

# Process Pitfalls in ISO 26262 Compliance

MathWorks Consulting  
Jason Moore

# Functional safety industry trend

**Driver assistance is a priority for major auto players**  
 Mentions of “advanced driver assistance,” “ADAS,” and “safety,” on automaker and Tier-1 supplier earnings calls

Number of mentions



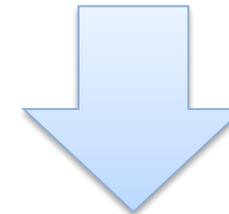
- Internal IEC Certification Kit data (Normalized)
- Support for ISO 26262

Source: cbinsights.com

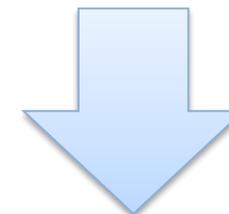
CBINSIGHTS

Growing interest in

safety



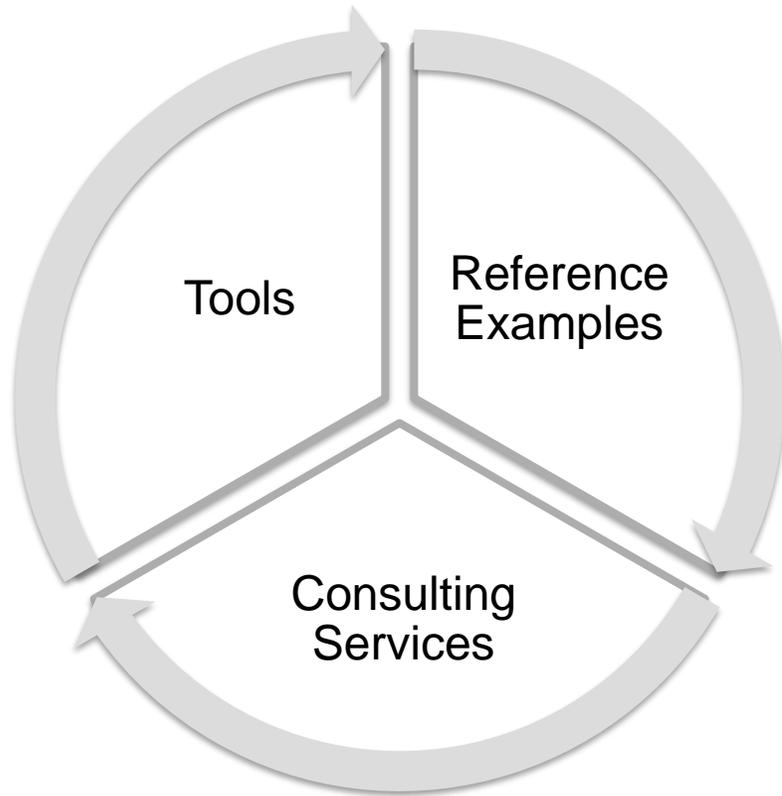
**ISO 26262**



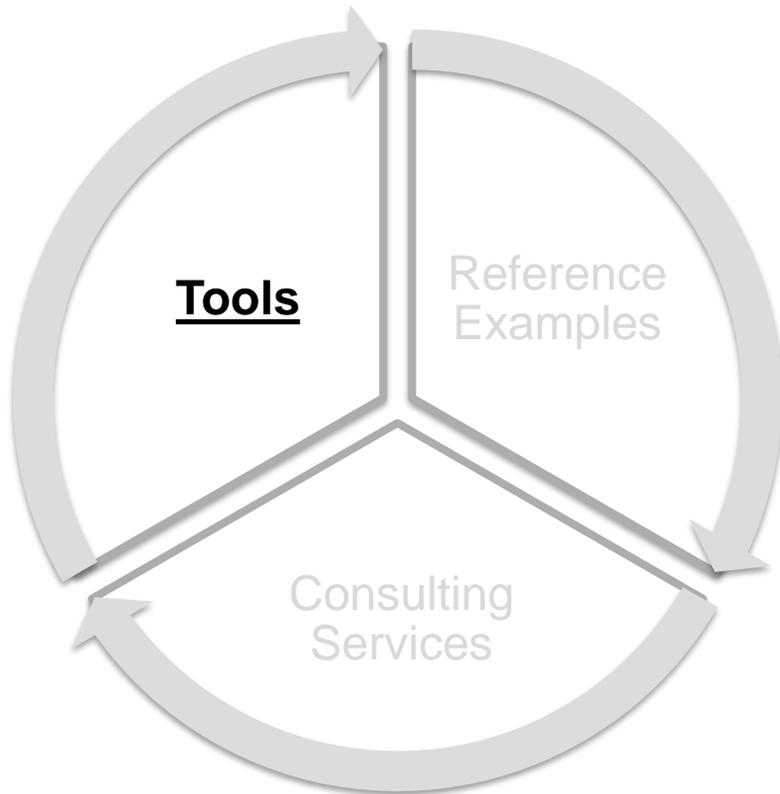
Is your organization

ready for ISO?

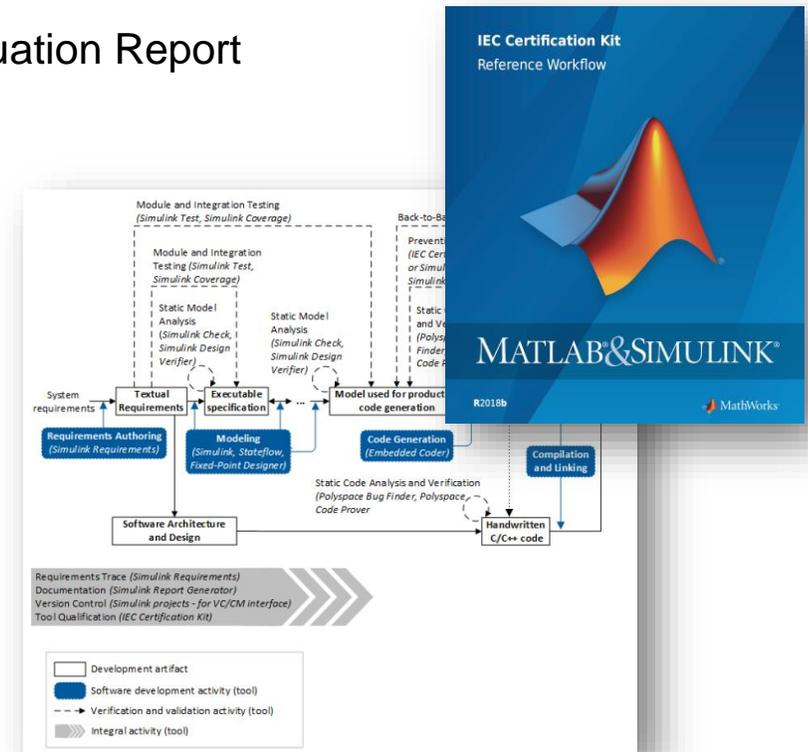
# Multifaceted support for ISO 26262



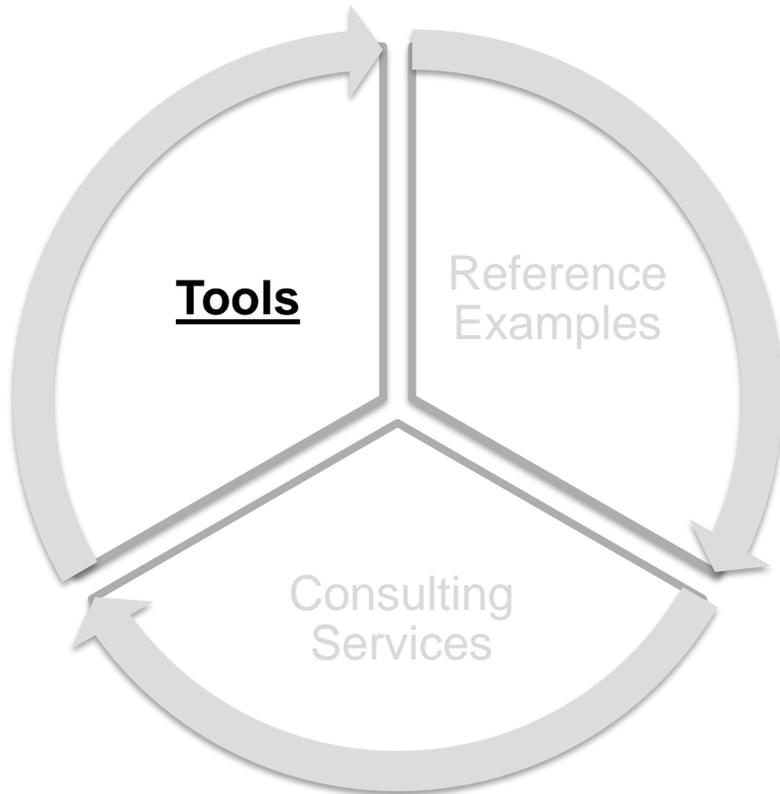
# Multifaceted support for ISO 26262



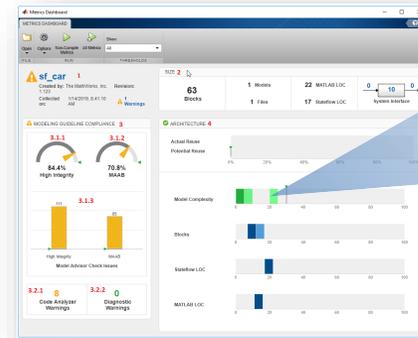
- IEC Certification Kit
  - Model-Based Design Reference Workflow
  - Tool Qualification Package
    - Software Tool Criteria Evaluation Report
    - Software Tool Qualification
    - Tool Validation Suite
    - ...etc.



# Multifaceted support for ISO 26262



- Targeted Features
  - Model Metrics Dashboard



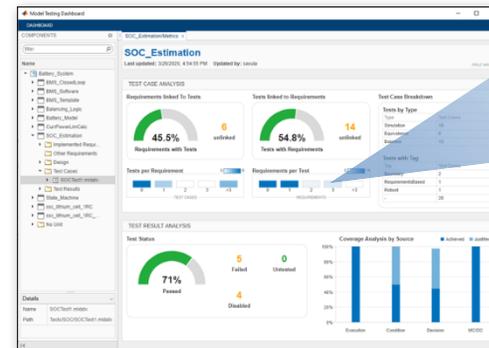
R2017b

## Design Compliance

Table 1 — Topics to be covered by modelling and coding guidelines

Topics	ASIL			
	A	B	C	D
1a Enforcement of low complexity <sup>a</sup>	++	++	++	++
1b Use of language subsets <sup>b</sup>	++	++	++	++
1c Enforcement of strong typing <sup>c</sup>	++	++	++	++
1d Use of defensive implementation techniques <sup>d</sup>	+	+	++	++
1e Use of well-trusted design principles <sup>e</sup>	+	+	++	++
1f Use of unambiguous graphical representation	+	++	++	++
1g Use of style guides	+	++	++	++
1h Use of naming conventions	++	++	++	++
1i Concurrency aspects <sup>f</sup>	+	+	+	+

- Model Testing Dashboard



R2020b

## Verification Compliance

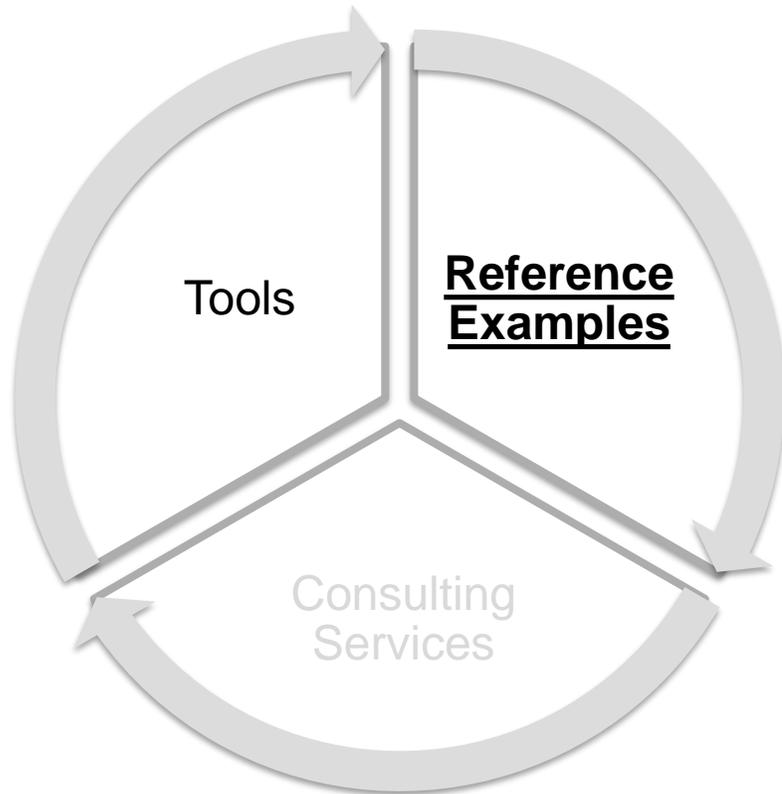
Table 8 — Methods for deriving test cases for software unit testing

Methods	ASIL			
	A	B	C	D
1a Analysis of requirements	++	++	++	++
1b Generation and analysis of equivalence classes <sup>a</sup>	+	++	++	++
1c Analysis of boundary values <sup>b</sup>	+	++	++	++
1d Error guessing based on knowledge or experience <sup>c</sup>	+	+	+	+

Table 9 — Structural coverage metrics at the software unit level

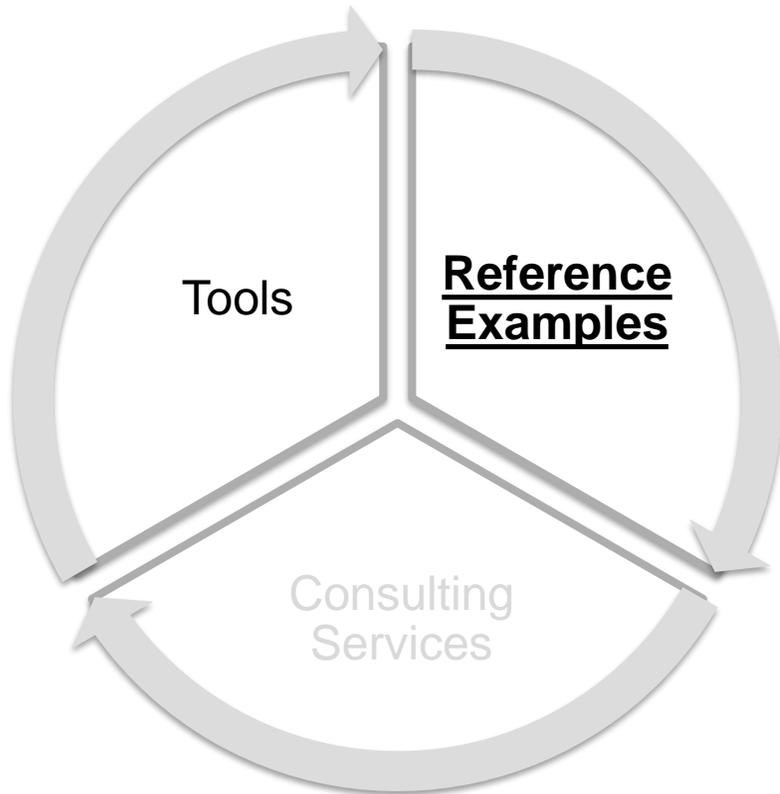
Methods	ASIL			
	A	B	C	D
1a Statement coverage	++	++	+	+
1b Branch coverage	+	++	++	++
1c MC/DC (Modified Condition/Decision Coverage)	+	+	+	++

# Multifaceted support for ISO 26262



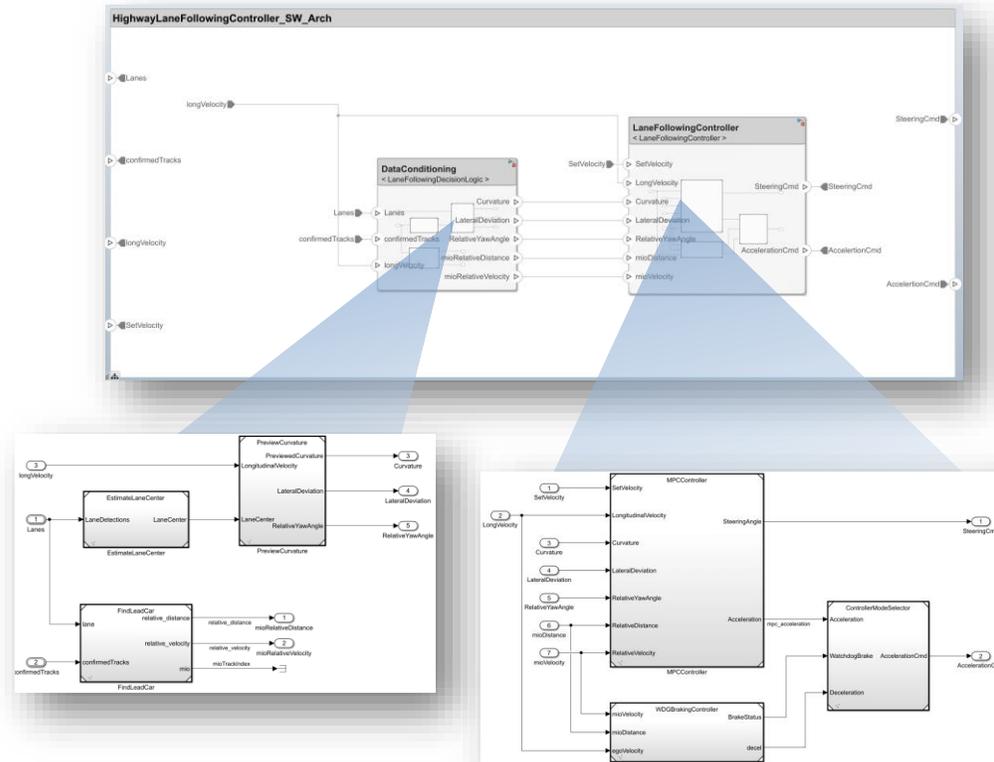
- Best Practice Paper
  - (2018) [Model Quality Objectives](#)
    - Recommended model metric and threshold
  - (2019) [11 Best Practices for Developing ISO 26262 Applications with Simulink](#)
    - How to achieve Freedom from Interference?
  - (2020) [An ISO 26262 Workflow for Automated Driving Applications Using MATLAB: Guidelines and Best Practices](#)
    - Use of MATLAB as part of ISO 26262 workflow

# Multifaceted support for ISO 26262



- Reference Application

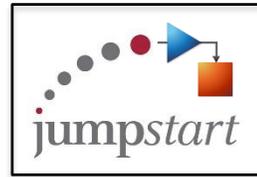
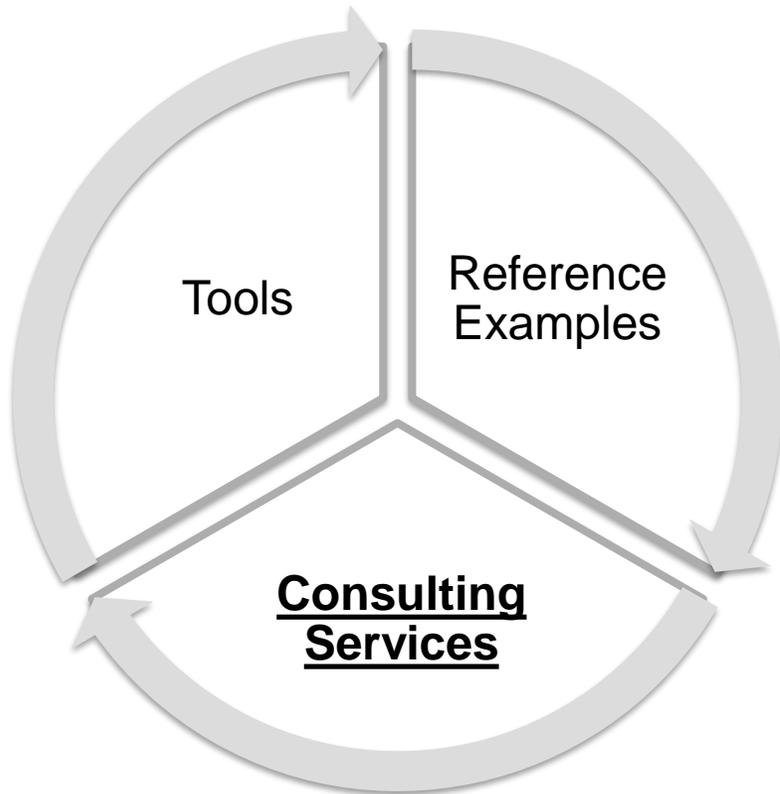
## Architecture Design with System Composer



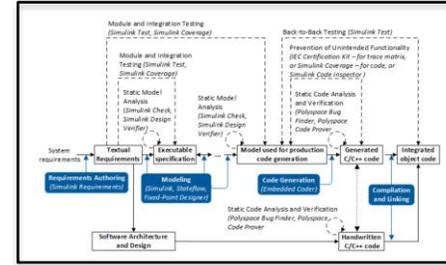
ISO 26262-6  
Workflow Example

## Component/Unit Design with Simulink

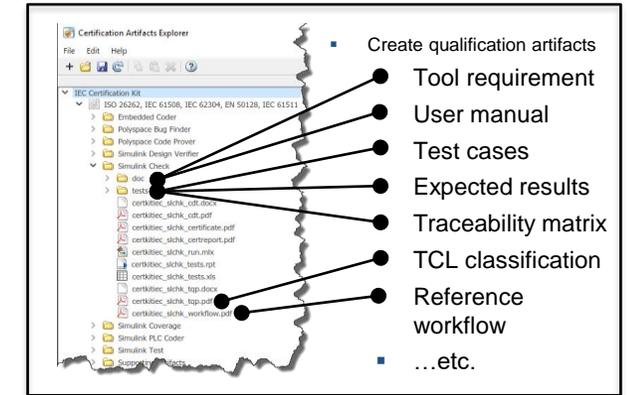
# Multifaceted support for ISO 26262



ISO Jumpstart



Process Establishment



Tool Qualification Support

## Range of Consulting Services



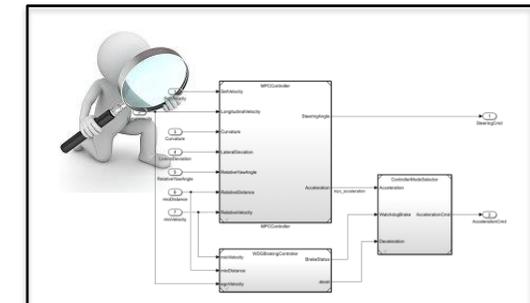
Process Gap Analysis

Model Review



Table 8 — Methods for deriving test cases for software unit testing

Methods	ASIL			
	A	B	C	D
1a Analysis of requirements	++	++	++	++
1b Generation and analysis of equivalence classes <sup>a</sup>	+	++	++	++
1c Analysis of boundary values <sup>b</sup>	+	++	++	++
1d Error guessing based on knowledge or experience <sup>c</sup>	+	+	+	+



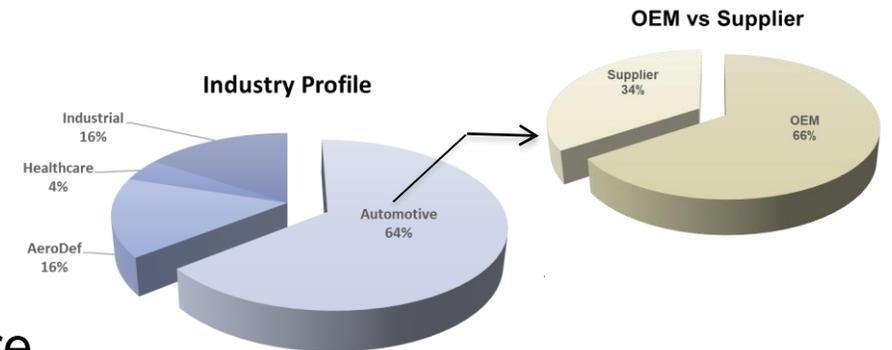
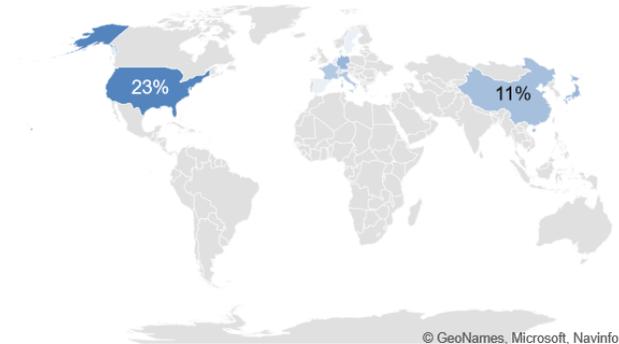
# Observations based on our work with industry

## Common pitfalls

- Unaware of ISO requirements
- Legacy components developed outside of ISO
- No clear mapping of ISO requirement to workflow
- Lack of tool implementation methods against ISO requirement
- No architecture consideration
- HIL-centric verification workflow
- No justification on method selections
- No clear definition of required work product
- Lack of consistency in work product
- No upfront consideration to tool qualification
- Lack of coordination between functional safety and software development
- Underestimate the effort (cost and timing) required for ISO project
- ...etc.

Process Assessment Locations

% of total assessments 2% 23%

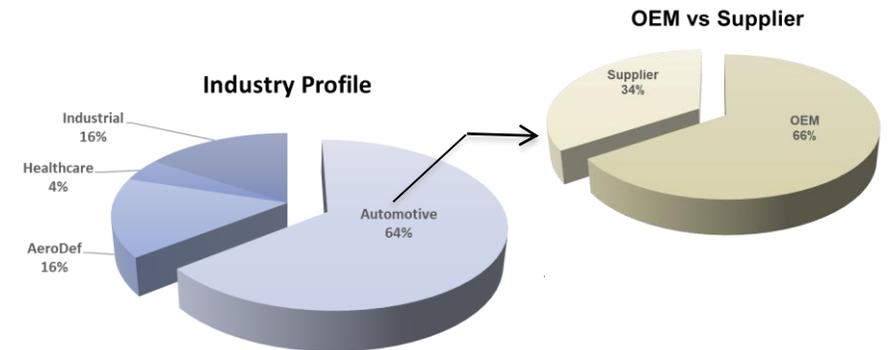
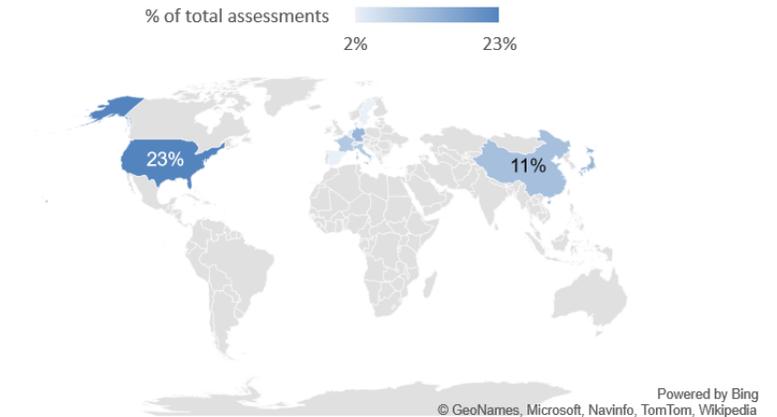


# Observations based on our work with industry

## Common Themes

- Process not clearly defined or documented
- Lack top-down architectural design approach
- Poor tool qualification awareness

### Process Assessment Locations



# Process not clearly defined or documented

1. Vocabulary		
2. Management of functional safety		
2-5 Overall safety management	2-6 Project dependent safety management	2-7 Safety management regarding production, operation, service and decommissioning
3. Concept phase		
3-5 Item definition	3-6 Hazard analysis and risk assessment	3-7 Functional safety concept
4. Product development at the system level		
4-5 General topics for the product development at the system level	4-6 Technical safety concept	4-7 System and item integration and testing
5. Product development at the hardware level		
5-5 General topics for the product development at the hardware level	5-6 Specification of hardware safety requirements	5-7 Evaluation of the hardware architectural metrics
6. Product development at the software level		
6-5 General topics for the product development at the software level	6-6 Specific software safety requirements	6-7 Software unit testing and verification
7. Production, operation, service and decommissioning		
7-5 Planning for production, operation, service and decommissioning	7-6 Production	7-7 Operation, service and decommissioning
8. Supporting processes		
8-5 Interfaces within distributed developments	8-6 Specification and management of safety requirements	8-7 Configuration management
8-8 Change management	8-9 Verification	8-10 Documentation management
8-11 Confidence in the use of software tools	8-12 Qualification of software components	8-13 Evaluation of hardware elements
8-14 Proven in use argument	8-15 Interfacing an application that is out of scope of ISO 26262	8-16 Integration of safety-related systems not developed according to ISO 26262
9. Automotive safety integrity level (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 analysis		
10. Guidelines on ISO 26262		
11. Guidelines on application of ISO 26262 to semiconductors		

Table 1 — Topics to be covered by modelling and coding guidelines

Topics	ASIL			
	A	B	C	D
1a Enforcement of low complexity <sup>a</sup>	++	++	++	++
1b Use of language subsets <sup>b</sup>	++	++	++	++
1c Enforcement of strong typing <sup>c</sup>	++	++	++	++
1d Use of defensive implementation techniques <sup>d</sup>	+	+	++	++

Table 3 — Principles for software architectural design

Principles	ASIL			
	A	B	C	D
1a Appropriate hierarchical structure of the software components	++	++	++	++
1b Restricted size and complexity of software components <sup>a</sup>	++	++	++	++

Table 4 — Methods for the verification of the software architectural design

Methods	ASIL			
	A	B	C	D
1a Walk-through of the design <sup>a</sup>	++	+	0	0
1b Inspection of the design <sup>a</sup>	+	++	++	++
1c Simulation of dynamic behaviour of the design	+	+	+	++
1d Prototype generation	0	0	+	++
1e Formal verification	0	0	+	+
1f Control flow analysis <sup>b</sup>	+	+	++	++
1g Data flow analysis <sup>b</sup>	+	+	++	++
1h Scheduling analysis	+	+	++	++

Table 14 — Methods for tests of the embedded software

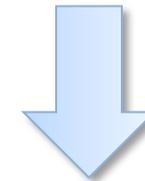
Methods	ASIL			
	A	B	C	D
1a	++	++	++	++
1b	+	+	+	++

Table 15 — Methods for deriving test cases for the test of the embedded software

Methods	ASIL			
	A	B	C	D
1a Analysis of requirements	++	++	++	++
1b Generation and analysis of equivalence classes	+	++	++	++
1c Analysis of boundary values	+	+	++	++
1d Error guessing based on knowledge or experience	+	+	++	++
1e Analysis of functional dependencies	+	+	++	++
1f Analysis of operational use cases <sup>a</sup>	+	++	++	++

## ISO 26262-6

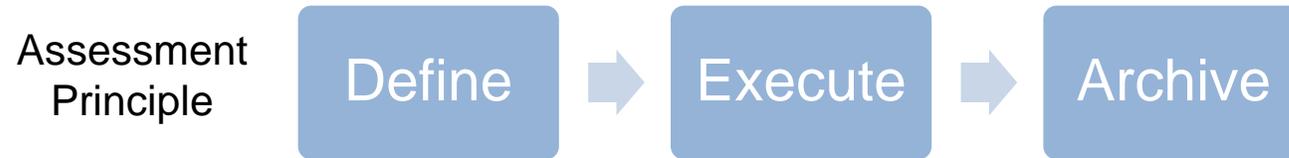
- 15 Tables
- 90 Topics/Methods/Principles



- Which **topics/method/principles** were chosen?
- What **justification** were used?
- What **evidence** were captured?
- What are the **implementation** steps?
- ...etc.?

# Process not clearly defined or documented

Define process: from ISO Requirement down to Detail Work Instructions



# Process not clearly defined or documented

## Define process: from ISO Requirement down to Detail Work Instructions

Assessment Principle



ISO recommendations

Table 7 – Methods for software unit verification

Methods	ASIL D	Applicable
1a	0	No*
1b	+	No*
1c	++	Yes
1d	++	Yes
1e	+	Yes
1f	++	Yes
1g	++	Yes
1h	++	Yes
1i	+	No
1j	++	Yes
1k	++	Yes
1l	++	Yes
1m	++	Yes
1n	++	Yes

Decision to follow recommendation

Mapping to Engineering Task

Task Name	ISO Requirement	Work Product
Model Review	ISO 26262-6:2018 Table 3-1b ISO 26262-6:2018 Table 3-1c ISO 26262-6:2018 Clause 5 ISO 26262-6:2018 Table 7-1c ISO 26262-8:2018 6.4.2.3 ISO 26262-8:2018 6.4.3.2	Model Review Checklist
Model Testing	ISO 26262-6:2018 Table 7-1d ISO 26262-6:2018 Table 7-1j ISO 26262-6:2018 Table 7-1k ISO 26262-6:2018 Table 9-1a ISO 26262-6:2018 Table 9-1b ISO 26262-6:2018 Table 9-1c	Software Unit Test Verification Specification Model Unit Test Tool (Excel Input File) Model Coverage Report Model Coverage Filter Software Unit Test Verification Report
Software-in-the-Loop (SIL) Testing	ISO 26262-6:2018 Table 7-1n ISO 26262-6:2018 Table 9-1a ISO 26262-6:2018 Table 9-1b ISO 26262-6:2018 Table 9-1c	Software Unit Test Verification Specification Model Unit Test Tool (Excel Input File) Code Coverage Report Software unit test verification report
Processor-in-the-Loop (PIL) Testing	ISO 26262-6:2018 Table 7-1l ISO 26262-6:2018 Table 7-1m ISO 26262-6:2018 Table 7-1n	Software Unit Test Verification Specification Model Unit Test Tool (Excel Input File) Software Unit Test Verification Report
Code Analysis	ISO 26262-6:2018 Table 6 ISO 26262-6:2018 Table 7-1f ISO 26262-6:2018 Table 7-1g ISO 26262-6:2018 Table 7-1h	Polyspace Bug Finder Report Polyspace Code Prover Report Software Unit Test Verification Specification Software Unit Test Verification Report
Regression Testing	All the above	All the above
Software Unit Verification Checklist Review	ISO 26262-6:2018 Clause 9	Software Unit Verification Checklist

Derive Consistent Work Instructions



Table 8 – Methods for deriving test cases for software unit testing

Methods	ASIL D	Applicable
1a	++	Yes
1b	++	Yes
1c	++	Yes
1d	++	No*

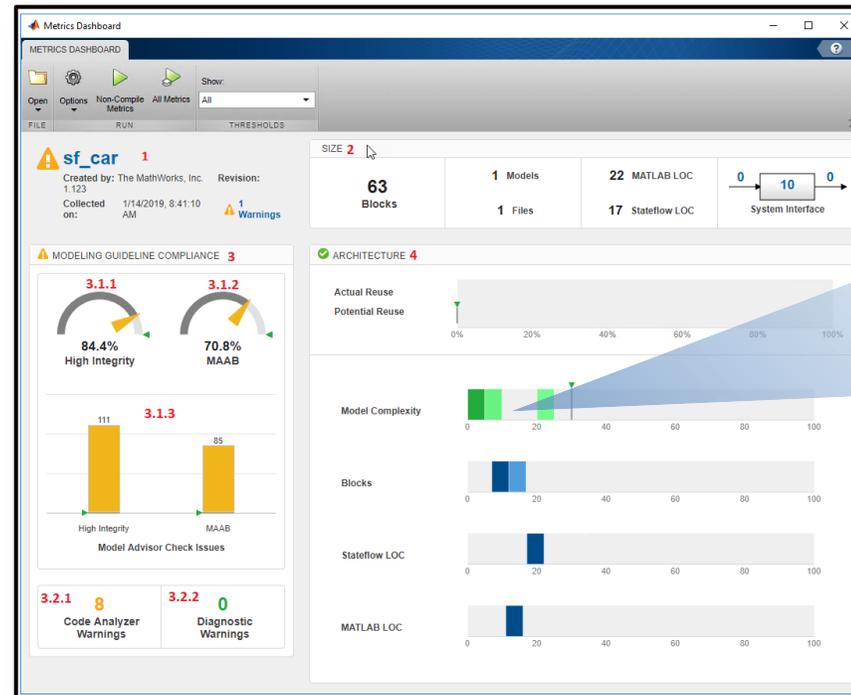
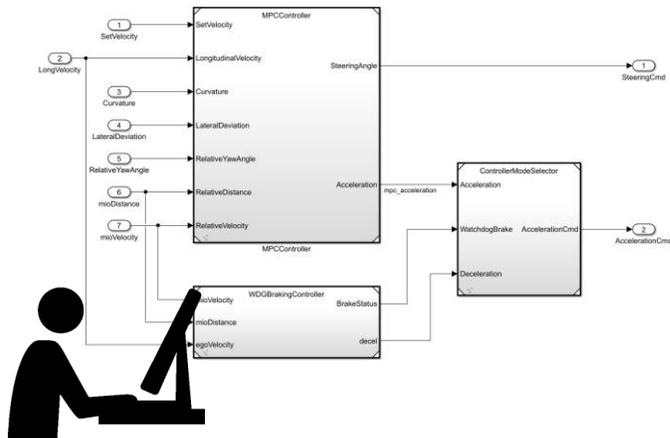
Table 9 – Structural coverage metrics at the software unit level

Methods	ASIL D	Applicable
1a	++	Yes
1b	++	Yes
1c	++	Yes

# Process not clearly defined or documented

## Define process: from ISO Requirement down to Detail Work Instructions

Assessment Principle



Model Metric Dashboard

Table 1 — Topics to be covered by modelling and coding guidelines

Topics	ASIL			
	A	B	C	D
1a Enforcement of low complexity <sup>a</sup>	++	++	++	++
1b Use of language subsets <sup>b</sup>	++	++	++	++
1c Enforcement of strong typing <sup>c</sup>	++	++	++	++
1d Use of defensive implementation techniques <sup>d</sup>	+	+	++	++
1e Use of well-trusted design principles <sup>e</sup>	+	+	++	++
1f Use of unambiguous graphical representation	+	++	++	++
1g Use of style guides	+	++	++	++
1h Use of naming conventions	++	++	++	++
1i Concurrency aspects <sup>f</sup>	+	+	+	+

# Process not clearly defined or documented

Define process: from ISO Requirement down to Detail Work Instructions

Assessment Principle

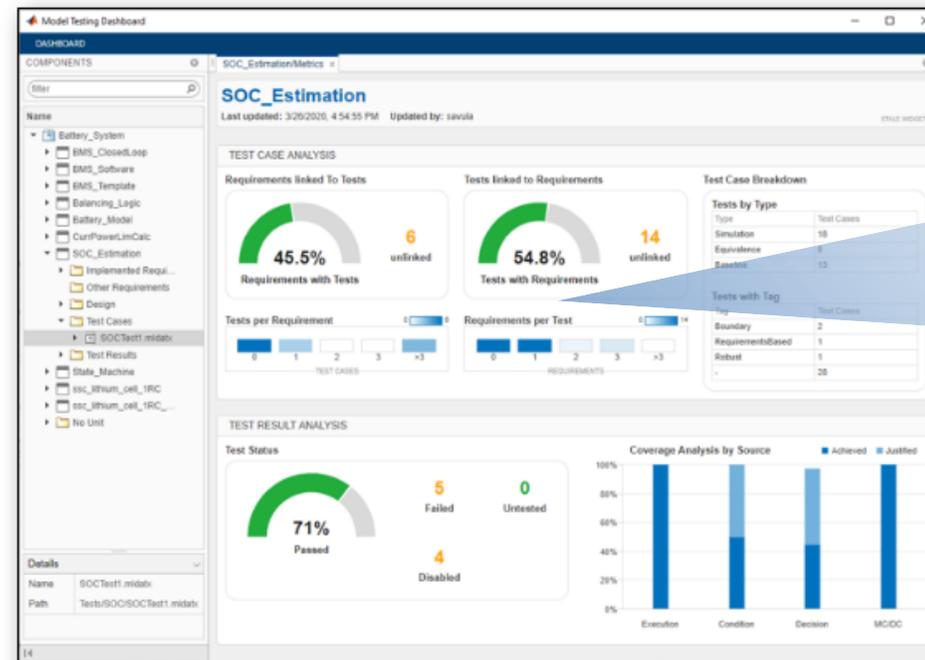
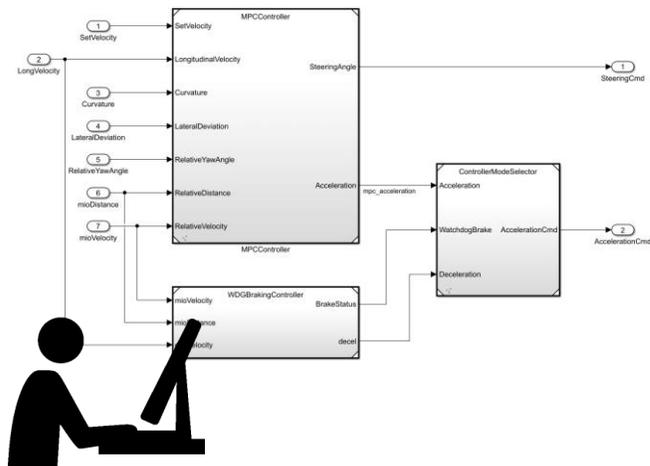


Table 8 — Methods for deriving test cases for software unit testing

Methods	ASIL			
	A	B	C	D
1a Analysis of requirements	++	++	++	++
1b Generation and analysis of equivalence classes*	+	++	++	++
1c Analysis of how down to test				
1d Error				

Table 9 — 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)	+	+	+	++

## Model Testing Dashboard

# Process not clearly defined or documented

## Define process: from ISO Requirement down to Detail Work Instructions

Assessment Principle



### Coverage Results

**Results: 2020-Sep-28 09:15:05**

Result Type: Result Set  
 Parent: None  
 Start Time: 28-Sep-2020 09:15:27  
 End Time: 28-Sep-2020 09:25:39  
 Outcome: Total: 9, Passed: 9

**Aggregated Coverage Results**

Analyzed Model	Sim Mode	Complexity	Decision	Condition	MDC	Execution
<a href="#">LaneFollowingTestBenchExample</a>	Normal	30	67%	60%	33%	99%

[Back to Report Summary](#)

### Test Results

**Summary**

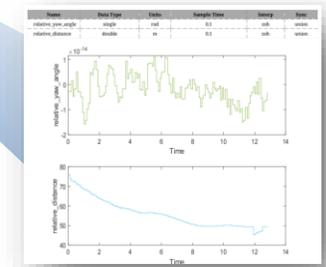
Name	Outcome	Duration (Seconds)
Results: 2020-Sep-28 09:15:05	9/9	612.623
<a href="#">LaneFollowingTestScenarios</a>	9/9	612.625
<a href="#">ACC_ISO_TargetDiscriminationTest</a>	9/9	612.625
<a href="#">ACC_ISO_AutoRetargetTest</a>	9/9	80.803
<a href="#">ACC_ISO_AutoRetargetTest</a>	9/9	68.31
<a href="#">ACC_ISO_CurveTest</a>	9/9	49.75
<a href="#">ACC_StopGo</a>	9/9	73.295
<a href="#">LFACC_DoubleCurve_DecelTarget</a>	9/9	47.105
<a href="#">LFACC_DoubleCurve_AutoRetarget</a>	9/9	47.424
<a href="#">LFACC_DoubleCurve_StopGo</a>	9/9	92.316
<a href="#">LFACC_Curve_CutInOut</a>	9/9	57.526
<a href="#">LFACC_Curve_CutInOut_TooClose</a>	9/9	58.182

**Report Generated by Test Manager**

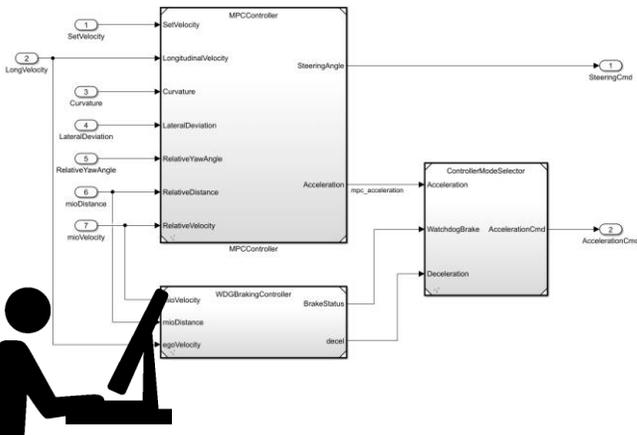
Title: Test  
 Author: Test  
 Date: 28-Sep-2020 09:26:37

Test Environment  
 Platform: PCWIN64 (62020a)  
 MATLAB:

### Verification Report



### Verification Plots

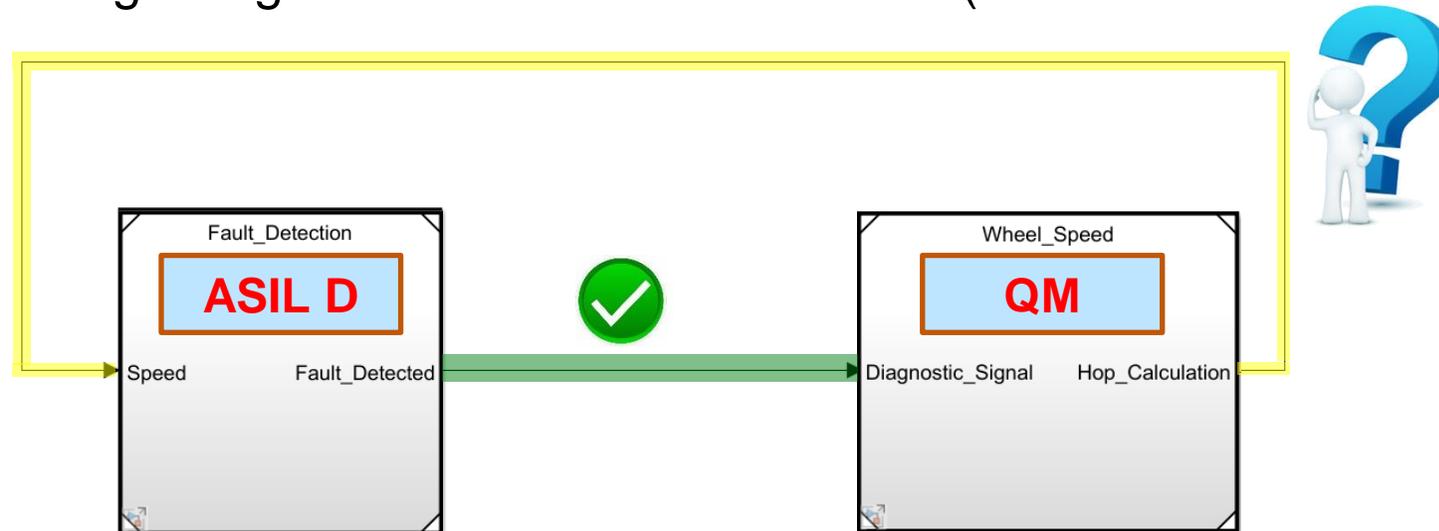


The screenshot shows the Test Manager interface. The 'TESTS' window is open, displaying a list of test scenarios under 'LaneFollowingTestScenarios'. The selected test is 'ACC\_ISO\_TargetDiscriminationTest'. The 'Test Browser' shows details for this test, including its type (Simulation Test), tags, description, requirements, and system under test (LaneFollowingTestBenchExample). The 'SYSTEM UNDER TEST' section shows the model name and various settings like test harness, simulation settings overrides, parameter overrides, callbacks, inputs, simulation outputs, configuration settings overrides, iterations, logical and temporal assessments, custom criteria, and coverage settings.

### Simulink Test (Test Manager)

## Lack top-down architectural design approach

- “Bottom up” (legacy) vs “top down” (functional safety) approach
- Concepts:
  - Static and Dynamic architecture description (ISO 26262-6:2018 Clause 7.4.5)
  - Criteria for coexistence of elements (ISO 26262-9: 2018 Clause 6)
  - Safety-Oriented analysis (ISO 26262-9:2018 Clause 8)
  - Analysis of dependent failures (ISO 26262-9:2018 Clause 7)
  - Software partitioning using Freedom From Interference (ISO 26262-6:2018 Annex D)



# Lack top-down architectural design approach

## Perform architectural review – Freedom From Interference

- ISO 26262-6 (Annex D)
  - Timing and execution
  - Memory
  - Exchange of information



### Model architecture

- Use model reference for unit-level models
- Pick a strategy for grouping units into features
- Split ASIL and QM levels at the top level of the model
- Eliminate algorithm content at the integration level
- Use model metrics to monitor unit complexity

### Signal routing and definition

- Group bus signals by ASIL, feature, and rate
- Pass only necessary signals to units
- Optimize placement of signal and parameter objects
- Protect data exchanged between ASILs

### Code generation configuration

- Determine a code placement strategy
- Use different name tokens for shared utilities

# Poor tool qualification awareness

- How do you qualify a tool?

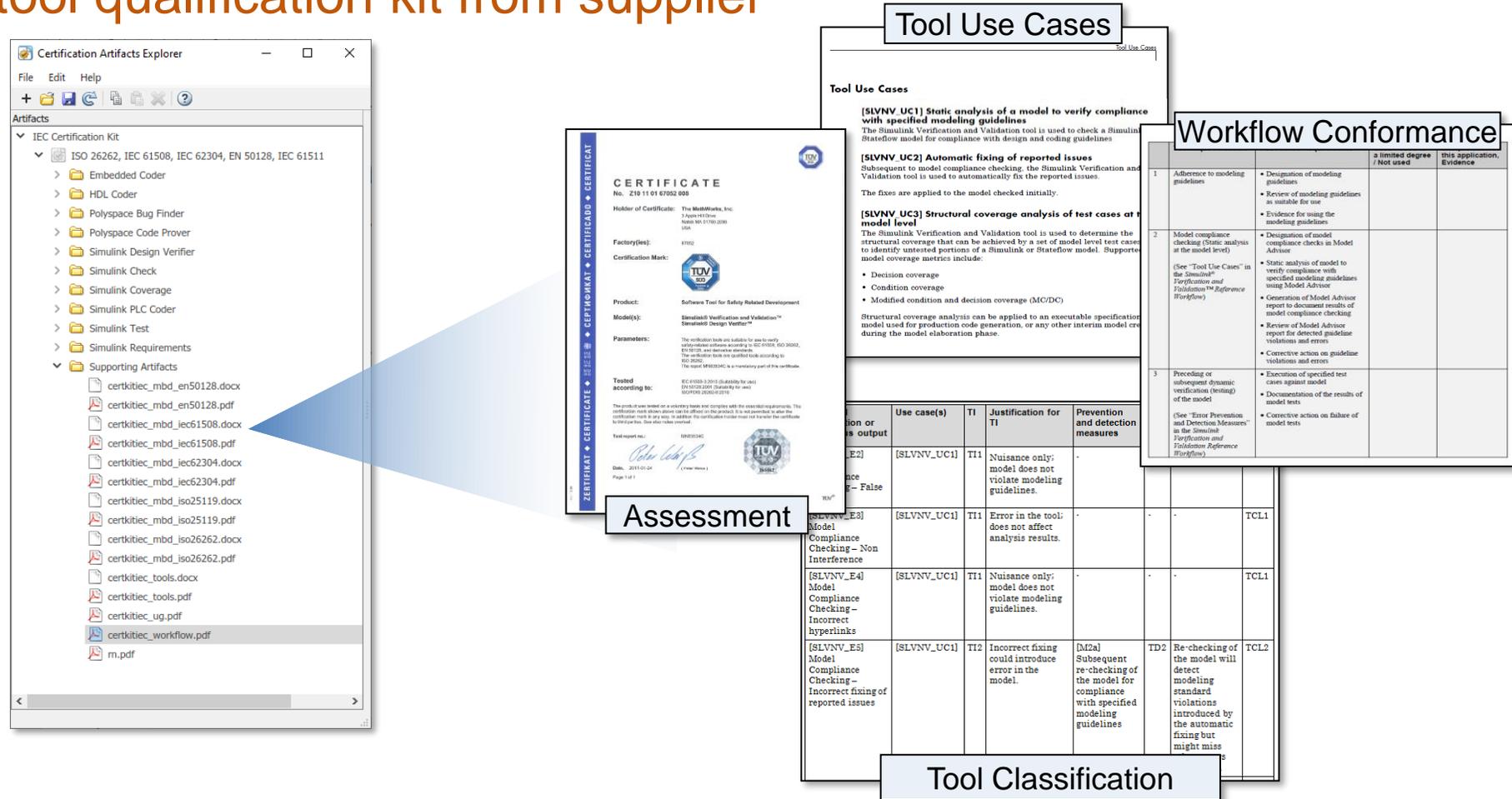
Method	TCL 2				TCL 3			
	ASIL A	ASIL B	ASIL C	ASIL D	ASIL A	ASIL B	ASIL C	ASIL D
1a Increased confidence from use	++	++	++	+	++	++	+	+
1b Evaluation of the tool development process	++	++	++	+	++	++	+	+
1c Validation of the software tool	+	+	+	++	+	+	++	++
1d Development in compliance with a safety standard	+	+	+	++	+	+	++	++

Source: ISO 26262:2018 Clause 11.4.6.1, Tables 4&5

+ ... Recommended    ++ ... Highly recommended

# Poor of tool qualification awareness

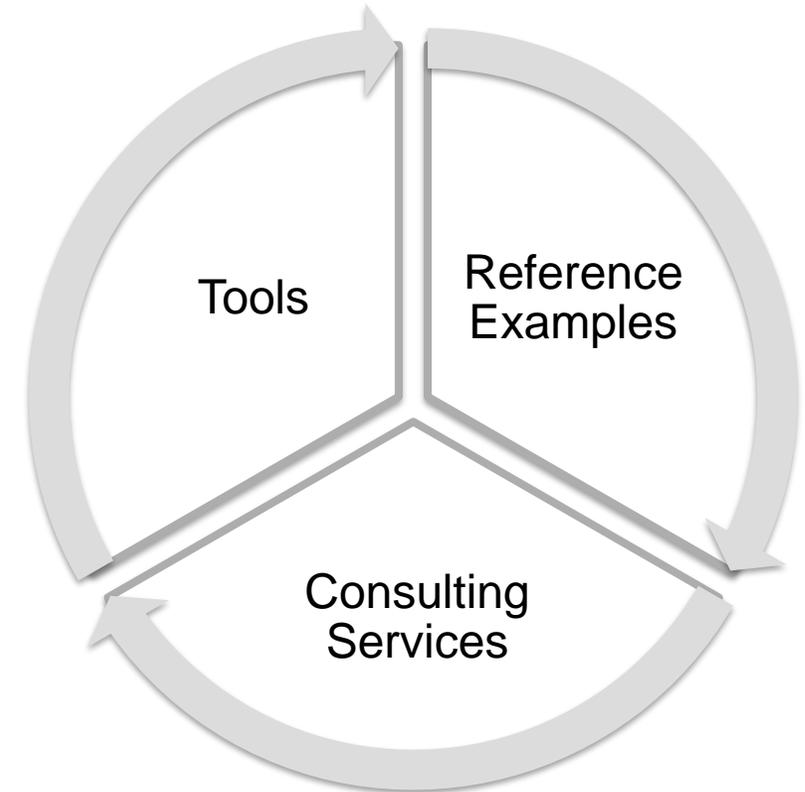
## Leverage tool qualification kit from supplier



- Utilize vendor provided tool qualification content as much as possible
- Have a plan to qualify any custom tools or use cases not covered by the tool vendor

# Summary

- Process not clearly defined or documented
  - Document process: from ISO Requirement down to Detail Work Instructions
- Lack top-down architectural design approach
  - Review architecture – Implementation of Freedom From Interference
- Poor tool qualification awareness
  - Leverage tool qualification content from tool vendor



## Presenter contact info and poll questions

Please contact me at [jasonm@mathworks.com](mailto:jasonm@mathworks.com) with questions

- Poll question : How would you rate your organizations activity on ISO 26262
  - a. No interest
  - b. Some interest but no activity
  - c. Currently implementing an ISO 26262 compliant process
  - d. Struggling to implement an ISO 26262 compliant process
  - e. Already fully ISO 26262 compliant
  
- If you would like to an individual follow-up, please let us know in the WebEx poll area.