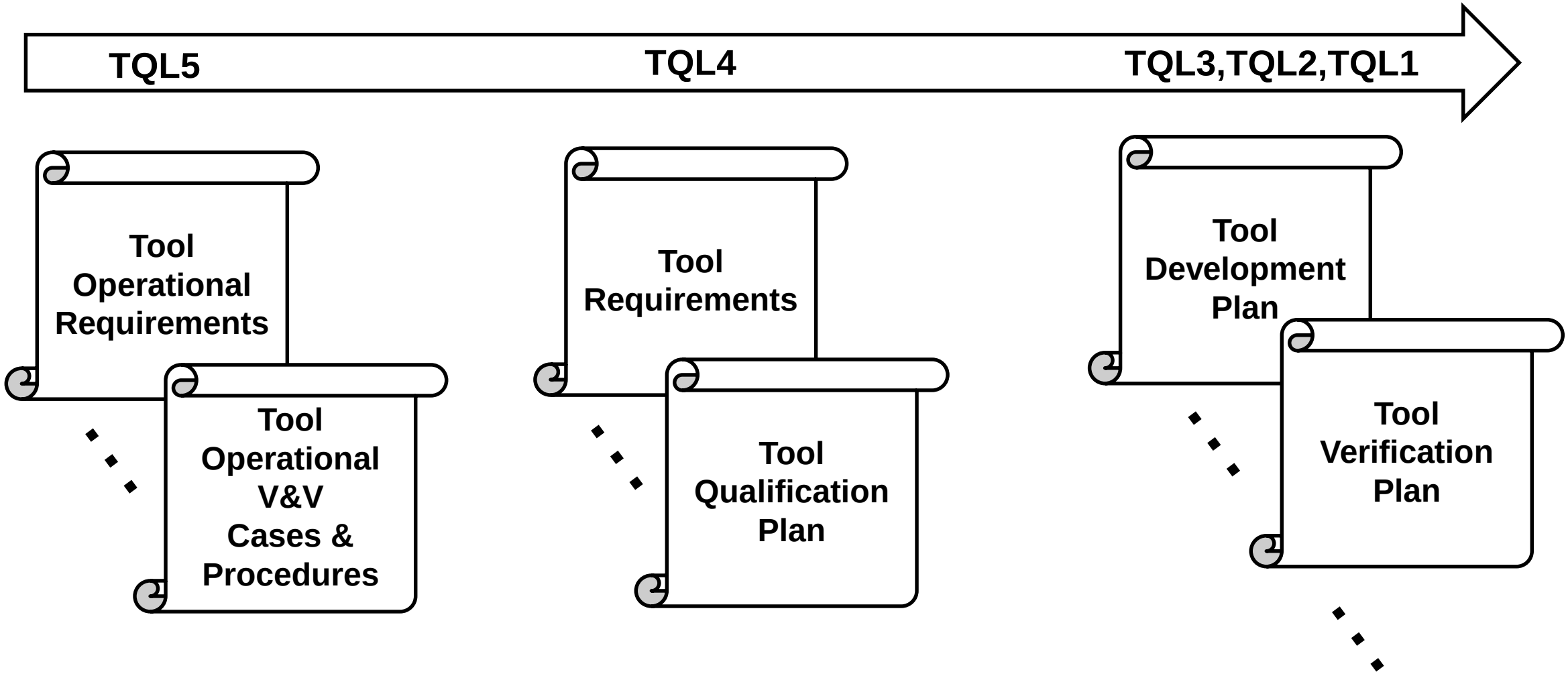
- the risk of systematic faults in the developed product due to malfunctions of the software tool leading to erroneous outputs is minimized, and
- the development process is adequate with respect to compliance with ISO 26262, if activities or tasks required by ISO 26262 rely on the correct functioning of the software tool used.

DO-178C Tool Classification

Criteria	Tools that
1	could insert an error.
2	could fail to detect an error, and are used eliminate/reduce: 1. Other verification process(es) 2. Development process(es) impacting the software
3	could fail to detect an error.

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

DO-178C Tool qualification methods

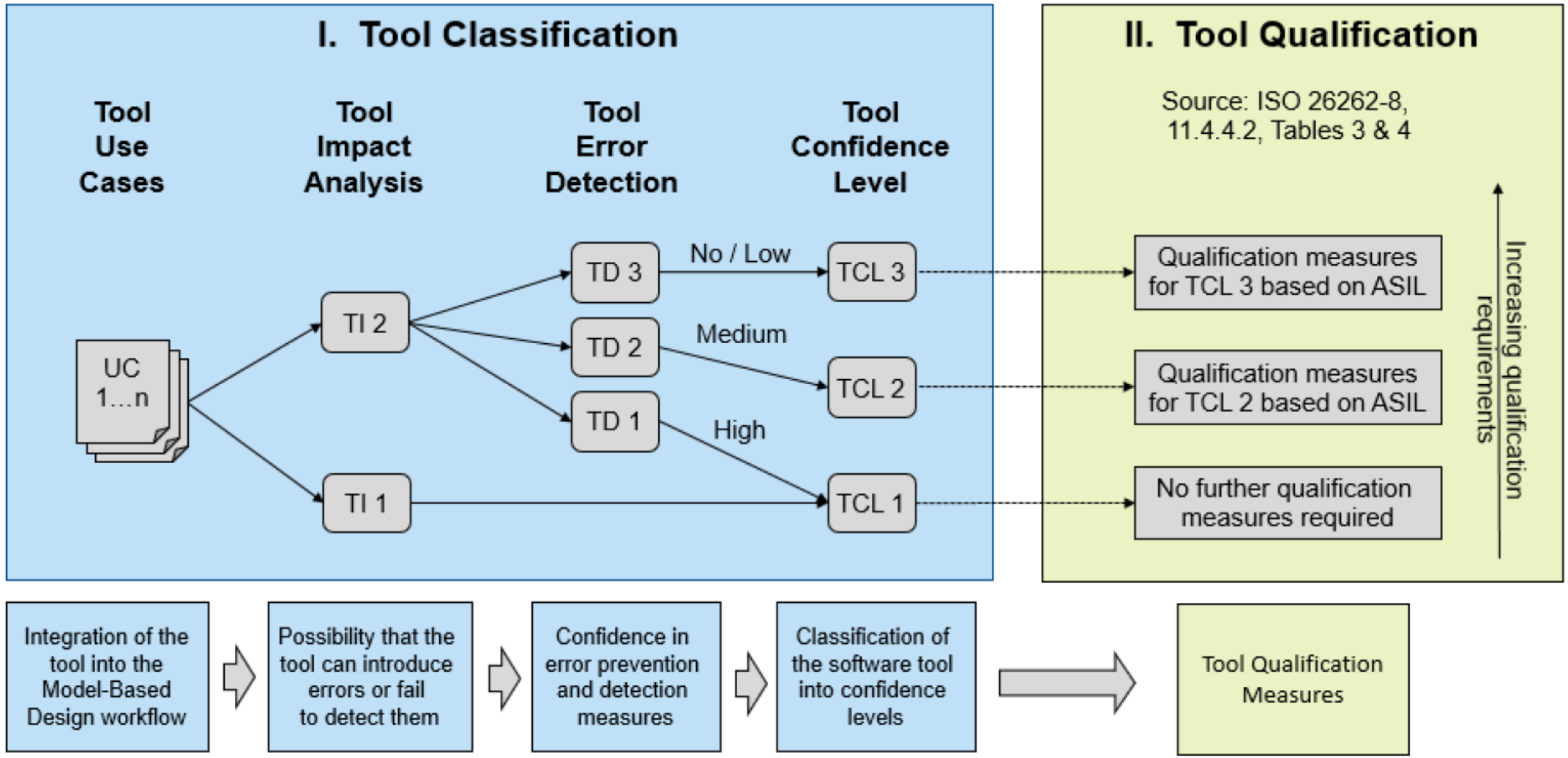


MathWorks DO Qualification Kit *(for DO-178)*

- **Tool Operational Requirements**
- **Test cases and procedures**
- **Tool Qualification plan**

- Polyspace Bug Finder
- Polyspace Code Prover
- Simulink Requirements
- Simulink Report Generator
- Simulink Check
- Simulink Coverage
- Simulink Code Inspector
- Simulink Test
- Simulink Design Verifier
- Model Comparison

ISO26262 Tool classification



ISO26262 Tool qualification methods

	Methods	ASIL				MBD Tools	
		A	B	C	D		
1a	Increased confidence from use in accordance with 11.4.7	++	++	++	+		
1b	Evaluation of the tool development process in accordance to 11.4.8	++	++	++	+	IEC Certification Kit	**TÜV SÜD Certificate**
1c	Validation of the software too in accordance with 11.4.9	+	+	+	++	IEC Certification Kit	**Test Cases & Procedures**
1d	Development in accordance with a safety standard	+	+	+	+		

MathWorks IEC Certification Kit *(for ISO 26262 and IEC 61508)*

- **Workflow description**
- **Tool Qualification plan**
- **TÜV SÜD Certificate**
- **Test cases and procedures**

- Embedded Coder
- Simulink PLC Coder
- Polyspace Bug Finder
- Polyspace Code Prover
- Simulink Check
- Simulink Coverage
- Simulink Test
- Simulink Design Verifier

Summary

- **Models are accepted by Standards**
- **Standards recognize benefits of tools**
- **Several Standards activities can be automated by models and tools**
- **MathWorks Certification/Qualification Kits describe those activities**