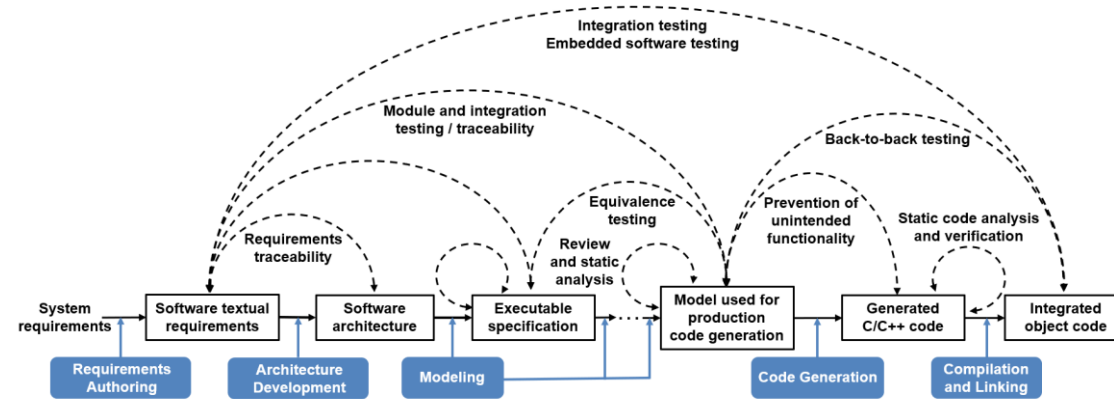


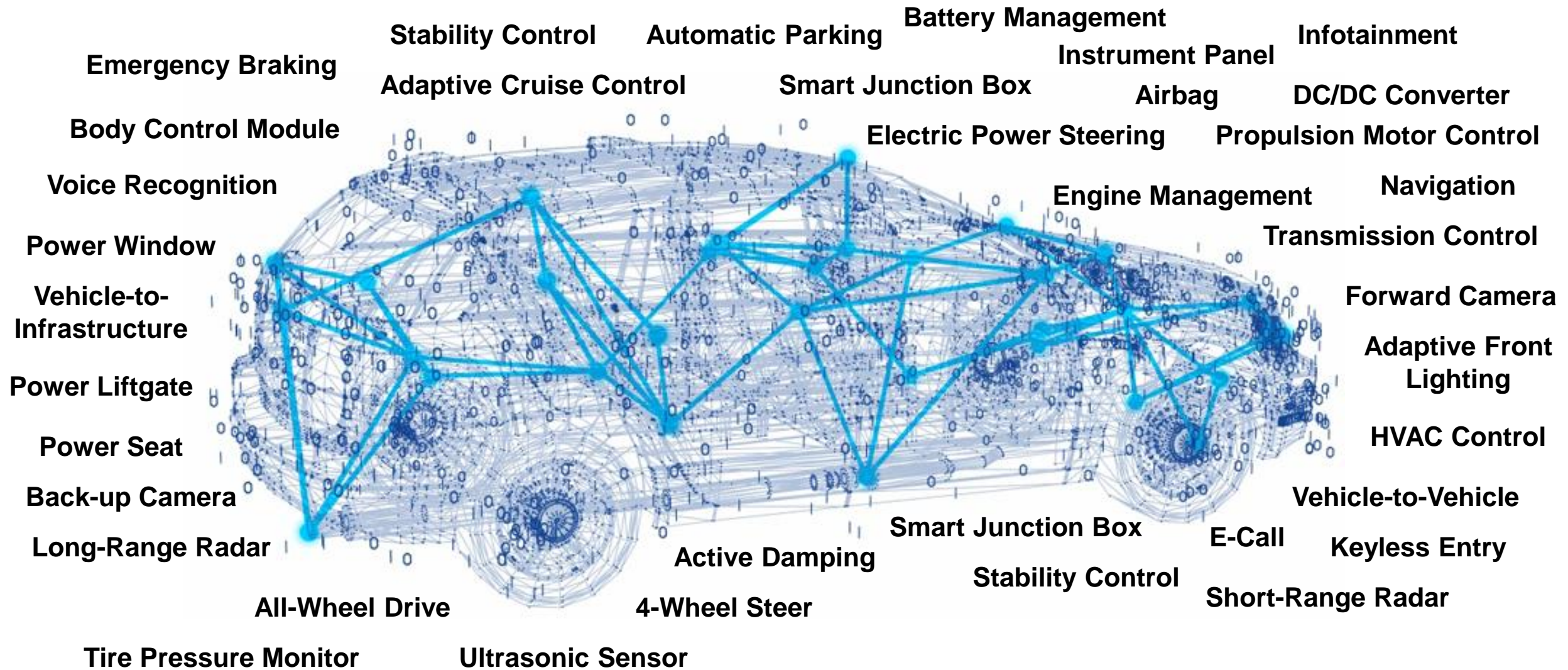
# Meeting ISO26262 using a Robust and Efficient Workflow

**Nukul Sehgal**  
*Application Engineer*  
[nsehgal@mathworks.com](mailto:nsehgal@mathworks.com)



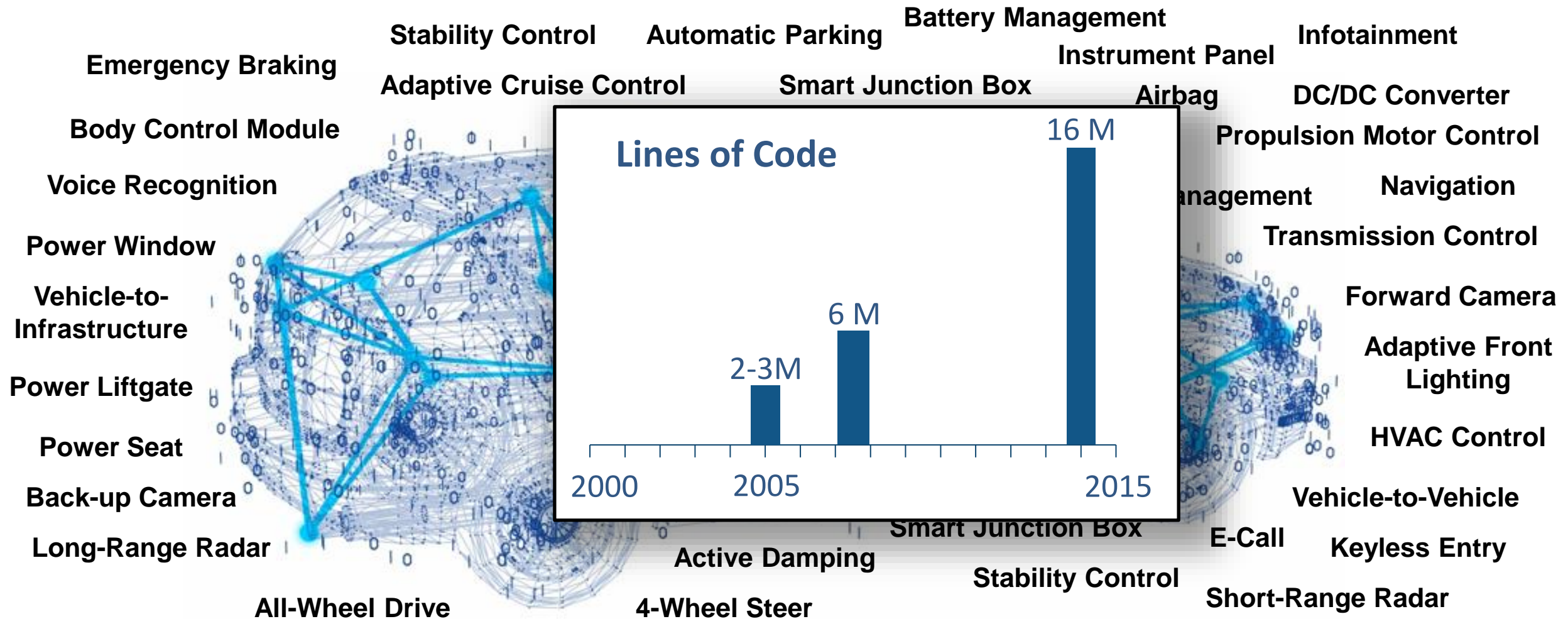
1. Vocabulary		
2. Management of functional safety		
2-5 Overall safety management	2-6 Project dependent safety management	2-7 Safety management regarding production, operation, service and decommissioning
3. Concept phase		
3-5 Item definition	3-6 Hazard analysis and risk assessment	3-7 Functional safety 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
4-8 Safety validation		
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 Hardware design
5-8 Evaluation of the hardware architectural features	5-9 Evaluation of safety analysis violation due to random hardware failures	5-10 Hardware integration and verification
6. Product development at the software level		
6-5 General topics for the product development at the software level	6-6 Specification of software safety requirements	6-7 Software architectural design
6-8 Software unit design and implementation	6-9 Software unit verification	6-10 Software integration and verification
6-11 Testing of the embedded software		
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 development according to ISO 26262
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 analysis		
10. Guideline on ISO 26262		
11. Guideline on application of ISO 26262 to semiconductors		

# Growing Complexity of Controls in the Automotive industry





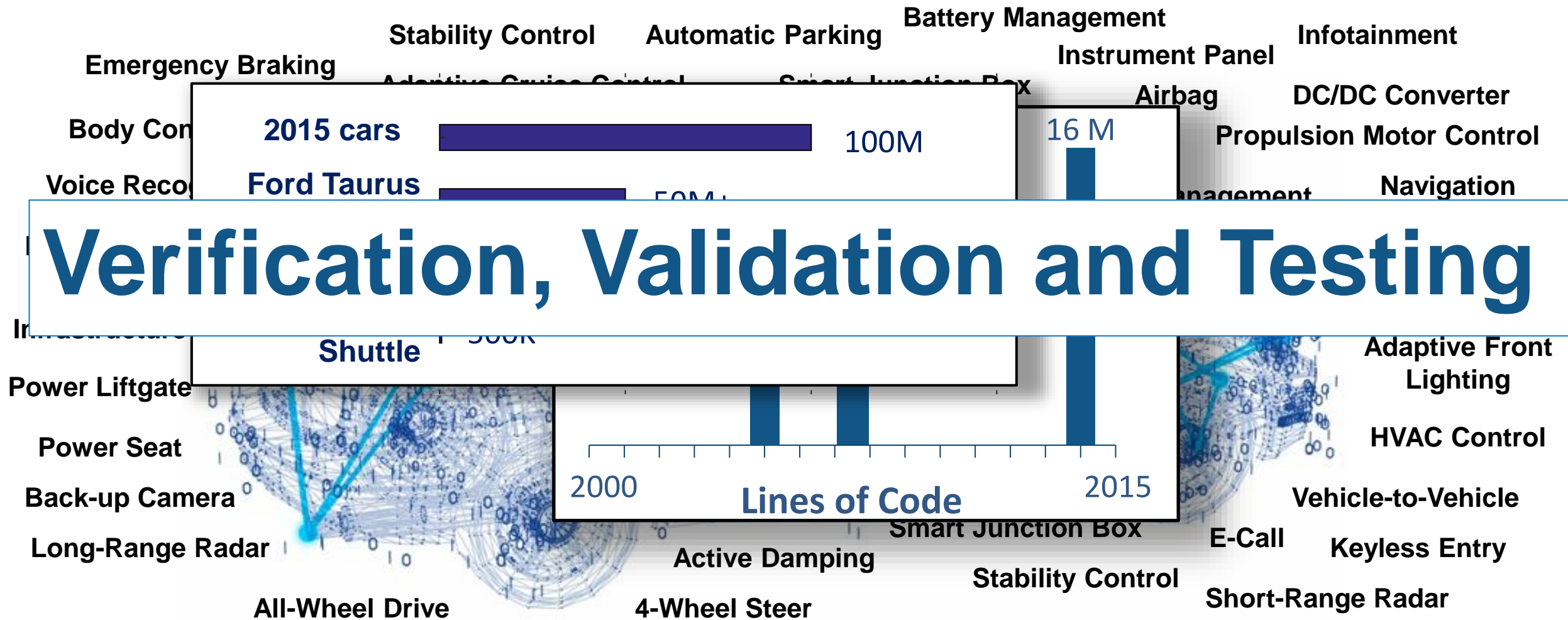
# Growing Complexity of Controls in the Automotive industry



Siemens, "[Ford Motor Company Case Study](#)," Siemens PLM Software, 2014

McKendrick, J. "[Cars become 'datacenters on wheels', carmakers become software companies.](#)" ZDJNet, 2013

# Growing Complexity of Controls in the Automotive industry



Source:

<https://interact.gsa.gov/sites/default/files/J3061%20JP%20presentation.pdf>

# Verification, Validation and Testing

Safety Critical System



Best Practices and Guidelines



Functional Safety Standard



Functional Safety Certification



# IEC 61508 - ISO 26262

IEC 61508 is a functional safety standard for Industrial Automation

Umbrella for industry-specific adaptations:

- ISO 26262 - Automotive / Motorcycle



- ISO 25119 - Agriculture and Forestry



- EN 50128 - Rail



- IEC 62304 - Medical



- IEC 61511 - Process Control



Supported by  
**IEC Certification Kit**

# ISO 26262: 2018 Structure

ISO 26262-1	Vocabulary
ISO 26262-2	Management of functional safety
ISO 26262-3	Concept phase
ISO 26262-4	Product development: system level
ISO 26262-5	Product development: hardware level
ISO 26262-6	Product development: software level
ISO 26262-7	Production, operation, service and decommissioning
ISO 26262-8	Supporting processes
ISO 26262-9	ASIL-oriented and safety-oriented analyses
ISO 26262-10	Guidelines on ISO 26262
ISO 26262-11	Guidelines on application of ISO 26262 to semiconductors
ISO 26262-12	Adaptation of ISO 26262 for motorcycles

# ISO 26262: Model-Based Design in Part 6 and Part 8

ISO 26262-1	Vocabulary	
ISO 26262-2	Management of functional safety	
ISO 26262-3	Concept phase	
ISO 26262-4	Product development: system level	
ISO 26262-5	Product development: hardware level	
ISO 26262-6	Product development: software level	{ Model-Based Design ❖ Development ❖ Verification & Validation ❖ Code generation
ISO 26262-7	Production, operation, service and decommissioning	
ISO 26262-8	Supporting processes	
ISO 26262-9	ASIL-oriented and safety-oriented analyses	{ Tool classification and qualification
ISO 26262-10	Guidelines on ISO 26262	
ISO 26262-11	Guidelines on application of ISO 26262 to semiconductors	
ISO 26262-12	Adaptation of ISO 26262 for motorcycles	



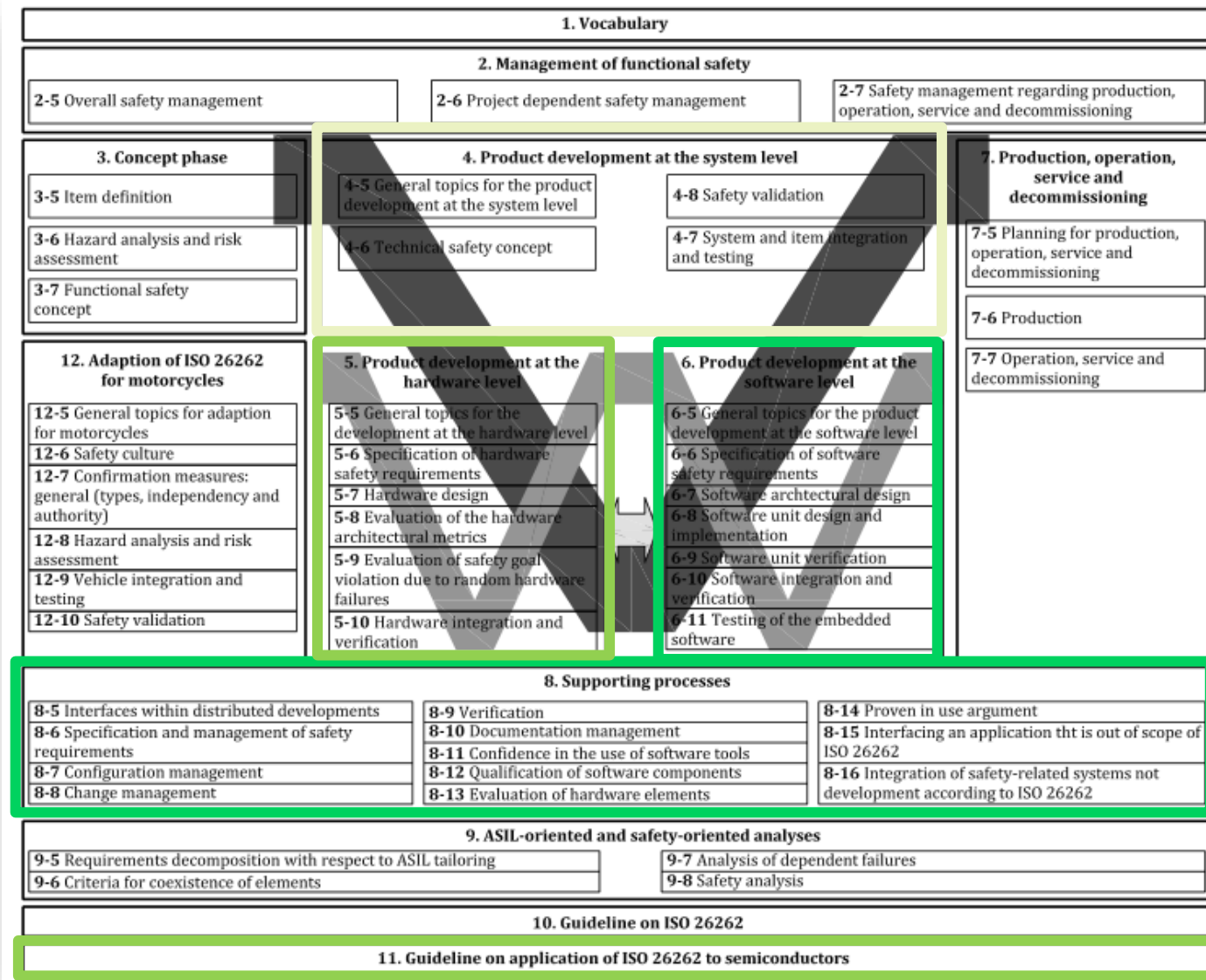
# ISO 26262: Model-Based Design in Part 4

ISO 26262-1	Vocabulary	
ISO 26262-2	Management of functional safety	
ISO 26262-3	Concept phase	
ISO 26262-4	Product development: system level	{ Model-Based Systems Engineering
ISO 26262-5	Product development: hardware level	
ISO 26262-6	Product development: software level	{ Model-Based Design ❖ Development ❖ Verification & Validation ❖ Code generation
ISO 26262-7	Production, operation, service and decommissioning	
ISO 26262-8	Supporting processes	{ Tool classification and qualification
ISO 26262-9	ASIL-oriented and safety-oriented analyses	
ISO 26262-10	Guidelines on ISO 26262	
ISO 26262-11	Guidelines on application of ISO 26262 to semiconductors	
ISO 26262-12	Adaptation of ISO 26262 for motorcycles	

# ISO 26262: Model-Based Design in Part 5 and Part 11

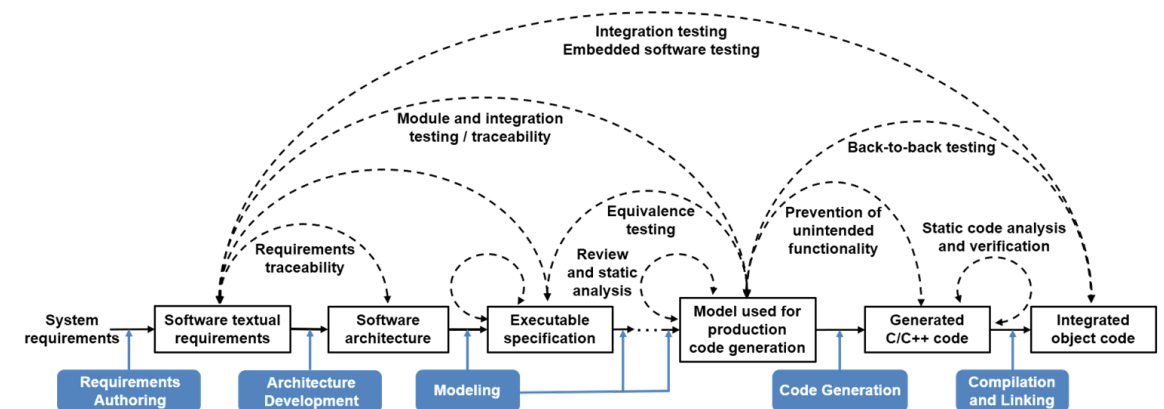
ISO 26262-1	Vocabulary	
ISO 26262-2	Management of functional safety	
ISO 26262-3	Concept phase	
ISO 26262-4	Product development: system level	{ Model-Based Systems Engineering
ISO 26262-5	Product development: hardware level	
ISO 26262-6	Product development: software level	{ Design of hardware system Model-Based Design ❖ Development ❖ Verification & Validation ❖ Code generation
ISO 26262-7	Production, operation, service and decommissioning	
ISO 26262-8	Supporting processes	
ISO 26262-9	ASIL-oriented and safety-oriented analyses	{ Tool classification and qualification
ISO 26262-10	Guidelines on ISO 26262	
ISO 26262-11	Guidelines on application of ISO 26262 to semiconductors	{ HDL Code generation
ISO 26262-12	Adaptation of ISO 26262 for motorcycles	

# ISO 26262 Overview



# ISO 26262 Part 6 – IEC Certification Kit and Reference Workflow

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	4-5 General topics for the product development at the system level	4-8 Safety validation
3-6 Hazard analysis and risk assessment	4-6 Technical safety concept	4-7 System and item integration and testing
3-7 Functional safety concept		
4. Product development at the system level		
		7-5 Planning for production, operation, service and decommissioning
		7-6 Production
		7-7 Operation, service and decommissioning
5. Product development at the hardware level		
5-5 General topics for the development at the hardware level	6-5 General topics for the product development at the software level	
5-6 Specification of hardware safety requirements	6-6 Specification of software safety requirements	
5-7 Hardware design	6-7 Software architectural design	
5-8 Evaluation of the hardware architectural metrics	6-8 Software unit design and implementation	
5-9 Evaluation of safety goal violation due to random hardware failures	6-9 Software unit verification	
5-10 Hardware integration and verification	6-10 Software integration and verification	
	6-11 Testing of the embedded software	
6. Product development at the software level		
7. Production, operation, service and decommissioning		
8. Supporting processes		
8-5 Interfaces within distributed developments	8-9 Verification	8-14 Proven in use argument
8-6 Specification and management of safety requirements	8-10 Documentation management	8-15 Interfacing an application that is out of scope of ISO 26262
8-7 Configuration management	8-11 Confidence in the use of software tools	8-16 Integration of safety-related systems not development according to ISO 26262
8-8 Change management	8-12 Qualification of software components	
	8-13 Evaluation of hardware elements	
9. ASIL-oriented and safety-oriented analyses		
9-5 Requirements decomposition with respect to ASIL tailoring	9-7 Analysis of dependent failures	
9-6 Criteria for coexistence of elements	9-8 Safety analysis	
10. Guideline on ISO 26262		
11. Guideline on application of ISO 26262 to semiconductors		





# ISO 26262 Part 6

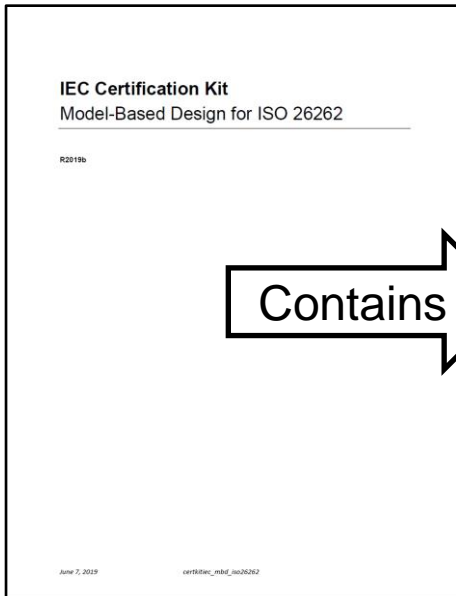
⊕	Highly recommended
+	Recommended
○	No recommendation

Table 1 – Topics To Be Covered By Modeling and Coding Guidelines					
Topics		ASIL			
		A	B	C	D
1i	Concurrency aspects	+	+	+	+
1h	Use of naming conventions	⊕	⊕	⊕	⊕
1g	Use of style guides	+	⊕	⊕	⊕
1f	Use of unambiguous graphical representation	+	⊕	⊕	⊕
1e	Use of well-trusted design principles	+	+	+	⊕
1d	Use of defensive implementation techniques	+	+	⊕	⊕
1c	Enforcement of strong typing	⊕	⊕	⊕	⊕
1b	Use of language subsets	⊕	⊕	⊕	⊕
1a	Enforcement of low complexity	⊕	⊕	⊕	⊕

Topics																
	Tables															
1o																
1n																
1m																
1l																
1k																
1j																
1i																
1h																
1g																
1f																
1e																
1d																
1c																
1b																
1a																

# ISO 26262 Part 6 – IEC Certification Kit Documentation

‡ Highly recommended  
 + Recommended  
 ○ No recommendation



**Table 1 – Topics to be Covered by Modelling and Coding Guidelines**

Topics		ASIL				Applicable Model-Based Design Tools and Processes	Comments
		A	B	C	D		
1a	Enforcement of low complexity	++	++	++	++	Simulink® – Modeling Guidelines	The High Integrity System Modeling Guidelines and the MathWorks® Automotive Advisory Board – Control Algorithm Modeling Guidelines as well as applicable coding standards (MISRA C®:2004, MISRA C:2012, MISRA C:2012 (Amendment 1:2016) MISRA® C++, or JSF®++) can be used to address topics listed in this table. The guideline subset used for a project should address a combination of topics applicable for the ASIL under
1b	Use of language subsets	++	++	++	++	Simulink Check	
1c	Enforcement of strong typing	++	++	++	++	Polyspace® Bug Finder™ and Polyspace Bug Finder™ Server™ – Coding Rules Checks	
1d	Use of defensive implementation techniques	+	+	++	++		

**Topics**

**ASIL**

**Tables**

1m

1l

1k

1j

1i

1h

1g

1f

1e

1d

1c

1b

1a

Table 1

Table 2

Table 3

Table 4

Table 5

Table 6

Table 7

Table 8

Table 9

Table 10

Table 11

Table 12

Table 13

Table 14

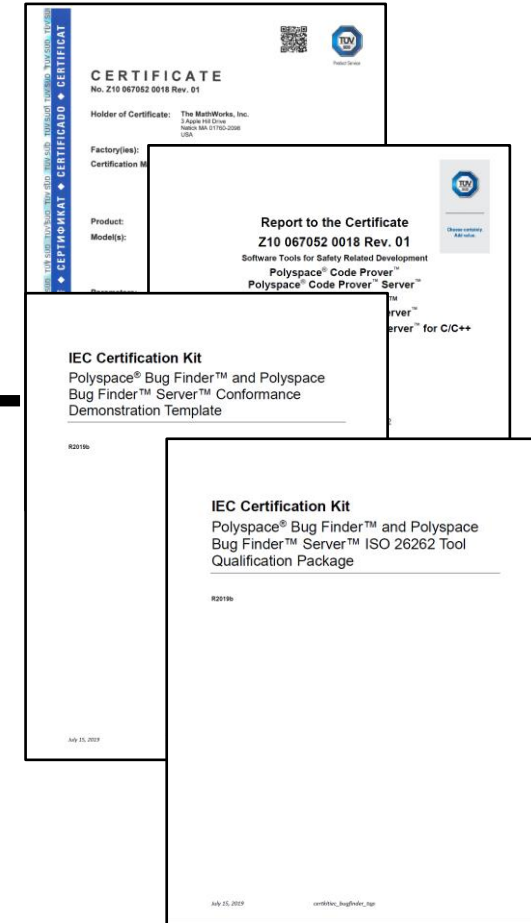
Table 15

‡ Highly recommended  
 + Recommended  
 ○ No recommendation

Annotated method tables with suggestions on how to use Model-Based Design processes and tools to apply the methods listed in ISO 26262-6



<b>Topics</b>		<b>ASIL</b>				<b>Applicable Model-Based Design Tools and Processes</b>	<b>Comments</b>
		<b>A</b>	<b>B</b>	<b>C</b>	<b>D</b>		
1a	Enforcement of low complexity	++	++	++	++	Simulink® – Modeling Guidelines	The High Integrity System Modeling Guidelines and the MathWorks® Automotive Advisory Board – Control Algorithm Modeling Guidelines as well as applicable coding standards (MISRA C®:2004, MISRA C:2012, MISRA C:2012 (Amendment 1:2016) MISRA® C++, or JSF®++) can be used to address topics listed in this table. The guideline subset used for a project should address a combination of topics applicable for the ASIL under
1b	Use of language subsets	++	++	++	++	Simulink Check	
1c	Enforcement of strong typing	++	++	++	++	Polyspace® Bug Finder™ and Polyspace® Bug Finder™ Server™ –	
1d	Use of defensive implementation techniques	+	+	++	++	Coding Rules Checks	

[illegible]

# ISO 26262 - Reference Workflow

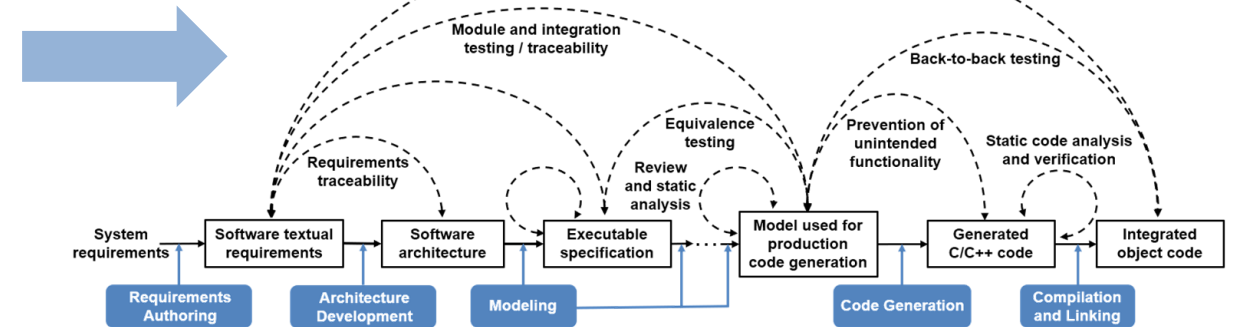
Table 1 — Topics to be Covered by Modelling and Coding Guidelines

Table 8 — Methods for Deriving Test Cases for Software Unit Testing

Table 9 — Structural Coverage Metrics at the Software Unit Level

Table 11 — Methods for Deriving Test Cases for Software Integration Testing

Methods	ASIL				Applicable Model-Based Design Tools and Processes	Comments
	A	B	C	D		
1a Analysis of requirements	++	++	++	++	Simulink Test	Simulink Test can be used to establish bidirectional links between textual requirements and test cases.
					Simulink Requirements	Simulink Requirements can be used to analyze the coverage or requirements by tests.
					Simulink Signal Builder block	The Signal Builder block can be used to create requirements-based model tests. Requirements pane in the Signal Builder block can be used to link tests with textual requirements.
1b Generation and analysis of equivalence classes	+	++	++	++	Simulink Design Verifier – Test case generation	The analysis of equivalence classes can be based on the interfaces of the model. Automatic test case generation in combination with Test Objective blocks can be used to generate test cases and test sequences for given equivalence classes.
1c Analysis of boundary values	+	++	++	++	Simulink Design Verifier – Test case generation	The analysis of boundary values can be based on the interfaces of the model. Automatic test case generation in combination with Test Objective blocks can be used to generate test cases and test sequences for given boundary values.
1d Error guessing based on knowledge or experience	+	+	+	+		





# ISO 26262 Part 6: 2018

## Simulink and Stateflow as Suitable for Software Architecture, Design and as basis for Code Generation

**Table 5 — Notations for software unit design**

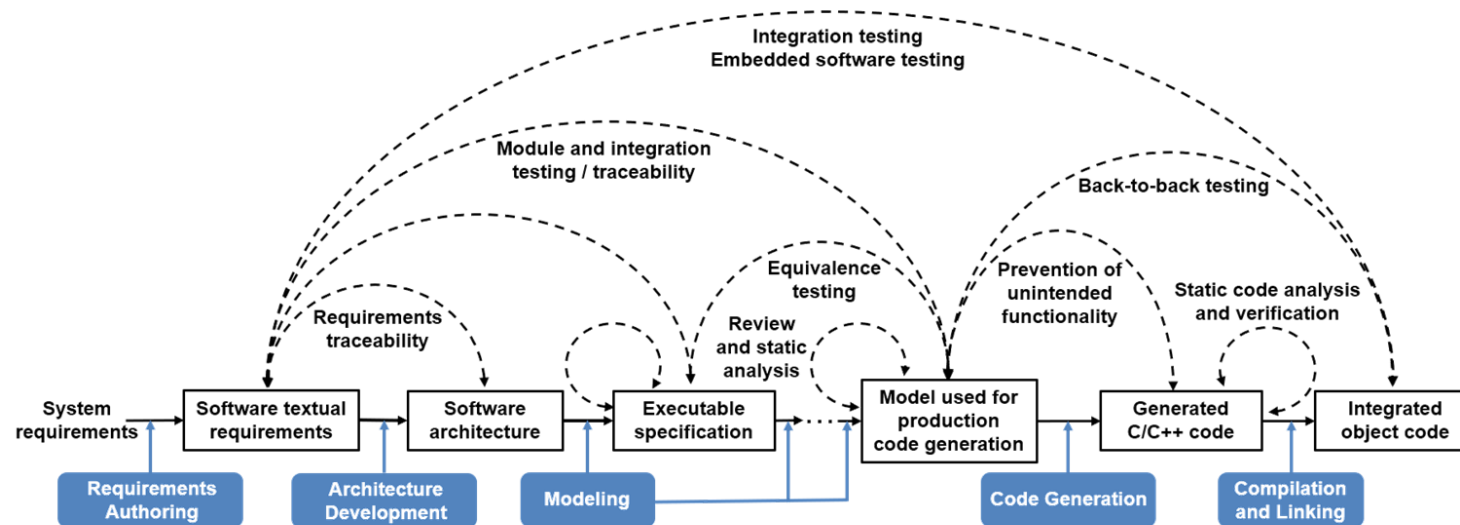
Notations		ASIL			
		A	B	C	D
1a	Natural language <sup>a</sup>	++	++	++	++
1b	Informal notations	++	++	+	+
1c	Semi-formal notations <sup>b</sup>	+	+	++	++
1d	Formal notations	+	+	+	+
<p><sup>a</sup> Natural language can complement the use of notations for example where some topics are more readily expressed in natural language or provide an explanation and rationale for decisions captured in the notations.</p> <p>EXAMPLE To avoid possible ambiguity of natural language when designing complex elements, a combination of an activity diagram with natural language can be used.</p> <p><sup>b</sup> Semi-formal notations can include pseudocode or modelling with UML®, SysML®, Simulink® or Stateflow®.</p> <p>NOTE UML®, SysML®, Simulink® and Stateflow® are examples of suitable products available commercially. This information is given for the convenience of users of this document and does not constitute an endorsement by ISO of these products.</p>					

**NOTE** In the case of model-based development with automatic code generation, the methods for representing the software unit design are applied to the model which serves as the basis for the code generation.

*Table 2 Software Architecture Design Notations has similar suitability wording for use of Simulink and Stateflow*

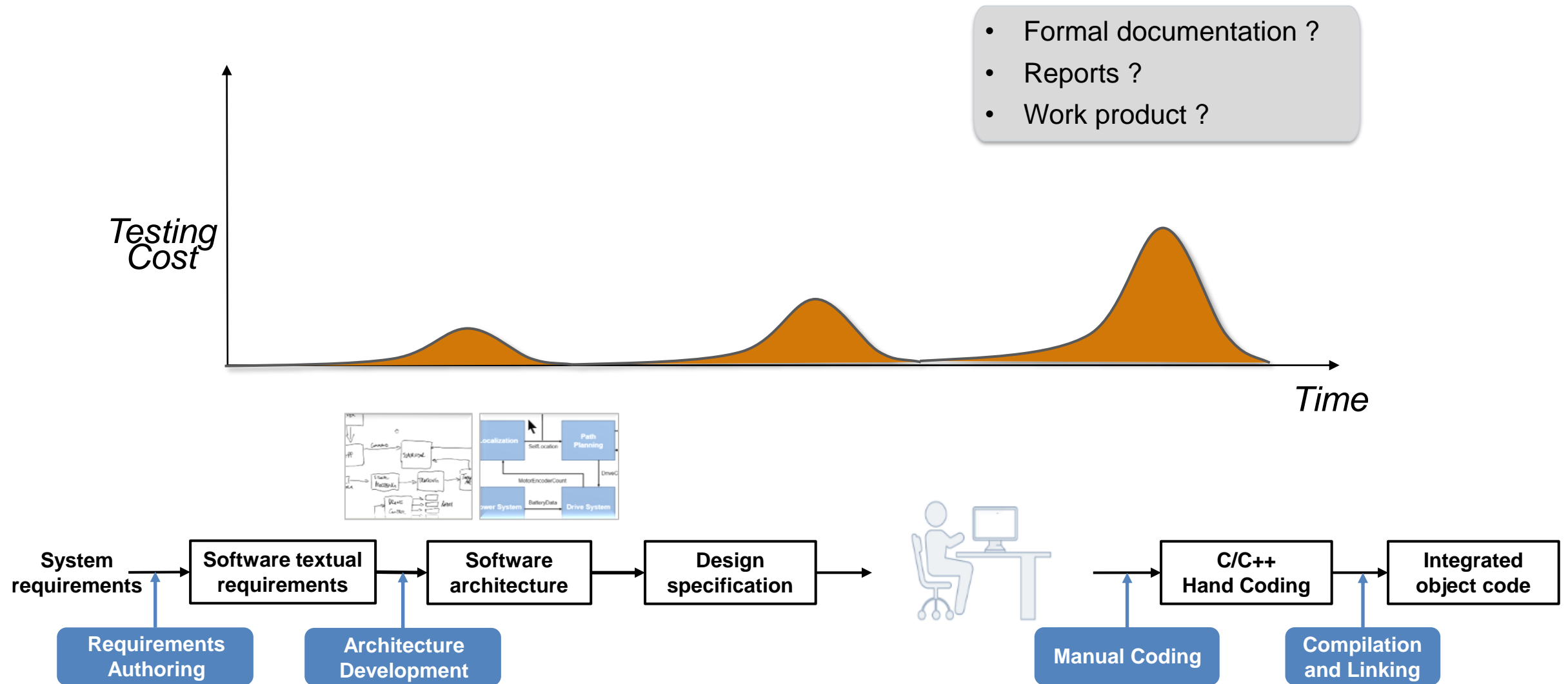
# Structure of the IEC Certification Kit

1. **Reference Workflow** for systematic **Verification and Validation (V&V)** of models and generated code
2. **Tool certification/qualification** accomplished by tool test suites, vendor audits, and reference workflow

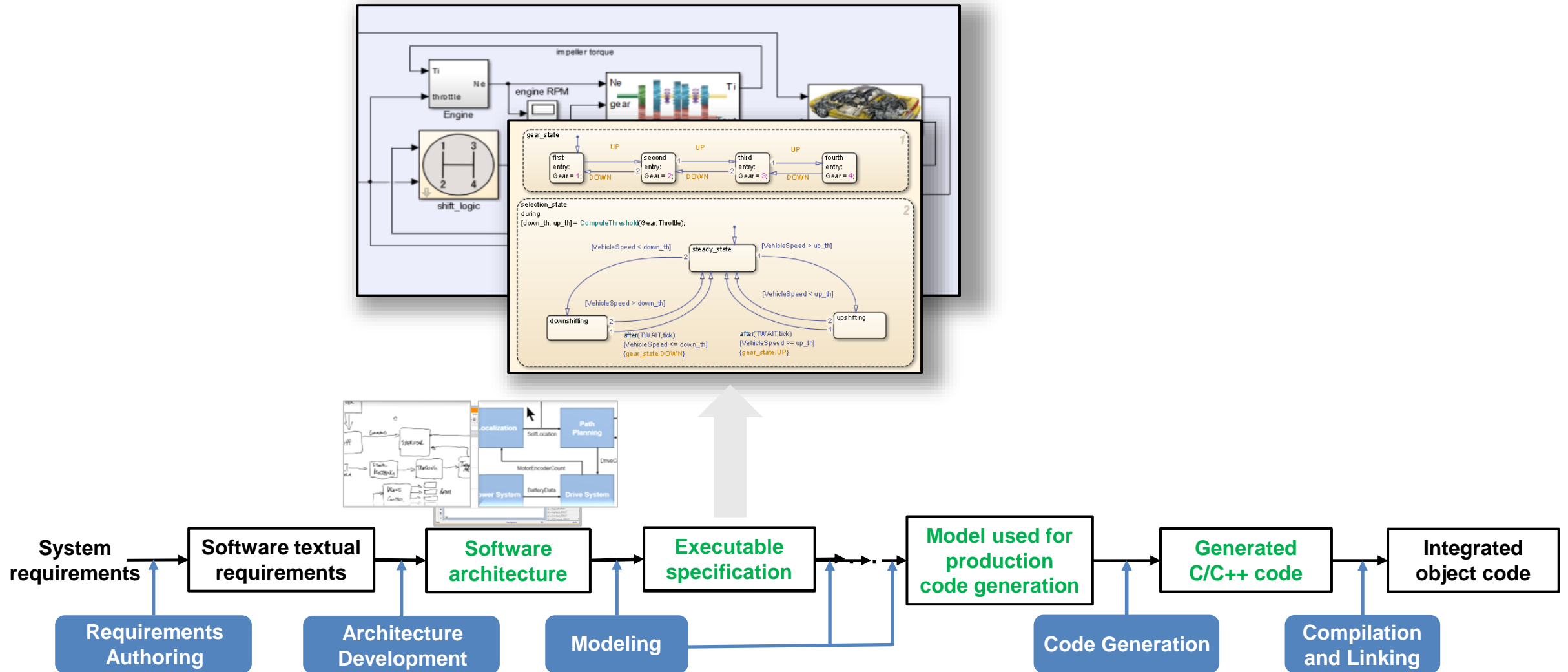


Tool Use Cases									
CERTIFICATE									
<p>No. Z10 11 01 67052 008</p> <p>Holder of Certificate: The MathWorks, Inc. 3 Apple Hill Drive Natick MA 01909-2099 USA</p> <p>Factory(ies): 67052</p> <p>Certification Mark: </p> <p>Product: Software Tool for Safety Related Development</p> <p>Model(s): Simulink® Verification and Validation™ Simulink® Design Verifier™</p> <p>Parameters: The verification tools are suitable for use to verify safety-related software according to ISO 9126, ISO 26262, EN 15120, and derivative standards. The verification tools are qualified code according to ISO 9001.</p> <p>The report M902042C is a mandatory part of this certificate.</p> <p>Tested according to: IEC 61508-3:2010 (suitability for use) EN 15120:2007 (suitability for use) ISO 9001:2008 (2010)</p> <p>The product was tested on a voluntary basis and complies with the essential requirements. This certificate mark shows where can be relied on the product. It is not permitted to alter the certificate mark in any way. In addition the certification holder must not transfer the certificate to third parties. See also rules control.</p> <p>Test report no.: M902042C</p> <p>Date: 2011-01-26 (Peter Weiss)</p> <p>Page 1 of 1</p> <p>TUV SUD Product Service GmbH · Zentralsiedlung · Industriestraße 10 · 80333 München · Germany</p>									
Tool Use Cases									
<p><b>[SLVNV_UC1] Static analysis of a model to verify compliance with specified modeling guidelines</b> The Simulink Verification and Validation tool is used to check a Simulink or Stateflow model for compliance with design and coding guidelines.</p> <p><b>[SLVNV_UC2] Automatic fixing of reported issues</b> Subsequent to model compliance checking, the Simulink Verification and Validation tool is used to automatically fix the reported issues. The fixes are applied to the model checked initially.</p> <p><b>[SLVNV_UC3] Structural coverage analysis of test cases at the model level</b> The Simulink Verification and Validation tool is used to determine the structural coverage that can be achieved by a set of model level test cases or to identify untested portions of a Simulink or Stateflow model. Supported model coverage metrics include:</p> <ul style="list-style-type: none"> <li>Decision coverage</li> <li>Condition coverage</li> <li>Modified condition and decision coverage (MC/DC)</li> </ul> <p>Structural coverage analysis can be applied to an executable specification, a model used for production code generation, or an other internal model created.</p>									
Initial function or output	Use case(s)	TI	Justification for TI	Prevention and detection measures	TD	Justification for TD	TCL		
NV_E2 Nuisance sing - False ive	[SLVNV_UC1]	TI1	Nuisance only: model does not violate modeling guidelines.	-	-	-	TCL1		
NV_E3 Nuisance sing - Non Source	[SLVNV_UC1]	TI1	Error in the tool: does not affect analysis results.	-	-	-	TCL1		
NV_E4 Nuisance sing - pect risks	[SLVNV_UC1]	TI1	Nuisance only: model does not violate modeling guidelines.	-	-	-	TCL1		
NV_E5 Compliance Checking - Incorrect fixing of reported issues	[SLVNV_UC1]	TI2	Incorrect fixing could introduce error in the model.	TD2a Subsequent re-checking of the model for compliance with specified modeling guidelines	TD2	Re-checking of the model will detect modeling standard violations introduced by the automatic fixing but might miss other errors introduced.	TCL2		

# Traditional Development Process



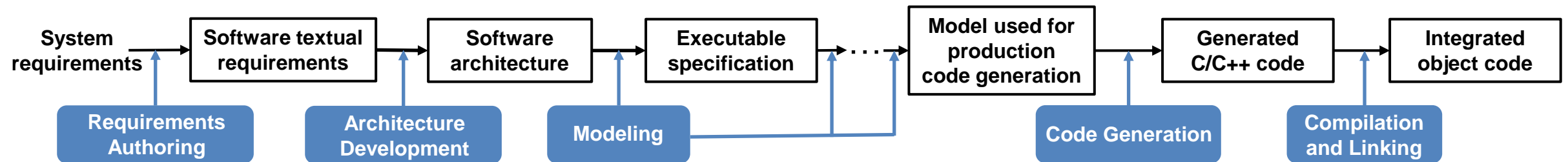
# Model-Based Design Workflow





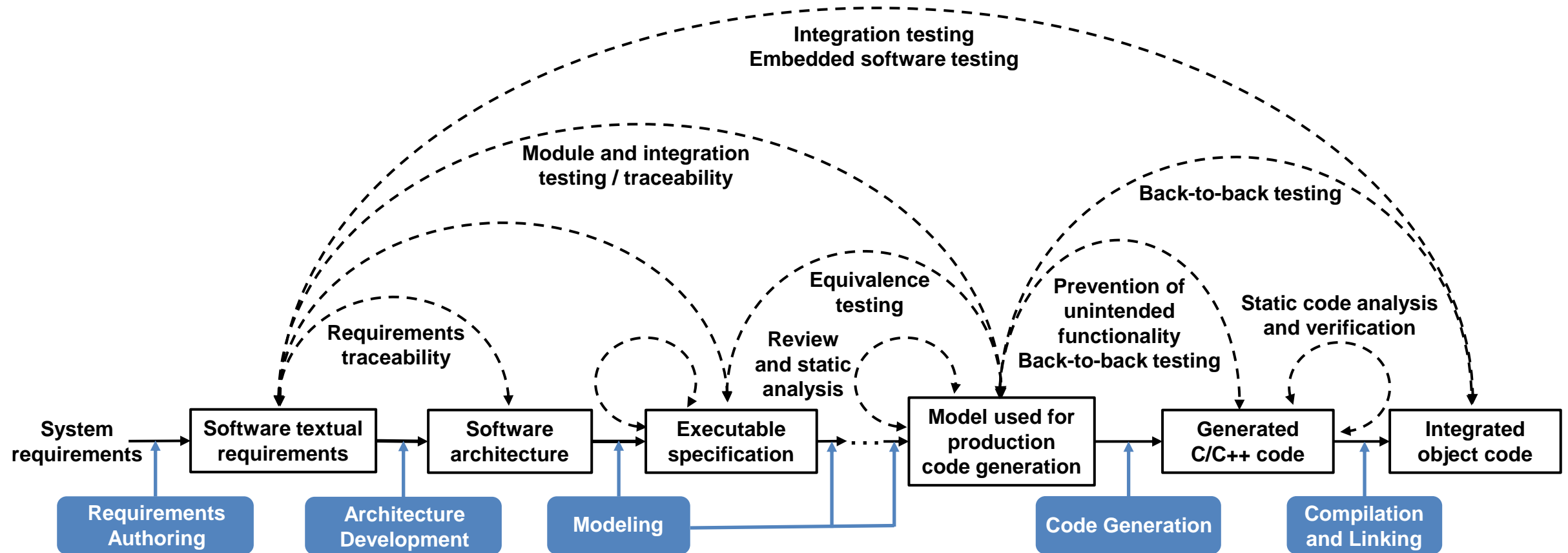
# Model-Based Design Workflow

**get the complete confidence in your design**



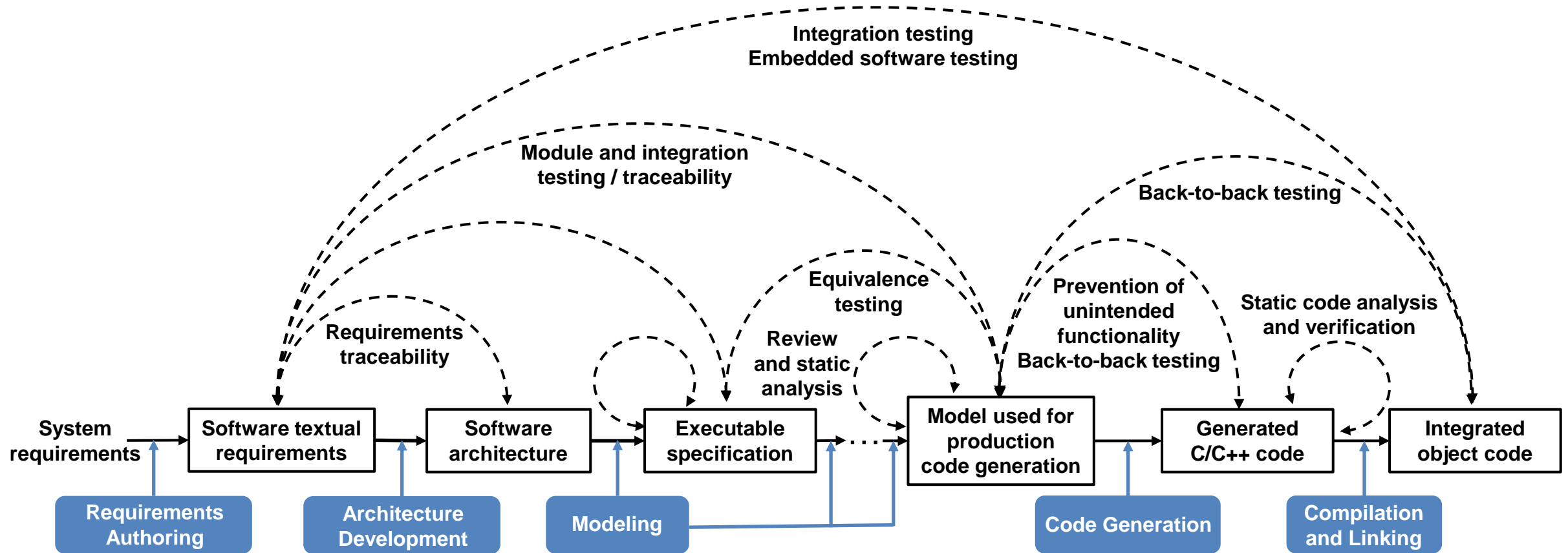
# Complete Model-Based Design Workflow

get the complete confidence in your design



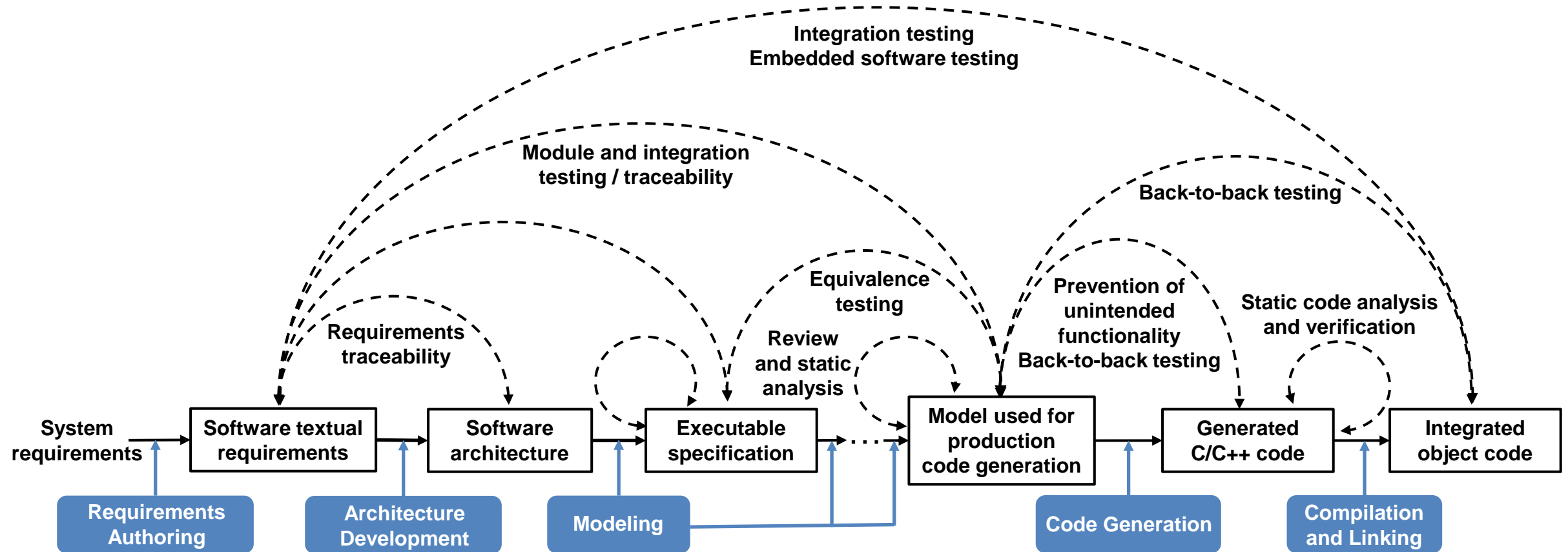
# Complete Model-Based Design Workflow for ISO 26262

- Certifiable Model-Based Design Workflow to develop critical embedded software
- Reviewed and approved by TÜV SÜD certification authority
- Detailed workflow documented in MathWorks *IEC Certification Kit*



# Reference Workflow

- Certifiable Model-Based Design Workflow to develop critical embedded software
- Reviewed and approved by TÜV SÜD certification authority
- Detailed workflow documented in MathWorks *IEC Certification Kit*





# KOSTAL Achieves ISO 26262 ASIL D Certification using Embedded Coder

## Challenge

Develop automotive electronic steering column lock software and certify it to the highest-level functional safety standard

## Solution

Use Model-Based Design to design, implement, and verify the application software via back-to-back PIL testing required for ISO 26262 ASIL D certification

## Results

- Development and certification time cut by 30%
- 80% of errors identified in modeling phase
- PIL test framework for ISO 26262 established

[Link to user story](#)



Kostal's electronic steering column lock module

*“Using Model-Based Design to design, implement, and verify our software for the highest functional safety standard enabled our team to save costs, increase efficiency, and ensure software quality. Without Model-Based Design, more engineers would be needed to complete the project in the same time frame.”*

– Cheng Hui, KOSTAL

# LG Chem Develops AUTOSAR - and ISO 26262 - Compliant Software of a Battery Management System (BMS) for a Hybrid Vehicle

## Challenge

Design and implement production battery management system (BMS) software for the Volvo XC90 plug-in hybrid

## Solution

Use Model-Based Design with MATLAB and Simulink to model, simulate, verify, and generate production code for AUTOSAR application layer software components

## Results

- Existing library of core components reused
- Software issues reduced by more than 50%
- ISO 26262 ASIL C certification achieved

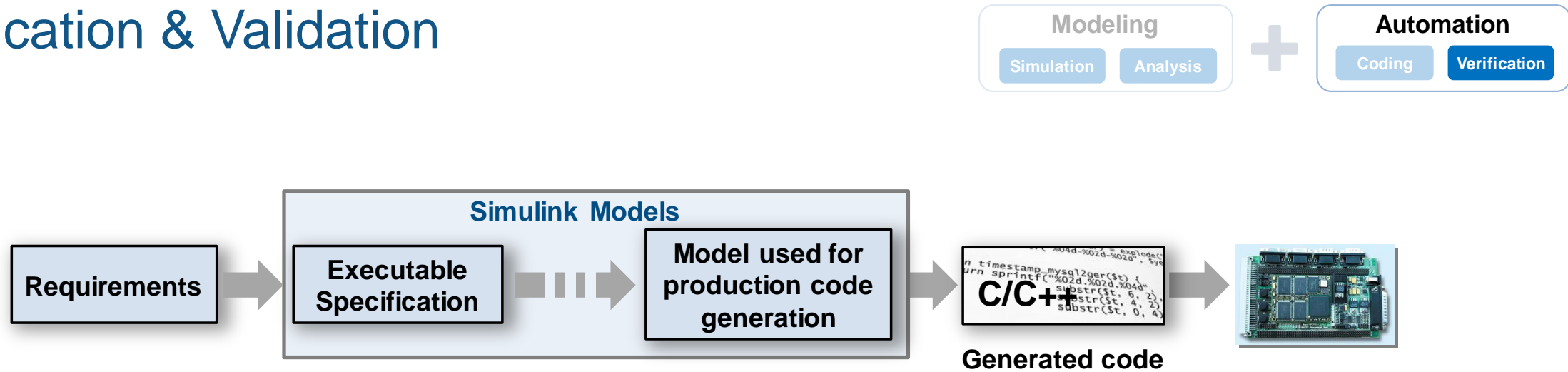


The LG Chem battery management system.

*“Model-Based Design with MATLAB and Simulink enables us to increase component reuse, reduce manual coding, improve communication with our customers, and ultimately deliver higher-quality BMS in less time.”*

*- Won Tae Joe, LG Chem*

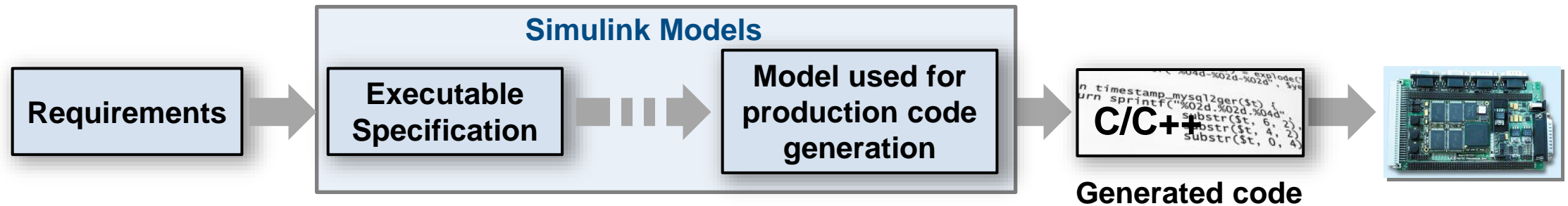
# Verification & Validation



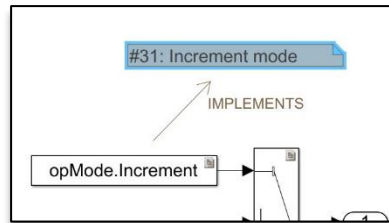
- Author, manage requirements in Simulink
- Early verification to find defects sooner
- Automate manual verification tasks
- Workflow that conforms to safety standards

*“Reduce costs and project risk through early verification, shorten time to market on a certified system, and deliver high-quality production code that was first-time right”*  
 Michael Schwarz, ITK Engineering

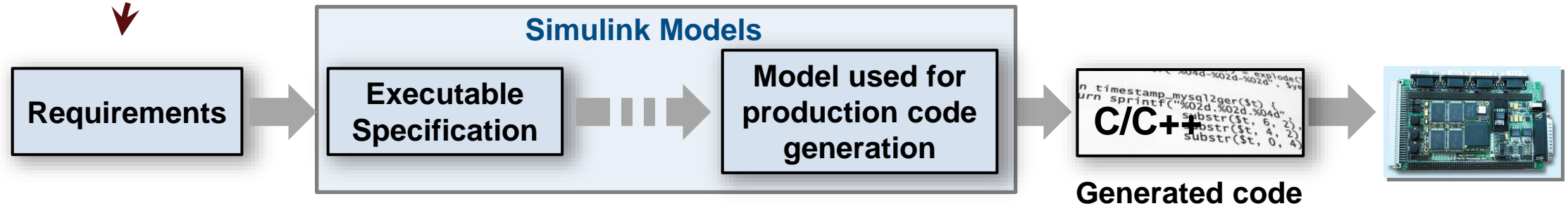
# High Integrity Verification Workflow



# High Integrity Verification Workflow



**Requirements Traceability**  
Prevent unintended design behavior



**Product Name:** Simulink Requirements

# High Integrity Verification Workflow

Modeling

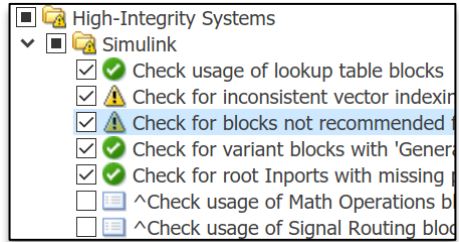
Simulation

Analysis

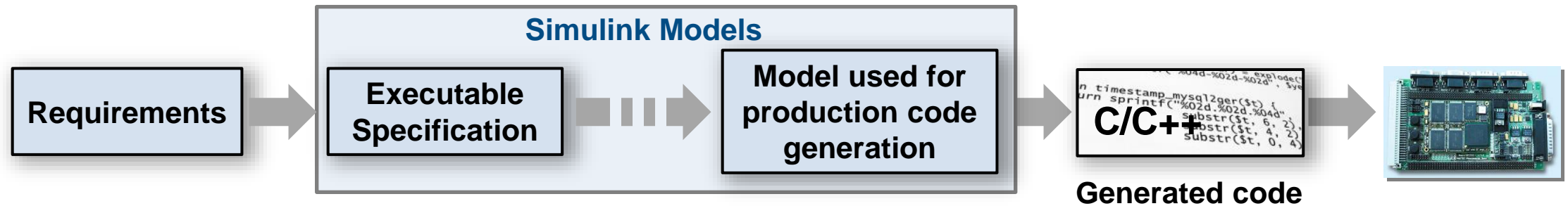
Automation

Coding

Verification



**Standards Compliance with Review and Static Analysis**  
Confirm design meets standard guidelines



**Product Name:** Simulink Check



# High Integrity Verification Workflow

Modeling

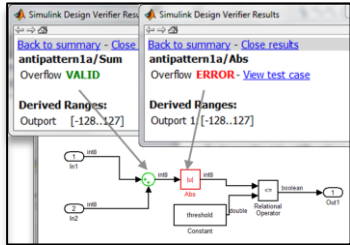
Simulation

Analysis

Automation

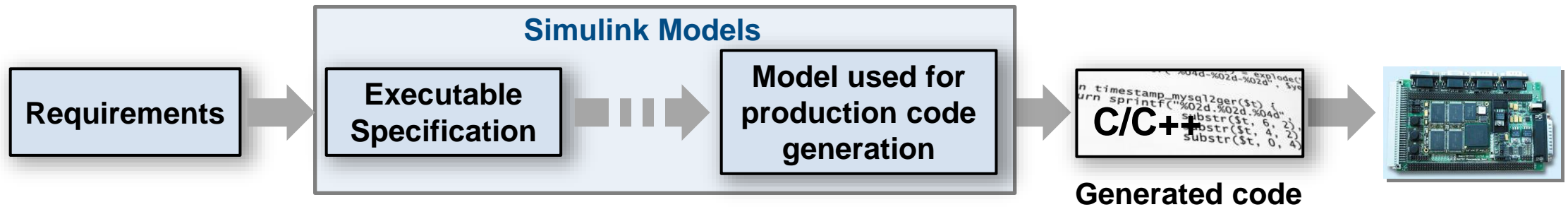
Coding

Verification



## Formal Verification

Prove that the design is robust and meets requirements



**Product Name:** Simulink Design Verifier

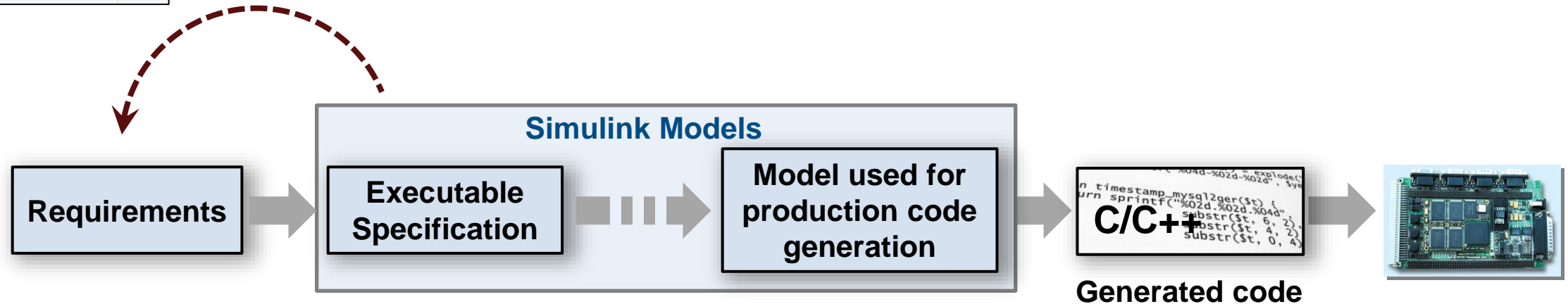
# High Integrity Verification Workflow



Unit test for DriverSwRequest	6	1
Enable button	✓	
Cancel button		✗
Sim Output (db_DriverSwf		
Custom Criteria Result		✗
verifyTrue failed. --> Th		✗

## Component and System Testing

Confirm with testing in simulation (MIL) that design meets requirements



**Product Name:** Simulink Test

**MIL**, Model in the Loop, simulate on host PC

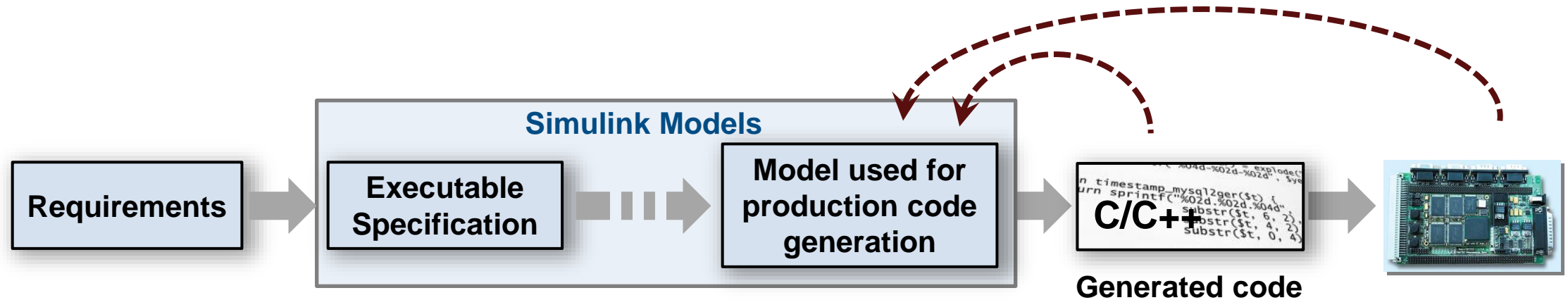
# High Integrity Verification Workflow



Fuel Rate Control Equivalence Test	✓
Equivalence Criteria Result	✓
air_fuel_ratio	✓
ego	✓
fuel	✓
map	✓

## Back to Back Testing

Equivalence checking and testing for SIL and PIL



**Product Name:** Simulink Test

**SIL**, Software in the Loop, execute on host PC  
**PIL**, Processor in the Loop, execute on target processor,  
 or instruction set simulator

# High Integrity Verification Workflow

Modeling

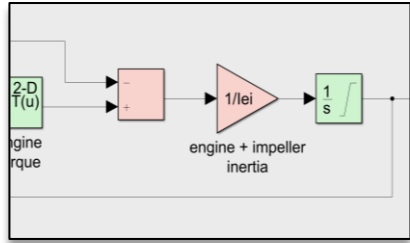
Simulation

Analysis

Automation

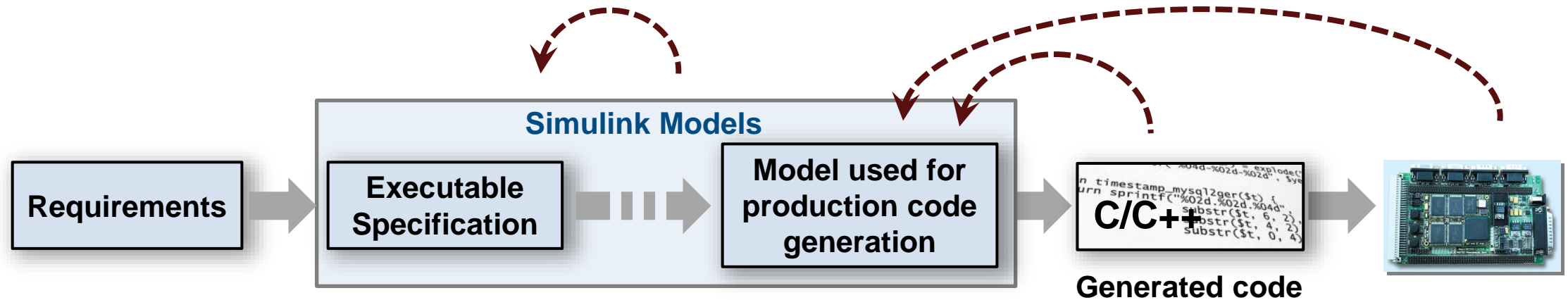
Coding

Verification



## Coverage Analysis

Verify that design has been completely tested in MIL, SIL, PIL



**MIL**, Model in the Loop, simulate on host PC

**SIL**, Software in the Loop, execute on host PC

**PIL**, Processor in the Loop, execute on target processor,  
or instruction set simulator

**Product Name:** Simulink Coverage

# High Integrity Verification Workflow

Modeling

Simulation

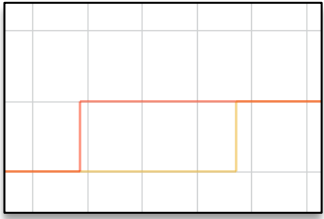
Analysis



Automation

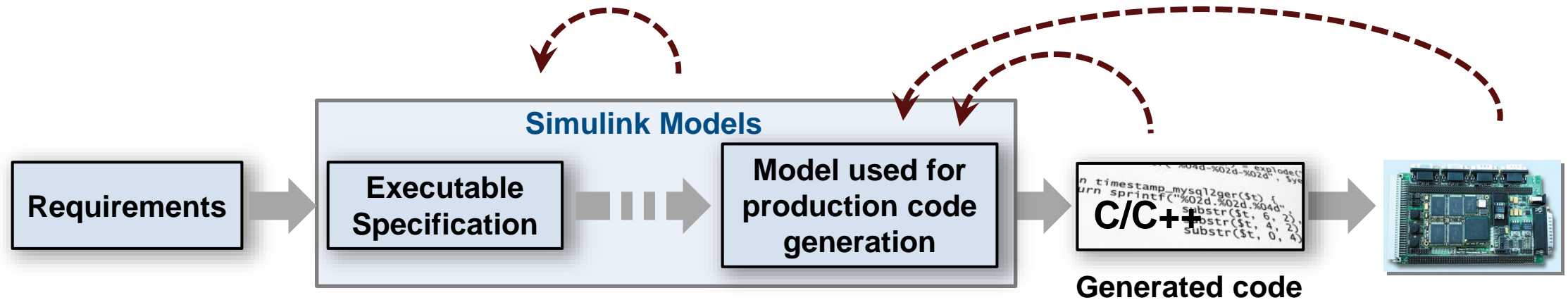
Coding

Verification



## Automatic Test Generation

Generate tests for back-to-back testing, coverage analysis, etc.



**Product Name:** Simulink Design Verifier

# High Integrity Verification Workflow

**Modeling**

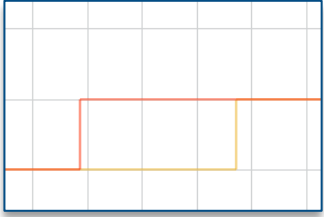
Simulation

Analysis

**Automation**

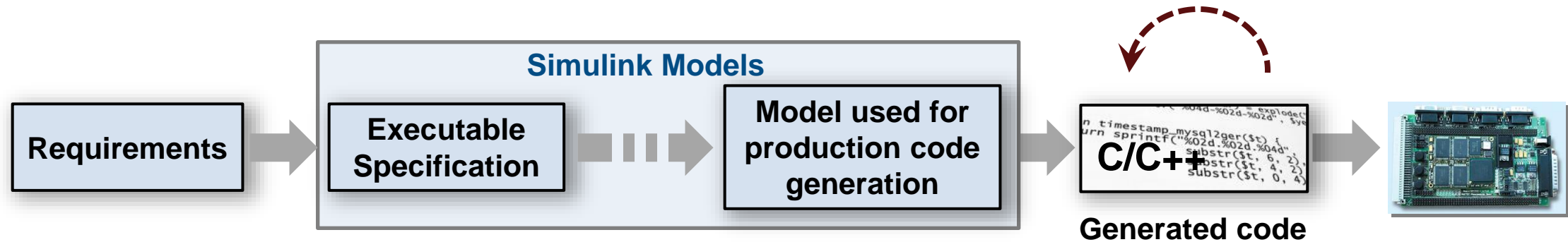
Coding

Verification



## Static Code Analysis

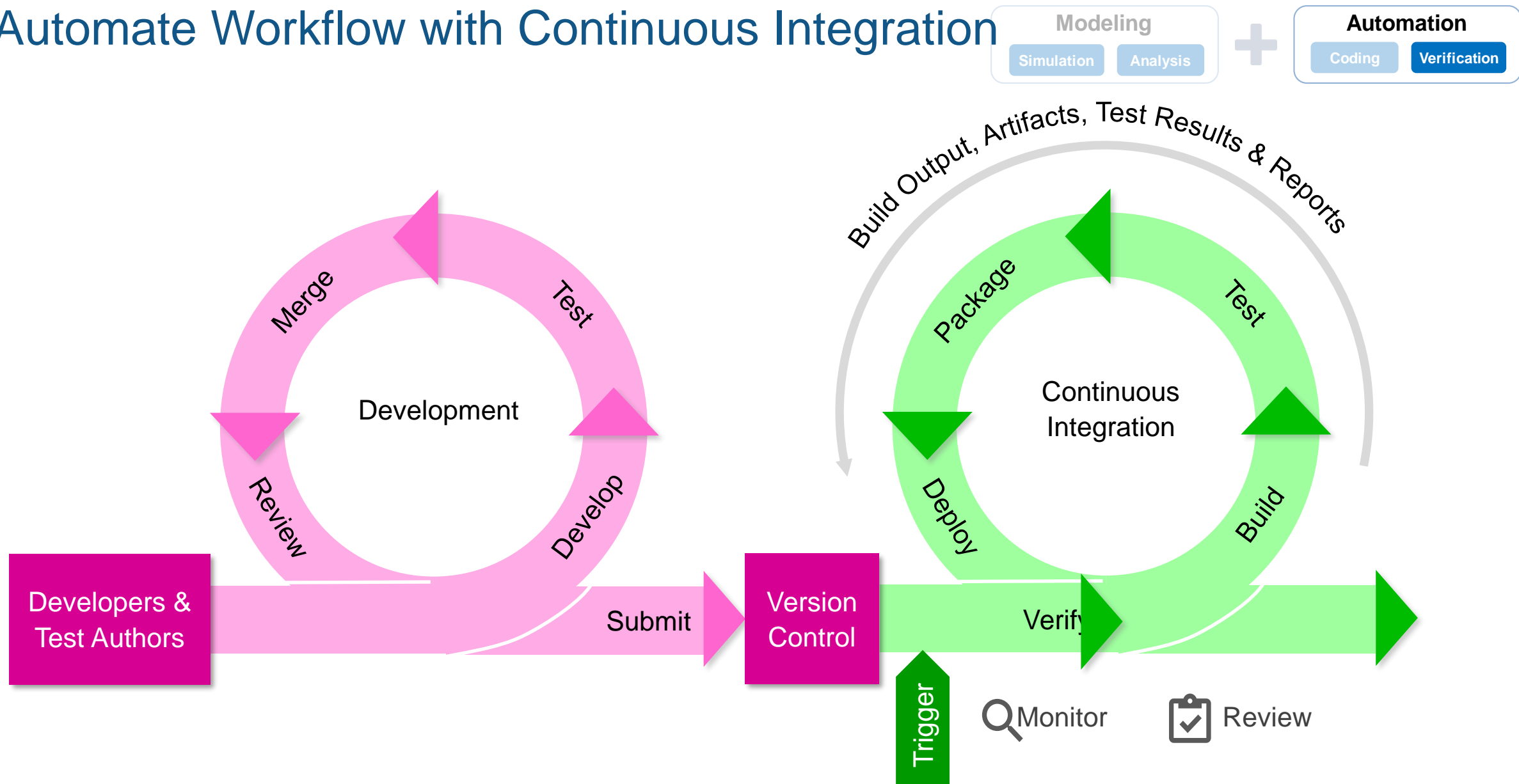
Check that code meets standards (MISRA) and free of run-time errors



**Product Name:** Polyspace Bug Finder, Polyspace Code Prover

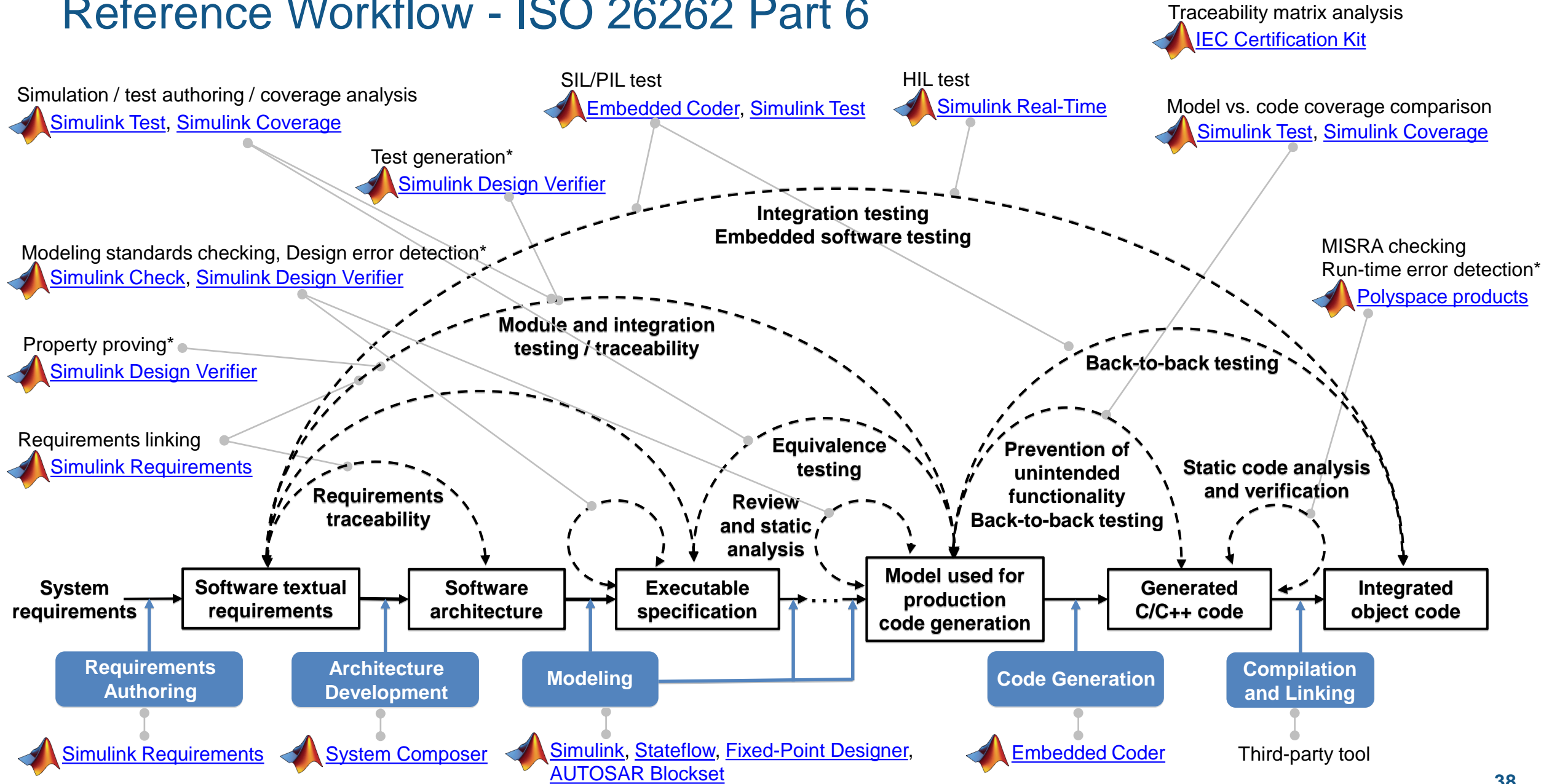


# Automate Workflow with Continuous Integration

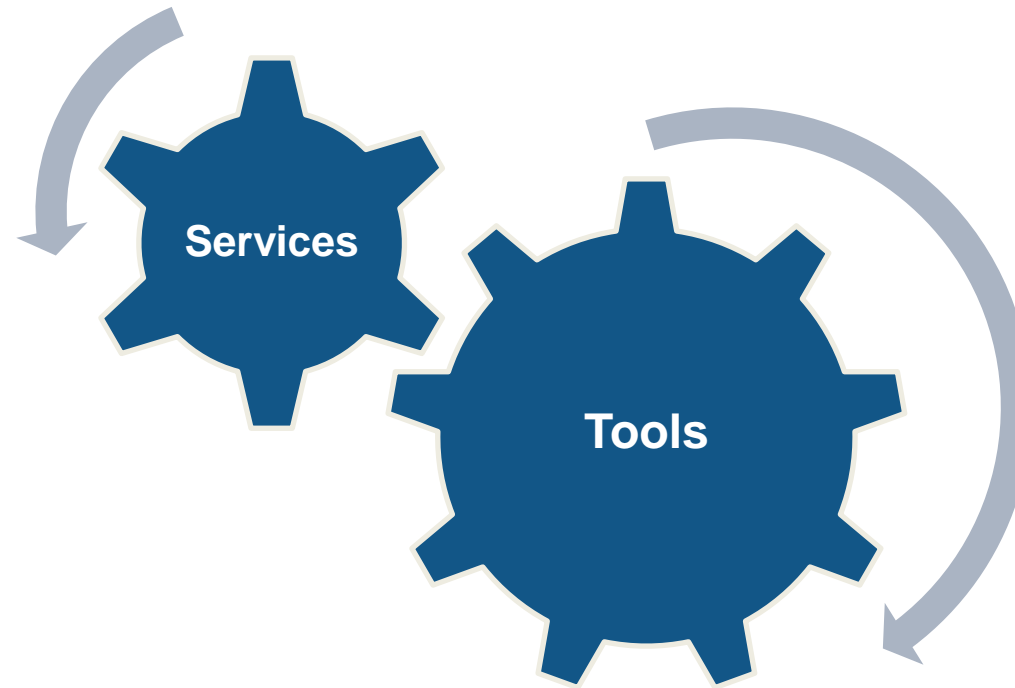


Learn more: [Continuous Integration for Verification of Simulink Models](#)

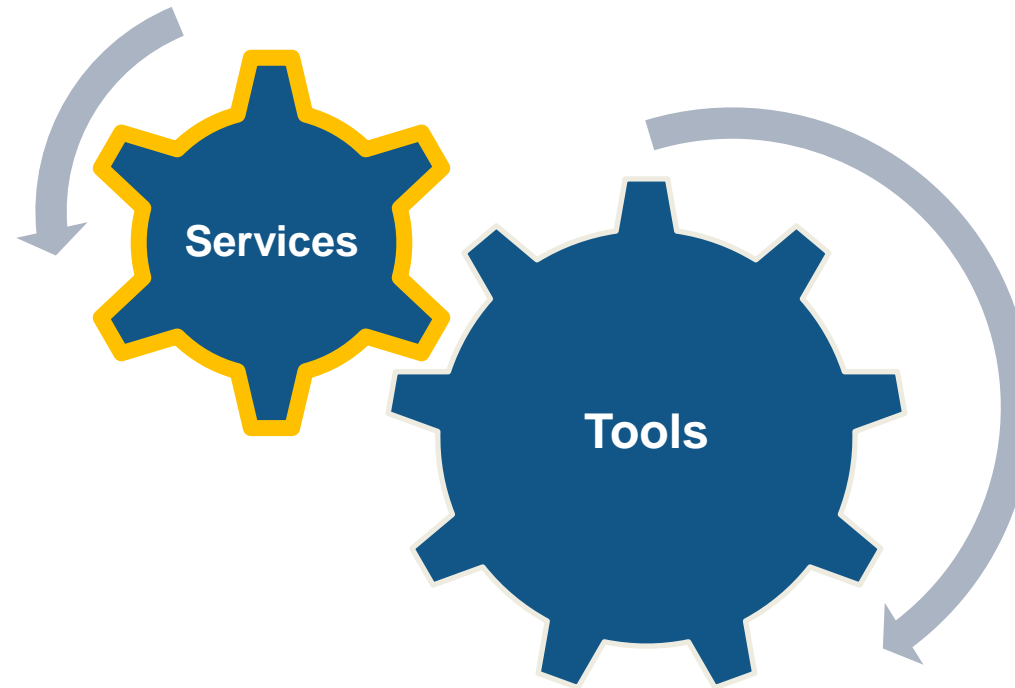
# Reference Workflow - ISO 26262 Part 6



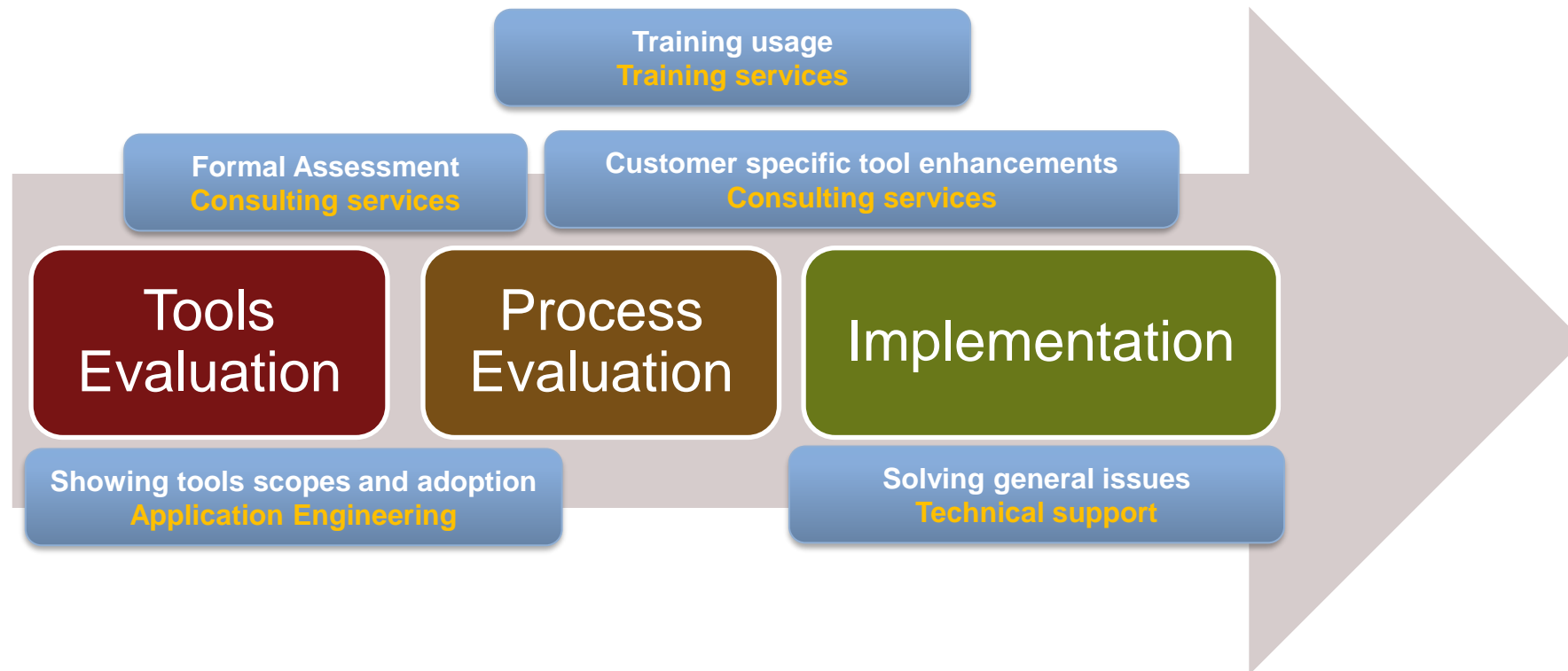
# How MathWorks helps you to achieve your goals?



# How MathWorks helps you to achieve your goals?



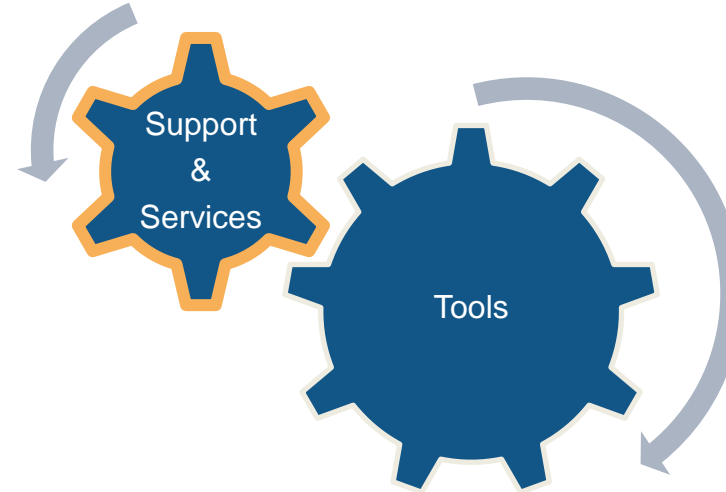
# Tools Adoption - MathWorks support and services



# How MathWorks helps you to achieve your goals?

Different groups for different aspects:

- Application Engineering
- Training Services
- Consulting Services
- Technical Support





# Application Engineering

- Show tools potential and how to adopt them
- Support directly with technical coaching
- Methodology support
- Suggestions on how adopt new features

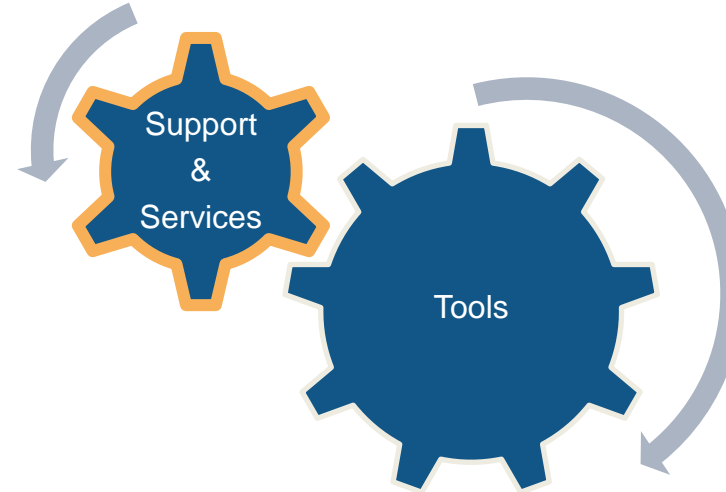
## Examples:

- Specific Seminars/Workshops (Code generation – V&V – System Engineering)
- Generic Seminars (What's new – New products)
- Advisory on specific topics
- Help on map MW workflow to customer needs
- Coordination of technical activities between MW and customers

# How MathWorks helps you to achieve your goals?

Different groups for different aspects:

- Application Engineering
- Training Services
- Consulting Services
- Technical Support



# Training Services

## SIMULINK

### FONDAMENTI

- 10 Fondamenti Simulink per la Modellazione di Algoritmi e Sistemi
- 10 Fondamenti Simulink per la Progettazione di Sistemi Aerospaziali
- 10 Fondamenti Simulink per la Progettazione di Sistemi Automobilistici
- 11 Elaborazione dei Segnali in Simulink

### INTERMEDIO

- 11 Integrazione di Codice in Simulink
- 11 Progetto di Sistemi di Controllo in MATLAB e Simulink
- 11 SimEvents per la Modellazione di Sistemi a Eventi Discreti **NUOVO**
- 11 Test Real-Time con Simulink Real-Time e Hardware Speedgoat **NUOVO**

### AVANZATO

- 11 Verifica e Validazione di Modelli Simulink
- 12 Architettura e Gestione di Modelli Simulink
- 12 Modellazione dei Sistemi di Comunicazione in Simulink

## GENERAZIONE DI CODICE

### FONDAMENTI

- 13 Test del Codice Generato in Simulink

### INTERMEDIO

- 13 Da MATLAB a C con MATLAB Coder

### AVANZATO

- 13 Embedded Coder per Generazione di Codice di Produzione
- 14 Generazione di Codice per Componenti Software AUTOSAR
- 14 Generazione di Codice HDL da Simulink
- 14 DSP per FPGA
- 14 Programmazione dei Dispositivi SoC Zynq di Xilinx in MATLAB e Simulink
- 15 Sistemi SDR (Software-Defined Radio) in Zynq con Simulink

## PRODOTTI POLYSPACE

### AVANZATO

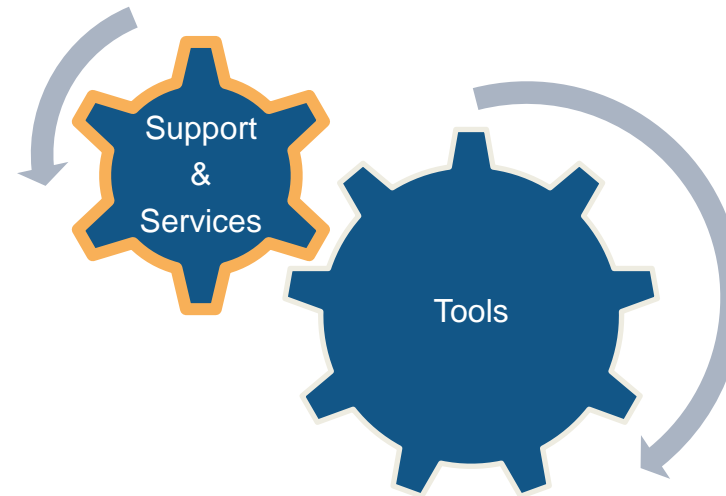
- 15 Polyspace per Verifica di Codice C/C++
- 15 Polyspace Bug Finder per l'Analisi del Codice C/C++

[Link](#)

# How MathWorks helps you to achieve your goals?

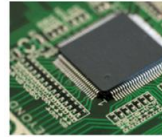
Different groups for different aspects:

- Application Engineering
- Training Services
- Consulting Services
- Technical Support



# Consulting Services

## Developing Embedded Software



Developing embedded controllers is an essential activity in bringing the power of digital electronics to automotive, aerospace, medical devices, robotics, industrial automation, and other computer controlled applications. A [Model-Based Design](#) methodology enables the realization of complex algorithms for embedded systems, from concept to validated implementation. Automatic generation of optimized, compact, and readable software code speeds up implementation for a variety of applications ranging from control to signal processing.

**MathWorks Consulting Services** brings a broad industry background and technical expertise gained from working with hundreds of companies to help you build workflows to generate code customized to your requirements.

### Generating efficient software code to meet size and speed constraints

MathWorks Consulting Services teaches you techniques to optimize the generated code for size and speed by leveraging our detailed and advanced knowledge of the tool and its configuration options. We can also build custom tool extensions when additional size and speed efficiency is required for a specific resource-constrained hardware environment.

### Controlling the generated code's functions, files, data and interfaces

We help you use built-in product features, functions, or APIs so your software code meets specified programming standards for function prototypes, file formats, file partitioning, data structures, and interfaces. If the functionality you need is not built-in, MathWorks Consultants can construct customized capabilities as needed and transfer the knowledge to you so you can evolve and maintain the new capabilities.

### Meeting certification and safety standards

MathWorks Consultants have worked with engineers to implement algorithms and development processes that comply with certification standards such as [DO-178](#), [ISO 26262](#), and [IEC 61508](#). We can help establish or fine-tune your development process around certification standards, ensuring that you achieve the best possible value from MATLAB and Simulink, as well as a significant reduction in development effort.

## Leverage the Expertise of the MathWorks Organization



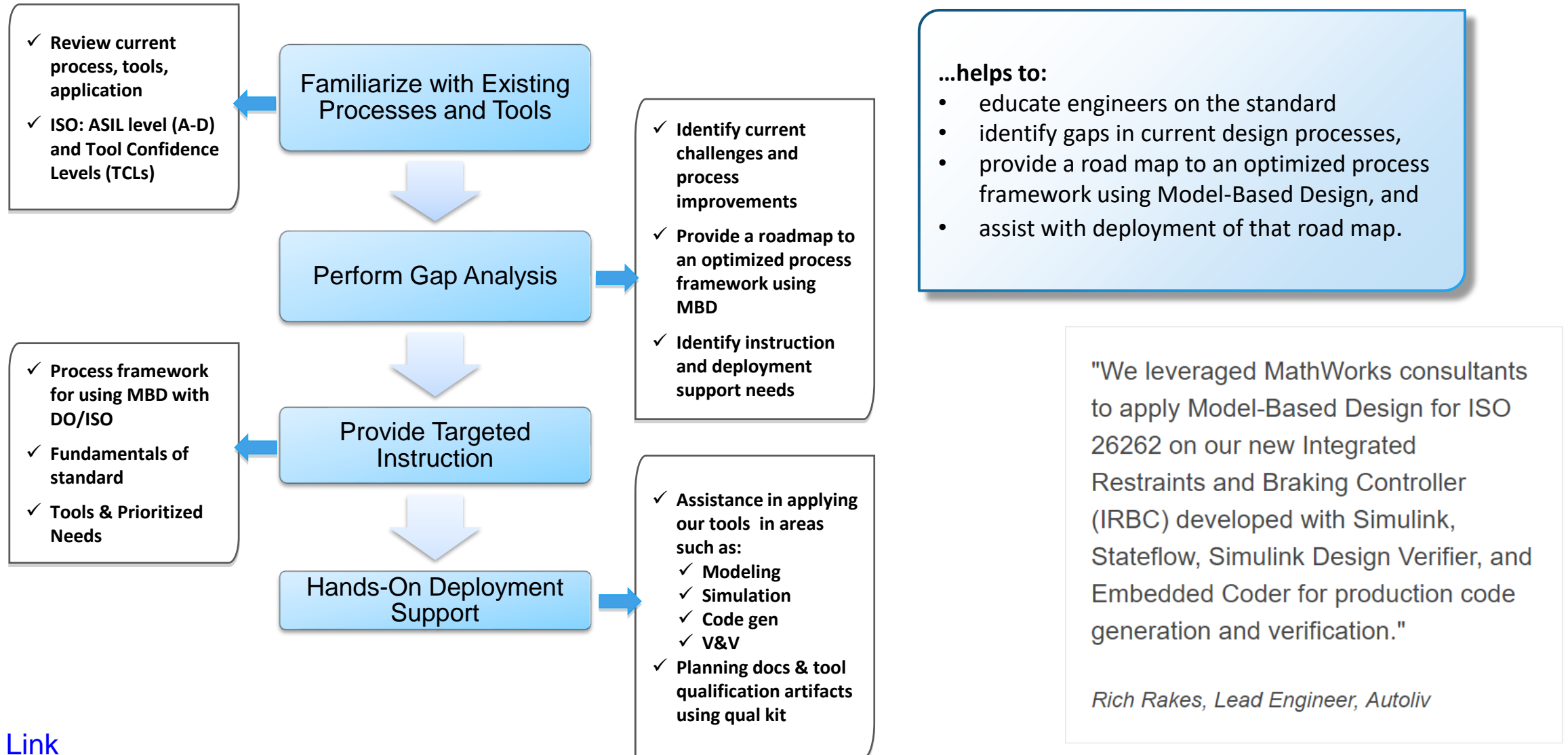
Having deep product knowledge and broad technical experience, we guide your team to apply best practices to your development projects. We offer "from the source" insights into our products and exclusive access to our in-house subject matter experts. Our worldwide presence means that we live and work where you do, speak your language, and understand local customs and business practices. Wherever you are, we're there to help.



## Proven Solutions

- [Battery Simulation and Controls](#)
- [Developing Embedded Software](#)
- [Developing Embedded Targets Advisory Service](#)
- [DO-178 Certification Advisory Service](#)
- [Early Verification and Validation with Model-Based Design](#)
- [Electrical Power Systems Simulation](#)
- [Financial Analysis and Trading](#)
- [Image Processing and Computer Vision](#)
- [ISO 26262 Process Deployment Advisory Service](#)
- [Load Forecasting](#)
- [MATLAB in Business Critical Applications](#)
- [MATLAB with Hadoop and Spark](#)
- [Model-Based Design for Production Real-Time Embedded Systems](#)
- [Model-Based Design Process Assessment and Maturity Framework](#)
- [Model-Based Design Process Establishment](#)
- [Motor Control Development](#)
- [Operations, Logistics, and Supply Chain Management](#)
- [Optimal Engine Calibration](#)
- [Predictive Maintenance](#)
- [Signal Processing and Communications](#)
- [Software Development with MATLAB](#)
- [Software Upgrade Service](#)
- [Thermal Systems Modeling](#)
- [Tools Integration](#)

# ISO 26262 Process Deployment Advisory Service

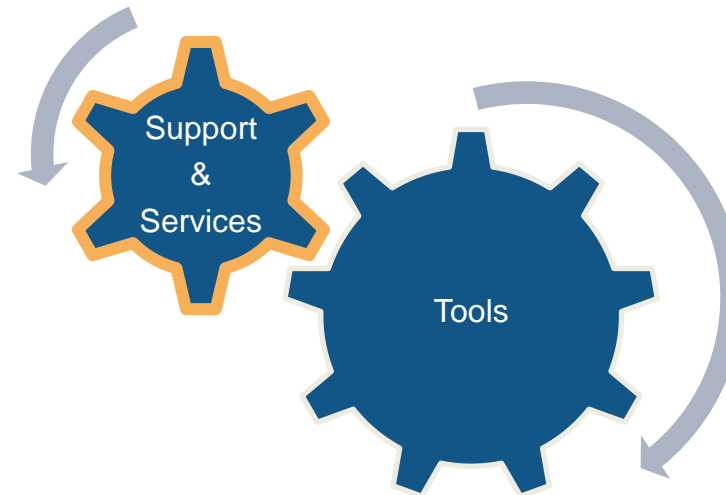




# How MathWorks helps you to achieve your goals?

Different groups for different aspects:

- Application Engineering
- Training Services
- Consulting Services
- Technical Support



# Technical Support

## Contact Support

Create Service Request

Hosted by [force.com](#)

**Eligibility:** Access to technical support requires a valid license number and a [Software Maintenance Service](#) subscription.

**Students:** Technical support from MathWorks is available for activation, installation and bug-related issues. For additional help visit our [student resource page](#) or contact your instructor.

## Did You Try?



### Installation Help

Explore resources for installation, activation, and startup



### Documentation

Explore MathWorks Documentation



### MATLAB Answers

Ask questions and get answers

## Support

### Step 1/2: Select Request Type

- ☐ Customer Service: License modifications, product activation
- ☒ Technical Support: Installation, product help, bugs, suggestions, documentation errors, outages
  - ☐ Installation, license manager, startup, outages
  - ☒ Product help, bugs, suggestions or documentation errors

MathWorks is a worldwide organization. Your submission will be accessed by staff who will assist with your service request. If you plan to attach any files that contain export controlled information, call [MathWorks Technical Support](#) for your country before you submit your request.

\*Indicates Required Information

### Step 2/2: Answer questions related to your request and submit it

\*Service Request Subject

\*Description

[What do I include in my description?](#)

\*Execute ver command and paste output:

[What is VER -Support?](#)

\*Release

-- Select Release --

\*Product

-- Select Product--

If you do not know which product to select, select one that is the closest match.

Attach File(s):

[Scegli file](#) | Nessun file selezionato

File size limit: 5MB

# Q&A

More information:

**Nukul Sehgal**

**Application Engineer**

**[nsehgal@mathworks.com](mailto:nsehgal@mathworks.com)**

