

Model-based design and quality management

an automated process for safety-reliable production code ■

IWES 2022

Politecnico di Bari, 22-23 September



Agenda.

01 Software-defined vehicle

02 Safety and quality

03 ISO-26262

04 Solution

05 Results

An aerial, long-exposure photograph of a multi-level highway interchange at night. The image is dominated by vibrant, multi-colored light trails from moving vehicles, creating a sense of dynamic motion. The trails are primarily in shades of blue, purple, and red, with some white and yellow highlights. The road structure is complex, with multiple levels and curved ramps. The overall color palette is dark, with the bright light trails providing the main visual interest.

1. Software-defined vehicle



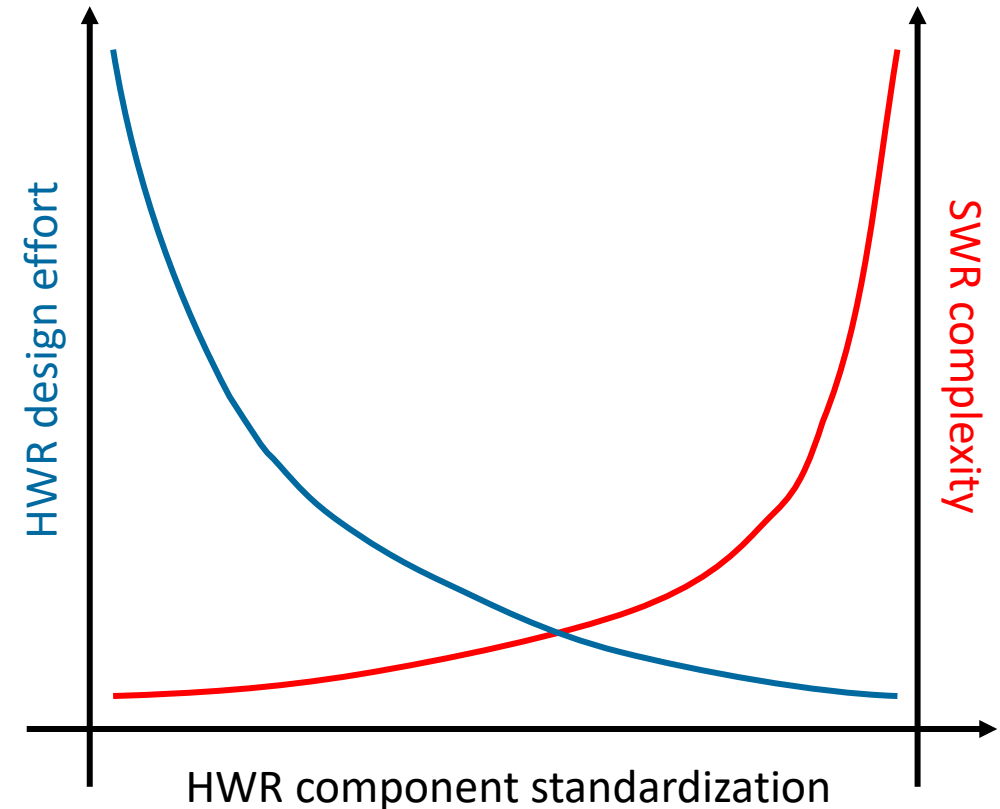
1. Software-defined vehicle.

Introduction

The standardization of hardware components is lowering the effort in designing them, while increasing the demand for a *more and more efficient software*. In this scenario, the concept of **software-defined vehicle** comes into play.

Nowadays, the electronic architecture of a vehicle includes up to hundreds of Electronic Control Units, with more than hundreds million lines of code: performances, configuration, connectivity and automation of vehicles are constantly upgraded by improving both the software and the development process; the car, which now can be considered a **connected and smart mobile device**, can be updated remotely (**OTA**), without entering a workshop.

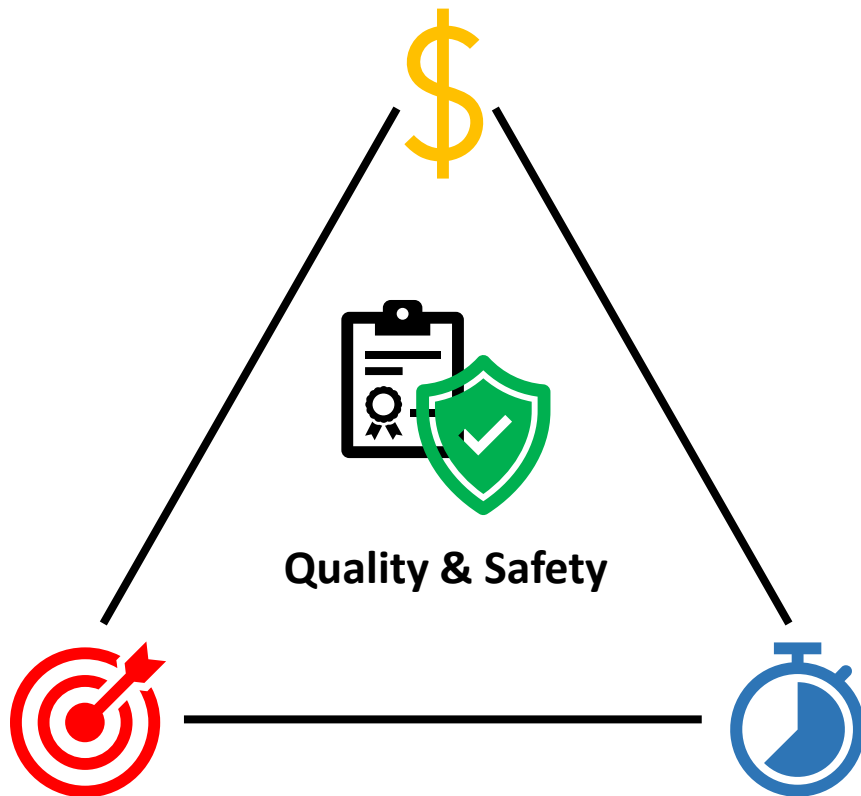
[Source1](#), [source2](#)





1. Software-defined vehicle.

New accountabilities for the software



The most obvious consequence of this new way of conceiving cars is that the **safety and security** aspects, previously attributed to the mechanical integrity and the performance of the parts constituting a vehicle, are now mostly related to the **software** and, not least, to the **quality management** during the **development process**.

As a result, in order to comply with quality management and new safety standards, the design goals are challenging, thus time and costs are increasing considerably.

An aerial, long-exposure photograph of a multi-level highway interchange at night. The image is dominated by vibrant light trails from cars, creating a sense of motion and energy. The trails are primarily in shades of blue, purple, and red, with some white and yellow from streetlights. The road markings and lane dividers are visible, and the overall scene is illuminated by the ambient light of the city at night.

2. Safety and quality



2. Safety and quality.

Good quality management for easily applying safety constraints

It is important for companies to be **certified** according to a **quality management (QM) standard** (ISO 9001, QM9000, SPICE). This would provide a strong foundation for building **functional safety management (FSM)**, which is the logical improvement of QM requirements, taking safety aspects into account.

Moreover, a good QM defines a standardized and optimized process for producing documentation: this is the best starting point for developing **automatic solutions** for the QM and the FSM. This will help achieving the requirements for the certification according to a **safety standard (ISO 26262)**.

FSM
(ISO 26262)

QM
(ISO 9001, QM9000, SPICE, CMMI, IATF16949)

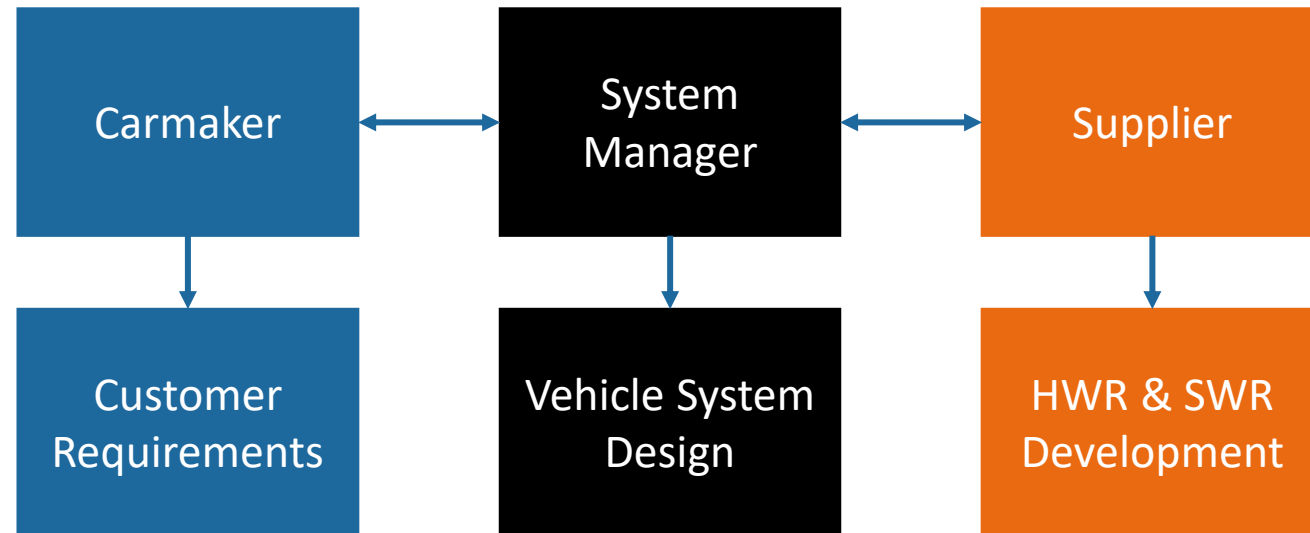
An aerial, long-exposure photograph of a multi-level highway interchange at night. The image is dominated by vibrant light trails from moving vehicles, creating a sense of dynamic motion. The trails are primarily in shades of blue, cyan, and magenta, with some white and red highlights. The road structure is complex, with multiple levels and curved ramps. The overall color palette is cool and futuristic. A large, semi-transparent orange oval is positioned on the left side of the image, partially overlapping the text.

3. ISO-26262



3. ISO-26262.

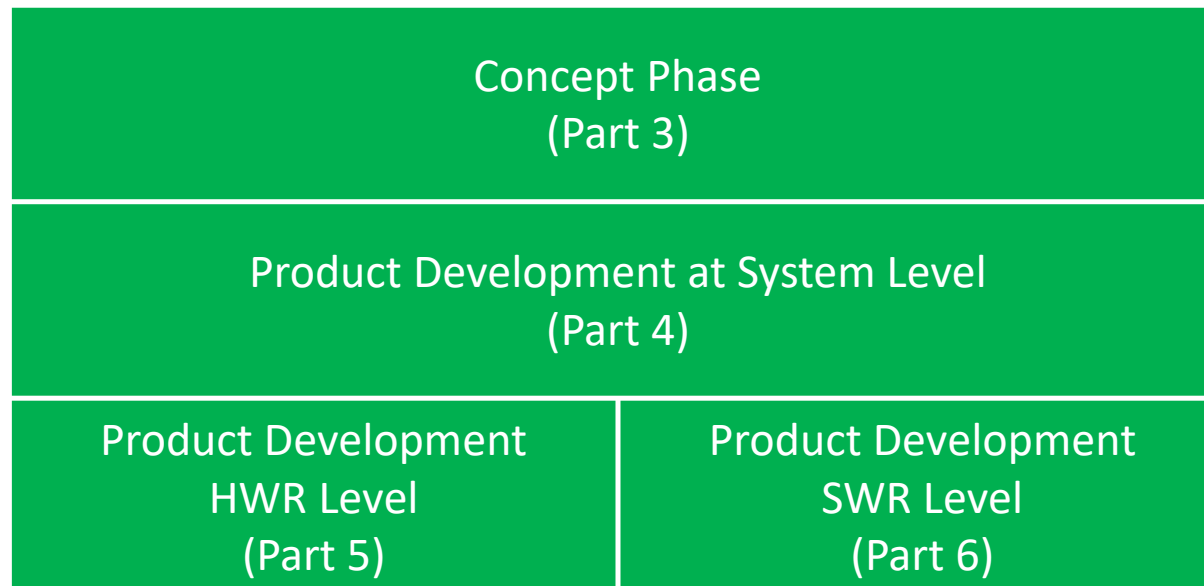
Actors





3. ISO-26262.

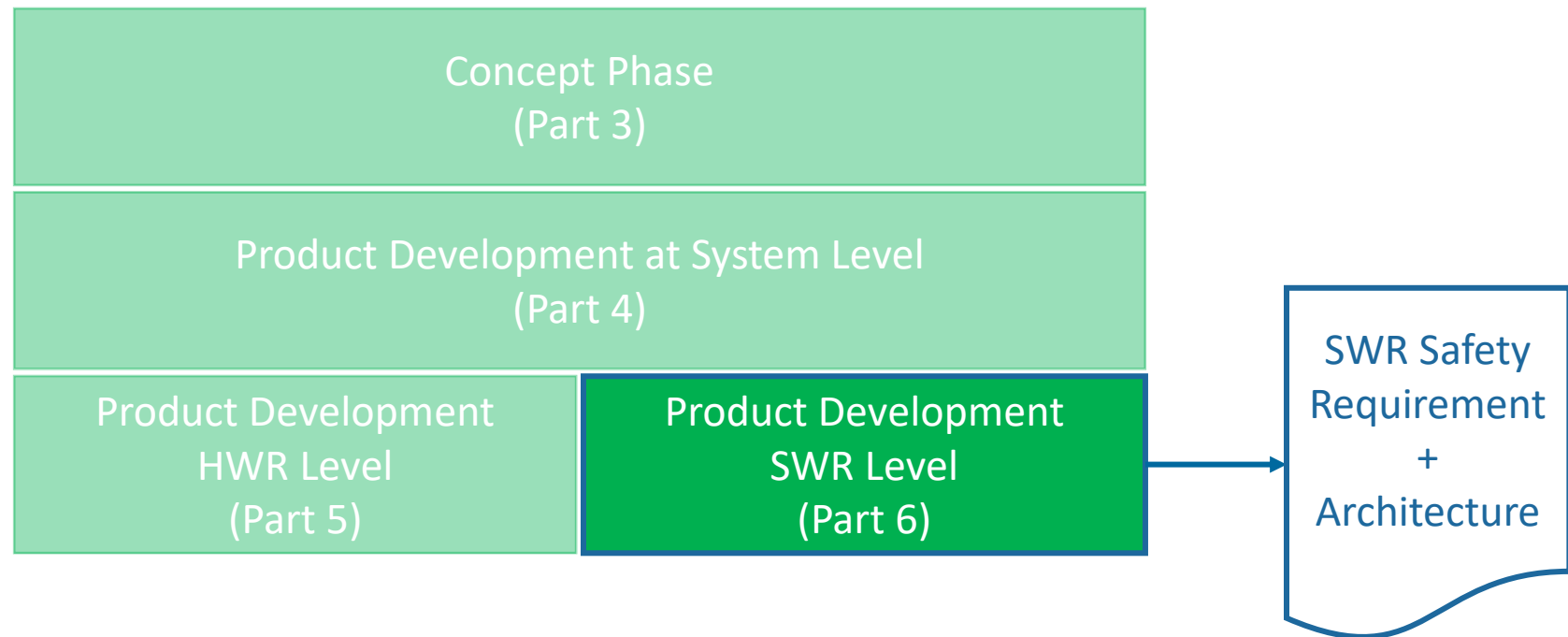
Process-long support





3. ISO-26262.

Part 6 WP's



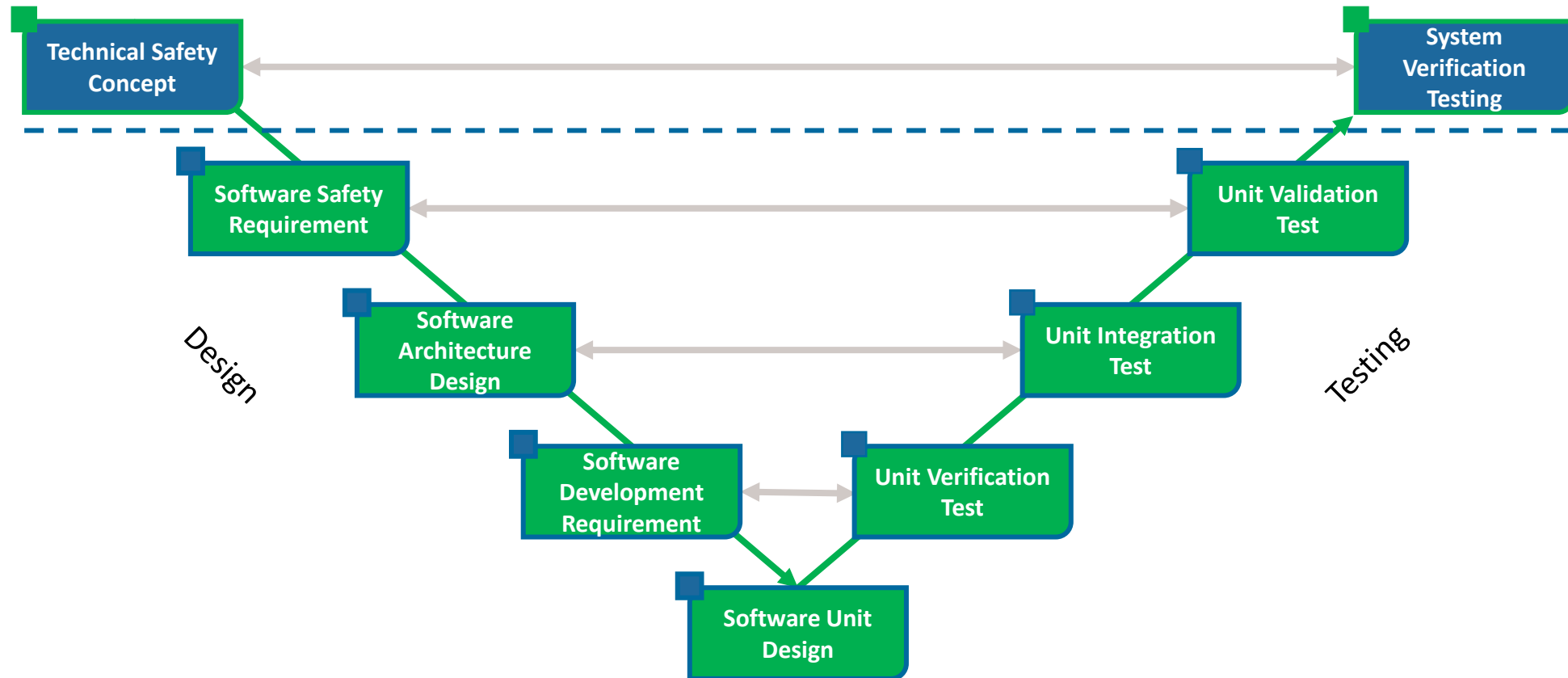
An aerial, long-exposure photograph of a multi-level highway interchange at night. The image is dominated by vibrant light trails from cars, creating a sense of motion and energy. The trails are primarily in shades of blue, purple, and red, with some white and yellow. The roads are illuminated by streetlights, and the overall scene is a complex network of curved and straight paths. The text '4. Solution' is overlaid on the left side of the image, partially obscured by an orange rounded rectangle.

4. Solution



4. Solution.

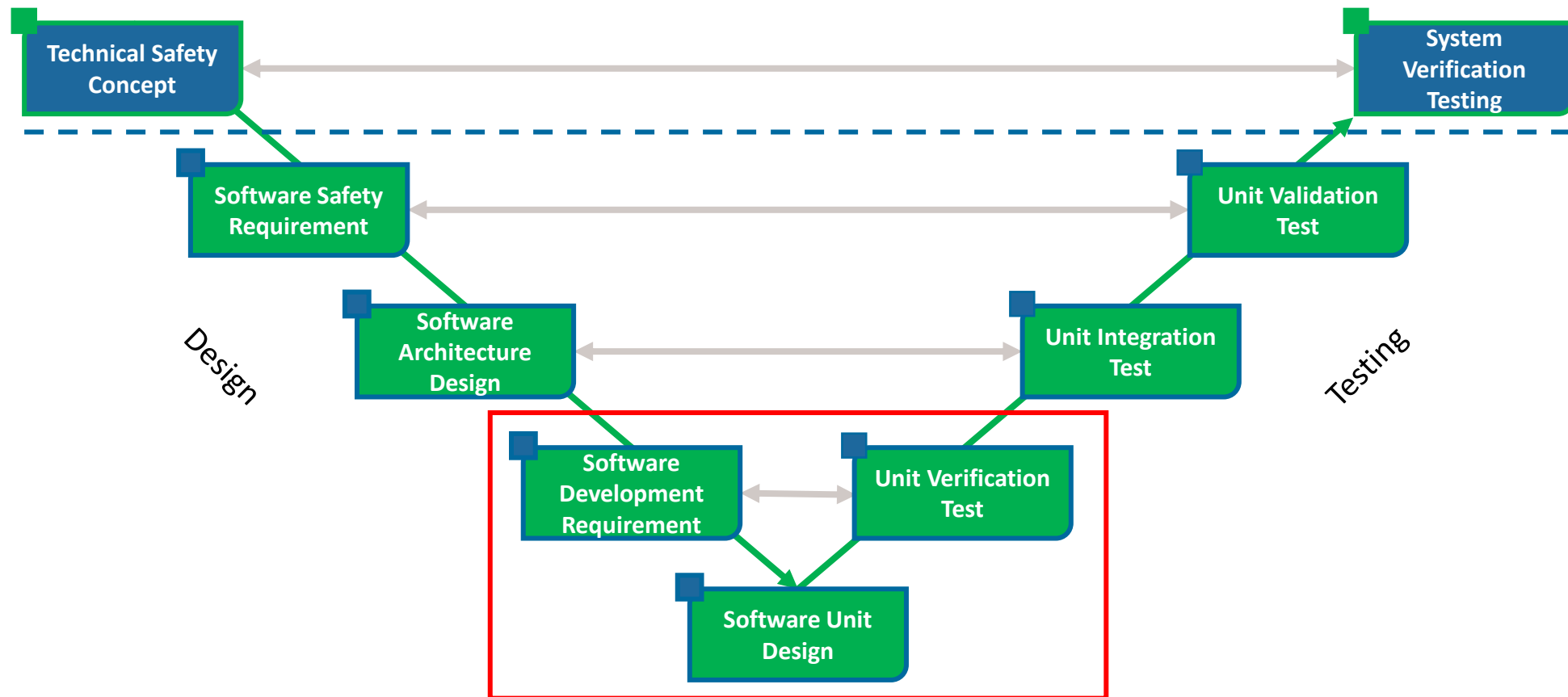
Software development and testing





4. Solution.

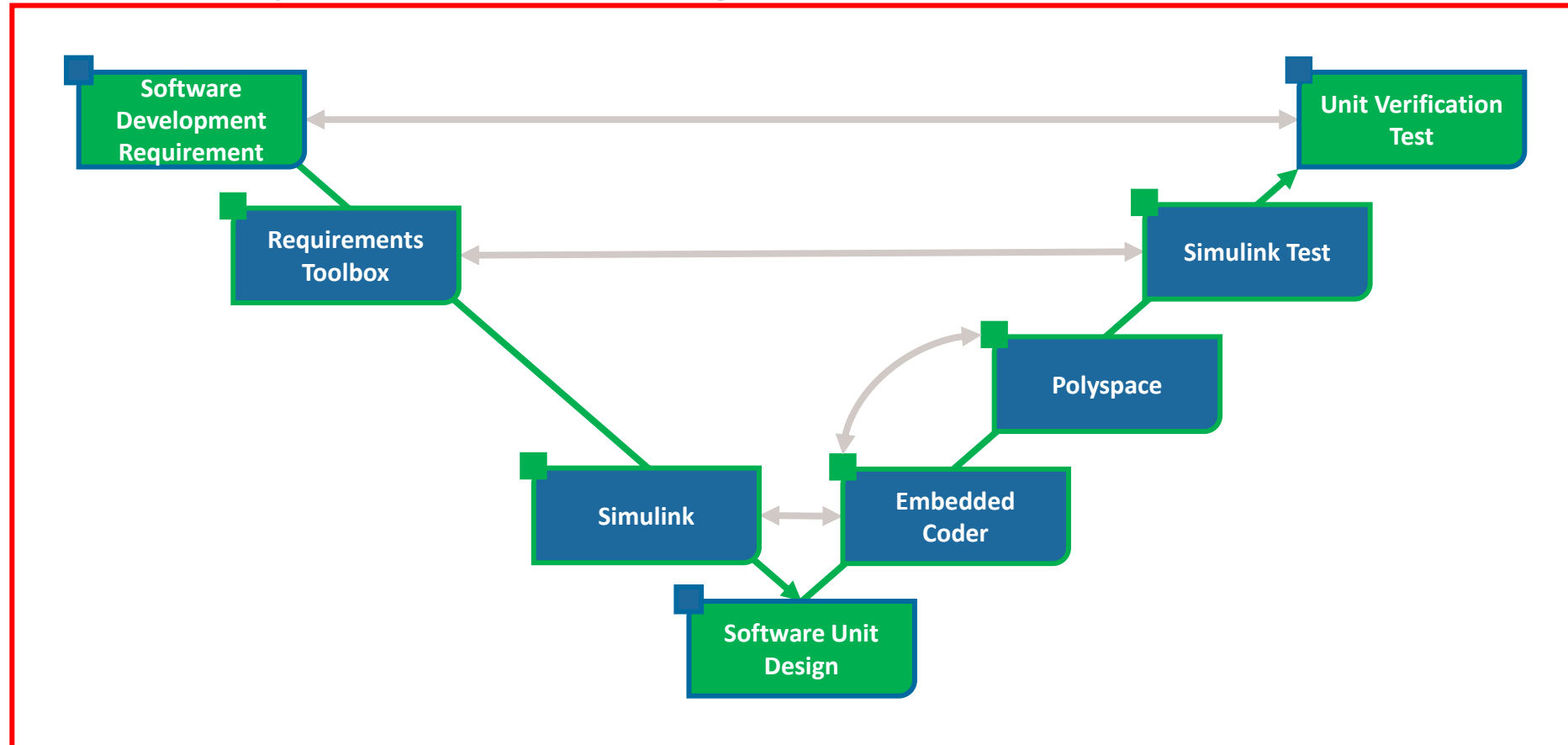
Software development and testing





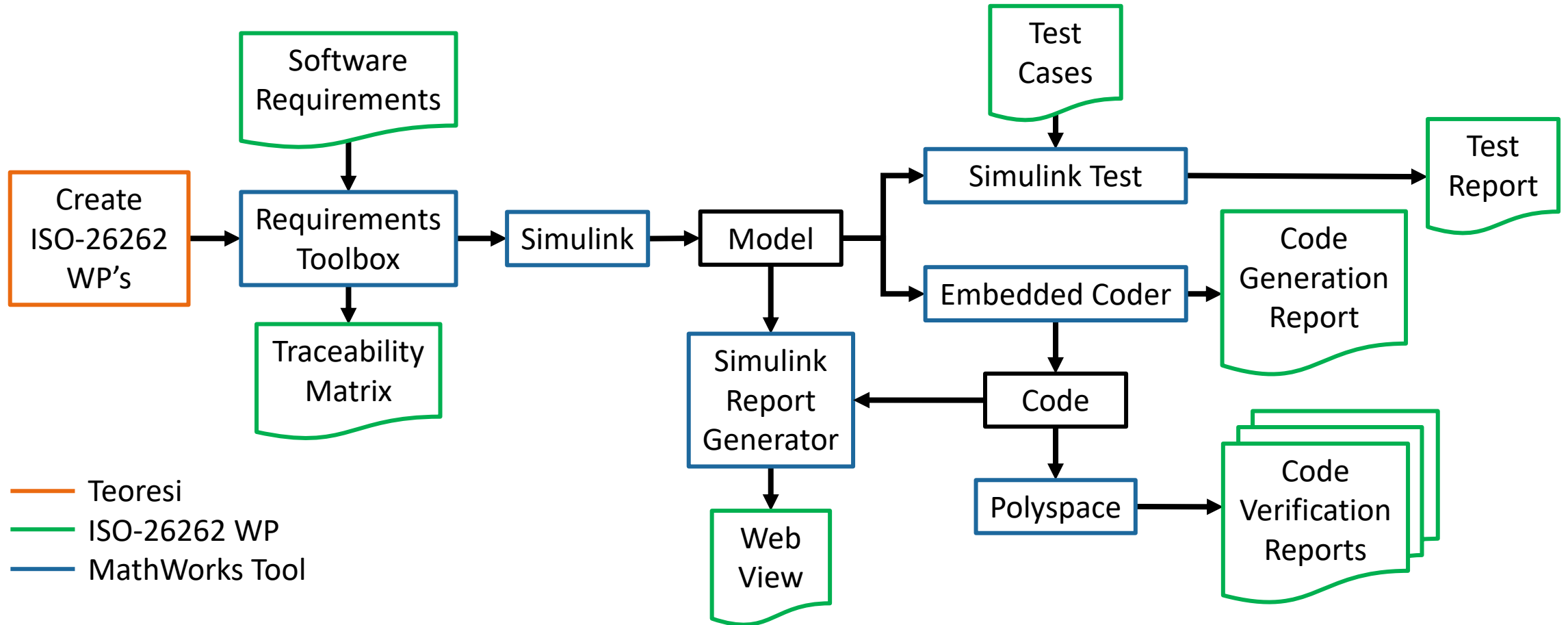
4. Solution.

Software development and testing





4. Solution.



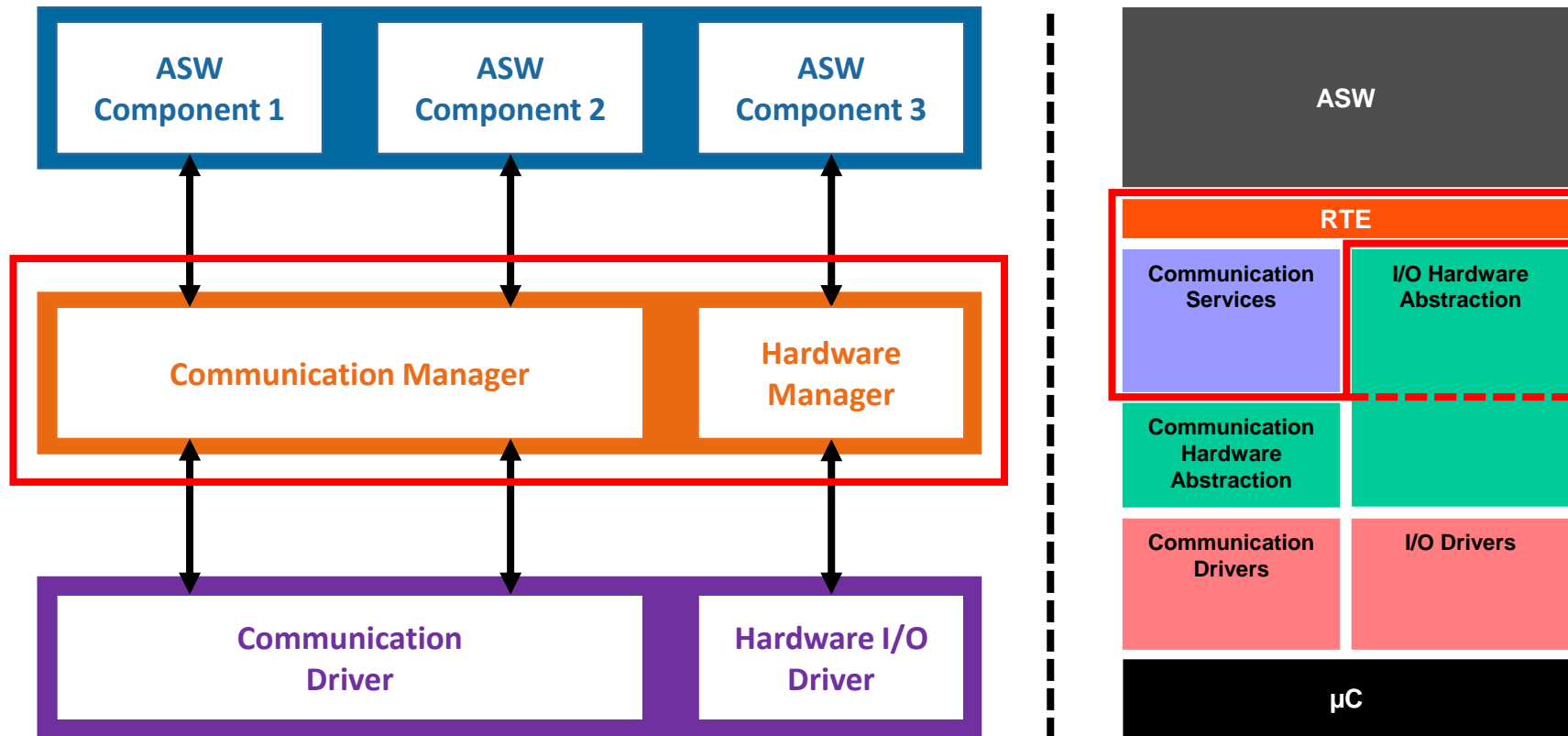
An aerial, long-exposure photograph of a multi-level highway interchange at night. The image is dominated by vibrant light trails from cars, creating a sense of motion and energy. The trails are primarily in shades of blue, purple, and red, with some white and yellow. The roads are illuminated by streetlights, and the overall scene is a complex network of curved and straight paths. The text '5. Results' is overlaid on the left side of the image, partially obscured by an orange shape.

5. Results



5. Results.

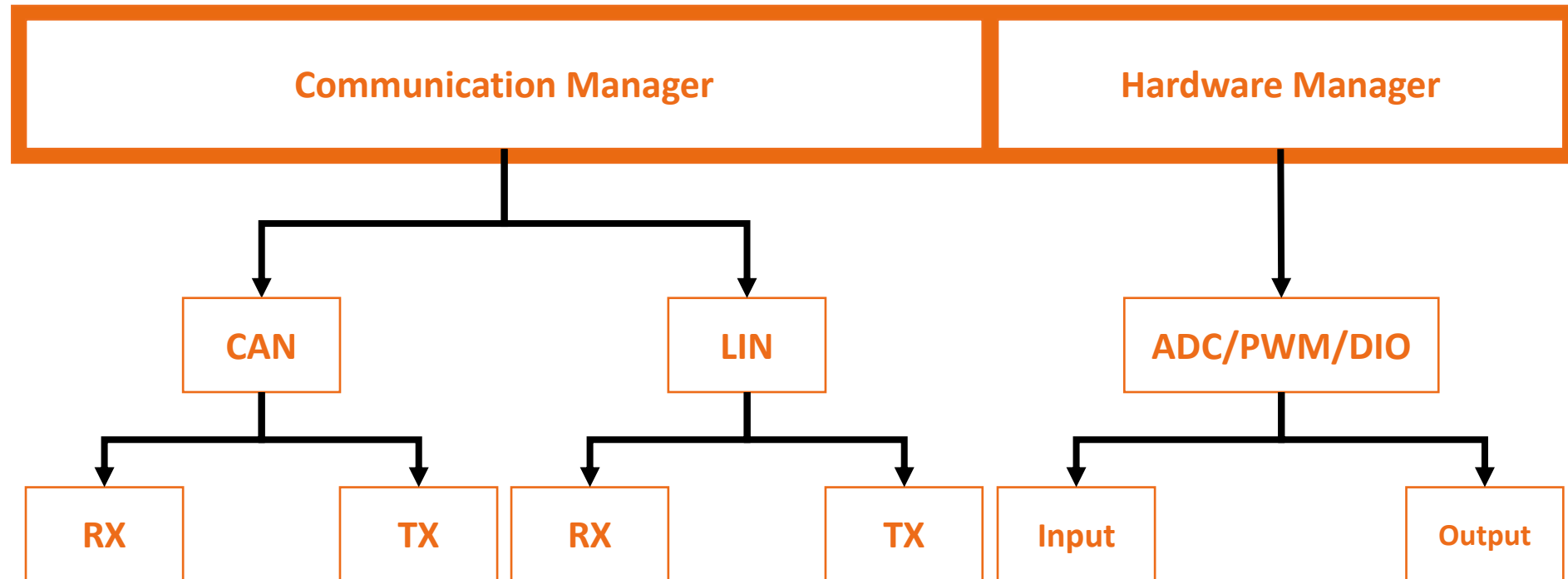
Middleware development





5. Results.

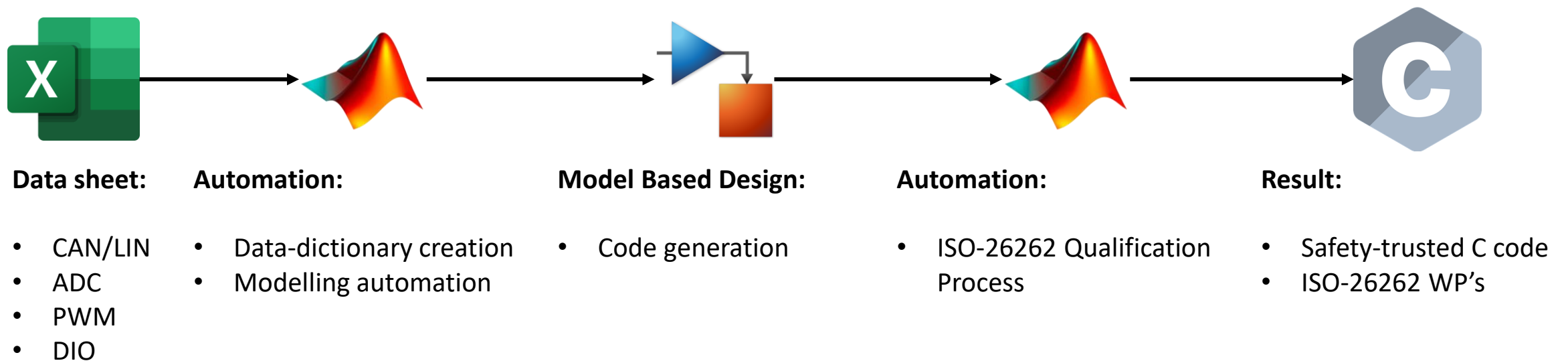
Middleware development





5. Results.

Middleware development





5. Results.

Project

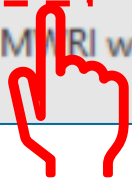
PLOTS	APPS	PROJECT	PROJECT SHORTCUTS	
0. Clean Project Folder (MWR)		1.1.1 Create MWRI WP's		2. Create MWR for proto
1. Create MWR		1.2 Create MWRO with ISO-26262 WP's		2.1 Create MWRI for proto
1.1 Create MWRI with ISO-26262 WP's		1.2.1 Create MWRO WP's		2.2 Create MWRO for proto
		MWR		



5. Results.

Project

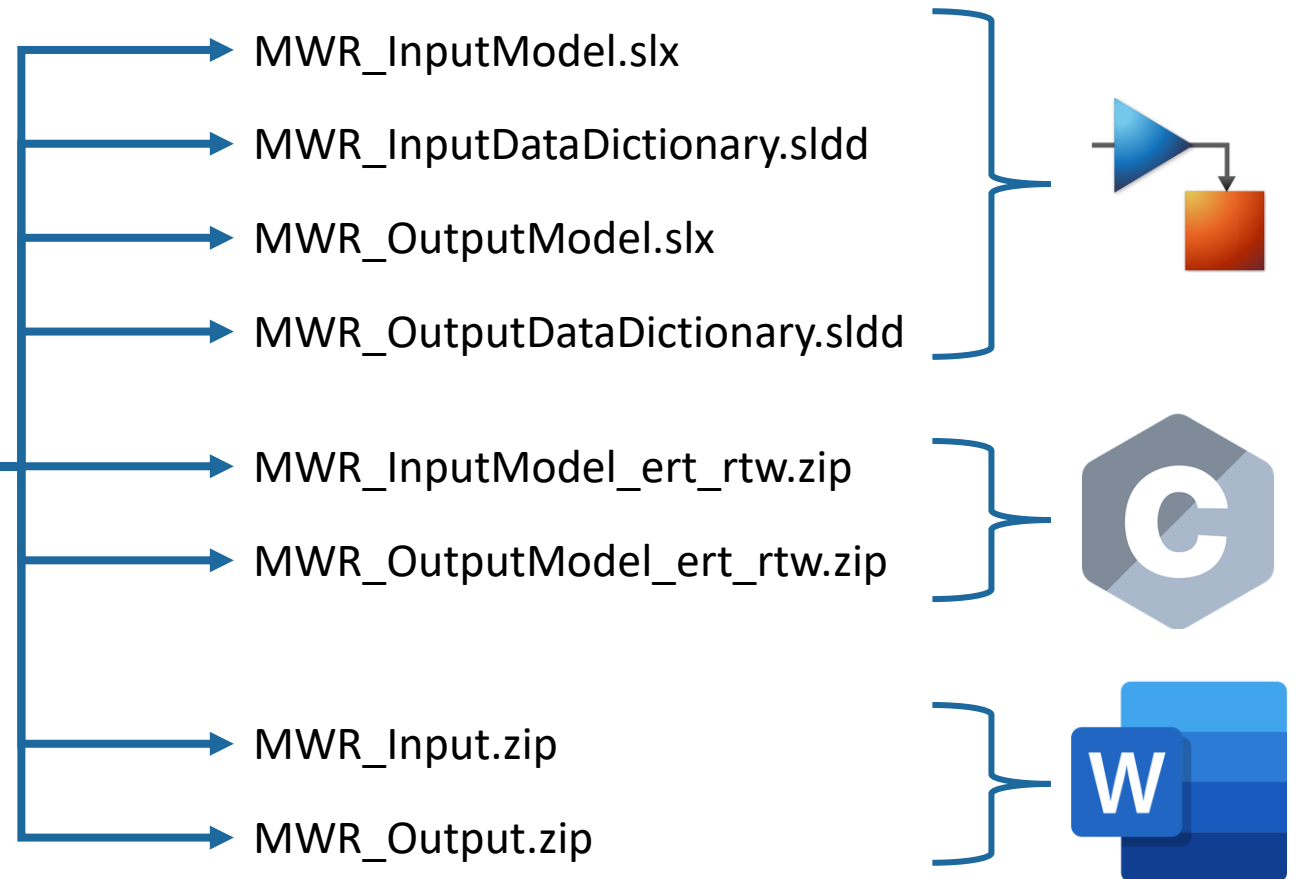
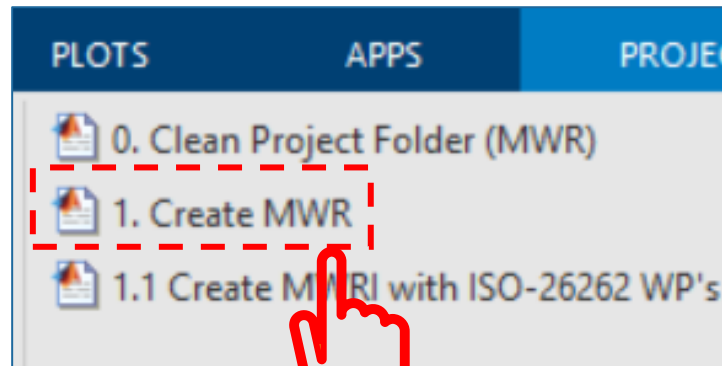
PLOTS	APPS	PROJECT	PROJECT SHORTCUTS
0. Clean Project Folder (MWR)		1.1.1 Create MWRI WP's	2. Create MWR for proto
1. Create MWR		1.2 Create MWRO with ISO-26262 WP's	2.1 Create MWRI for proto
1.1 Create MWRI with ISO-26262 WP's		1.2.1 Create MWRO WP's	2.2 Create MWRO for proto
		MWR	





5. Results.

Project





5. Results.

WP's | Software requirements specification

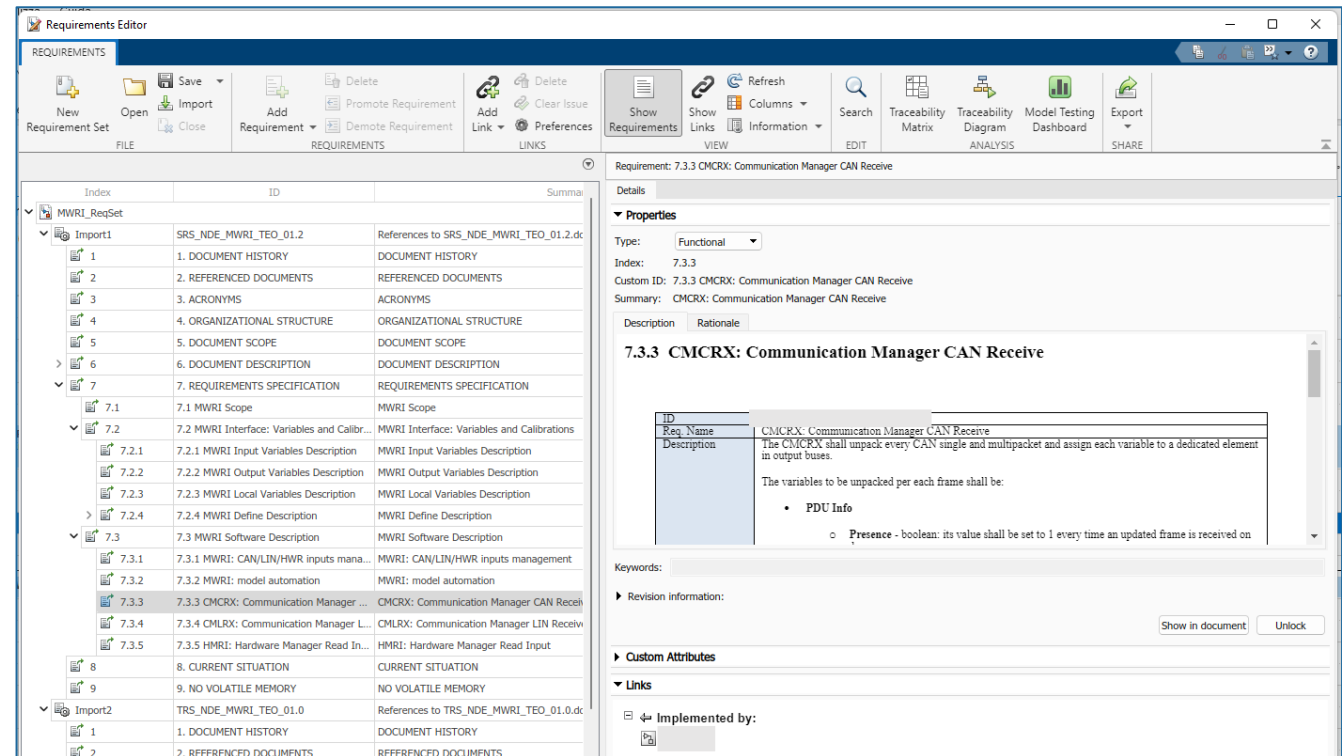


SOFTWARE REQUIREMENTS SPECIFICATIONS

[SRS_NDE_MWRI_TEO_01.2.docx] Version: 01.2



MWR Input - ISO-26262 Software Requirements Specifications



The screenshot shows the Requirements Editor interface. On the left, a tree view displays the hierarchy of requirements, with 7.3.3 CMCRX: Communication Manager CAN Receive selected. The main area shows the details for this requirement, including its ID, name, and description. The description states: "The CMCRX shall unpack every CAN single and multipacket and assign each variable to a dedicated element in output buses. The variables to be unpacked per each frame shall be: PDU Info. Presence - boolean: its value shall be set to 1 every time an updated frame is received on".



5. Results.

The screenshot displays the Requirements Editor interface. The left pane shows a hierarchical tree of requirements under 'MWRI_ReqSet'. The right pane provides a detailed view of the selected requirement, '7.3.3 CMCRX: Communication Manager CAN Receive'.

Requirement Details:

- Title:** Requirement: 7.3.3 CMCRX: Communication Manager CAN Receive
- Type:** Functional
- Index:** 7.3.3
- Custom ID:** 7.3.3 CMCRX: Communication Manager CAN Receive
- Summary:** CMCRX: Communication Manager CAN Receive

Description:

7.3.3 CMCRX: Communication Manager CAN Receive

ID	Req. Name	Description
7.3.3	CMCRX: Communication Manager CAN Receive	The CMCRX shall unpack every CAN single and multipacket and assign each variable to a dedicated element in output buses.

The variables to be unpacked per each frame shall be:

- **PDU Info**
 - **Presence** - boolean: its value shall be set to 1 every time an updated frame is received on

Keywords:

Revision information:

Custom Attributes:

Links:

Implemented by:



5. Results.

WP's | Test cases and test reports



Report Generated by Test Manager

Title: MWRI Equivalence Test Report
Author: [REDACTED]
Date: 18-Sep-2022 20:46:47

Test Environment

Platform: PCWIN64
 MATLAB: (R2022a)

Summary

Name

Outcome **Duration (Seconds)**

Results: 2022-Sep-18 20:31:51	5 ✓	689.516
MWRI TestFile 18 Sep 2022 20 30 22	5 ✓	689.516
MWRI TestSuite Equivalence	5 ✓	689.517
MWRI TestCase Equivalence	5 ✓	689.517
Iteration1	✓	253.752
Iteration2	✓	109.339
Iteration3	✓	106.609
Iteration4	✓	105.97
Iteration5	✓	110.624



5. Results.

Results: 2022-Sep-18 20:31:51

Result Type: Result Set
 Parent: None
 Start Time: 18-Sep-2022 20:33:38
 End Time: 18-Sep-2022 20:45:07
 Outcome: Total: 5, Passed: 5

Aggregated Coverage Results

Analyzed Model	Sim Mode	Complexity	Decision	Condition	MCDC	Function	Function call	Execution
MWRI model	ModelRefSIL	380	98%	98%	100%	100%	100%	100%
MWRI model	Normal	1204	100%	100%	100%	--	--	100%
MWRI model	SIL	30	--	--	--	0%	--	0%

[Back to Report Summary](#)

MWRI_TestFile_18_Sep_2022_20_30_22

Test Result Information

Result Type: Test File Result
 Parent: [Results: 2022-Sep-18 20:31:51](#)
 Start Time: 18-Sep-2022 20:33:38
 End Time: 18-Sep-2022 20:45:07
 Outcome: Total: 5, Passed: 5



5. Results.

Equivalence Comparison														
Name	Abs Tol	Rel Tol	Lead Tol	Lag Tol	Max Diff	Data Type 1	Units 1	Sample Time 1	Data Type 2	Units 2	Sample Time 2	Interp	Sync	Link to Plot
✓	0	0	0	0	0	uint32		0.005	uint32		0.005	zoh	union	Link
✓	0	0	0	0	0	boolean		0.005	boolean		0.005	zoh	union	Link
Ok	0	0	0	0	0	boolean		0.005	boolean		0.005	zoh	union	Link
✓	0	0	0	0	0	boolean		0.005	boolean		0.005	zoh	union	Link
✓	0	0	0	0	0	uint8		0.005	uint8		0.005	zoh	union	Link
1	0	0	0	0	0	uint8		0.005	uint8		0.005	zoh	union	Link
xPc	0	0	0	0	0	uint8		0.005	uint8		0.005	zoh	union	Link
xPc	0	0	0	0	0	uint8		0.005	uint8		0.005	zoh	union	Link



5. Results.

WP's | Traceability matrix

Filter Status																	
Top:																	
Focused Scope:	MWRI_model.slx																
Left:																	
Focused Scope:	MWRI_ReqSet.slreqx																
		MWRI_model	MIL_MODE	CODEGEN_MODE	SIL_MODE	CMCRX	CMCRX_me	CanRx07_TSCI	CanRx08_TSCI	CanRx06_BatIn	TimerReantNorm	CanRxMpp0_PB	CanRxMpp1_PB	CanRx5	TimerRe	CanRx1	CanRx7?
(MessagesInput)																	
_MWR_IA_CMCRX_1																	←
_MWR_IA_CMCRX_2														←			
_MWR_IA_CMCRX_3																	←
_MWR_IA_CMCRX_4							←										
_MWR_IA_CMCRX_5								←									
_MWR_IA_CMCRX_6									←								
_MWR_IA_CMCRX_7													←				
_MWR_IA_CMCRX_8													←				



Filter Status

Top:	
Focused Scope:	MWRI_model.slx
Left:	
Focused Scope:	MWRI_Request.slreqx

	MWRI_model	MIL_MODE	CODEGEN_MODE	SIL_MODE	CMCRX	CMCRX_mc	CanRx87_TSCI	CanRx88_TSCI	CanRx96_BatIn	TimerReadNorm	CanRxMp0_PB	CanRxMp1_PB	CanRx5	TimerRe	CanRx1	CanRx7
(MessagesInput)																
_MWR_IA_CMCRX_1																↙
_MWR_IA_CMCRX_2													↙			
_MWR_IA_CMCRX_3																↙
_MWR_IA_CMCRX_4							↙									
_MWR_IA_CMCRX_5								↙								
_MWR_IA_CMCRX_6									↙							
_MWR_IA_CMCRX_7										↙						
_MWR_IA_CMCRX_8											↙					



Contacts

www.teoresigroup.com



Job Teoresi – job@teoresigroup.com
Thesis & Internship – thesisandinternships@teoresigroup.com

Alberto Bertone – alberto.bertone@teoresigroup.com
Paolo Bizzarri – paolo.bizzarri@teoresigroup.com
Massimiliano Curti – massimiliano.curti@teoresigroup.com
Eugenio Romeo – eugenio.romeo@teoresigroup.com

Thank you!