

VERIFICATION
VALIDATION
METHODS

Final Event 21 / 22 November 2023

Test Execution & Test Evaluation

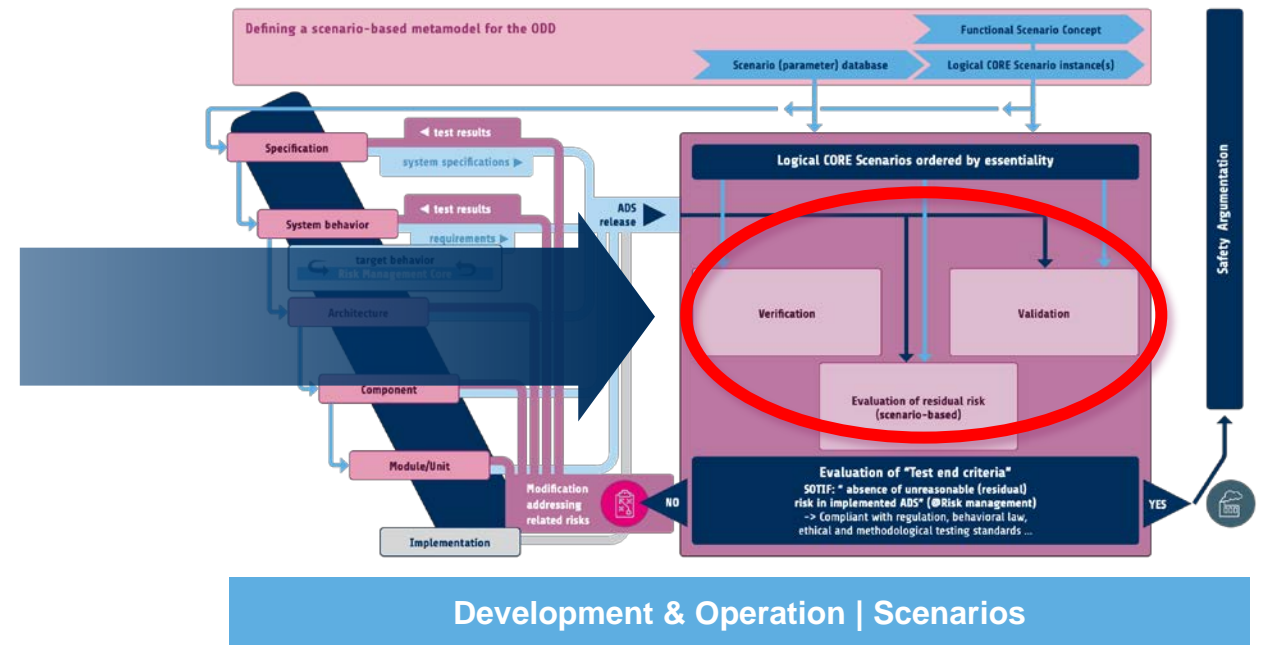
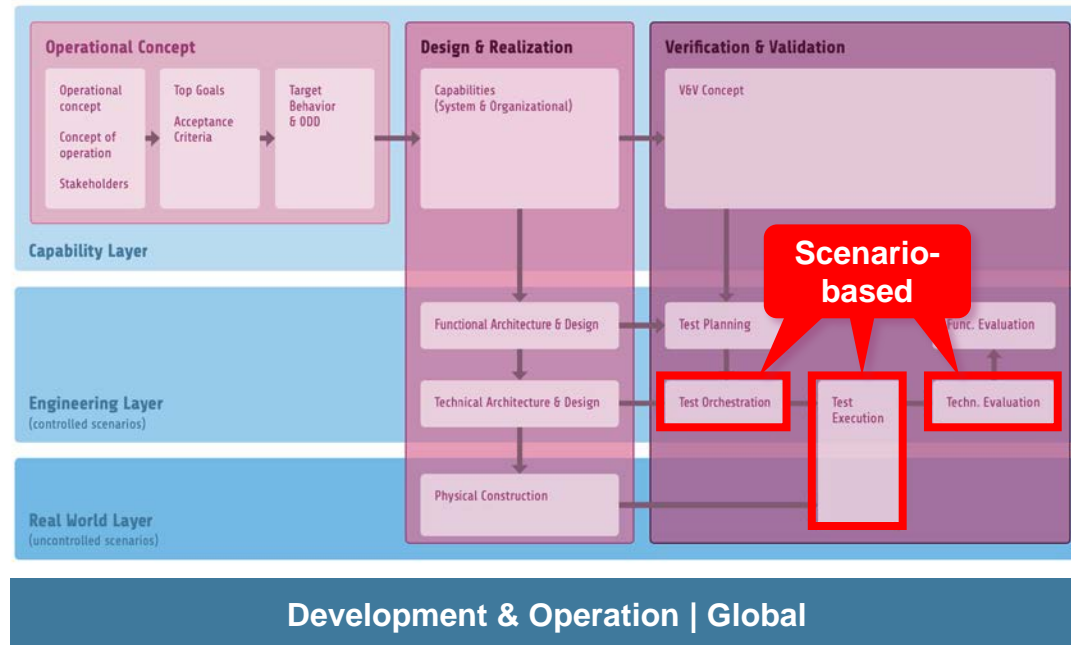
Helmut Schittenhelm, Mercedes-Benz

Supported by:



on the basis of a decision
by the German Bundestag

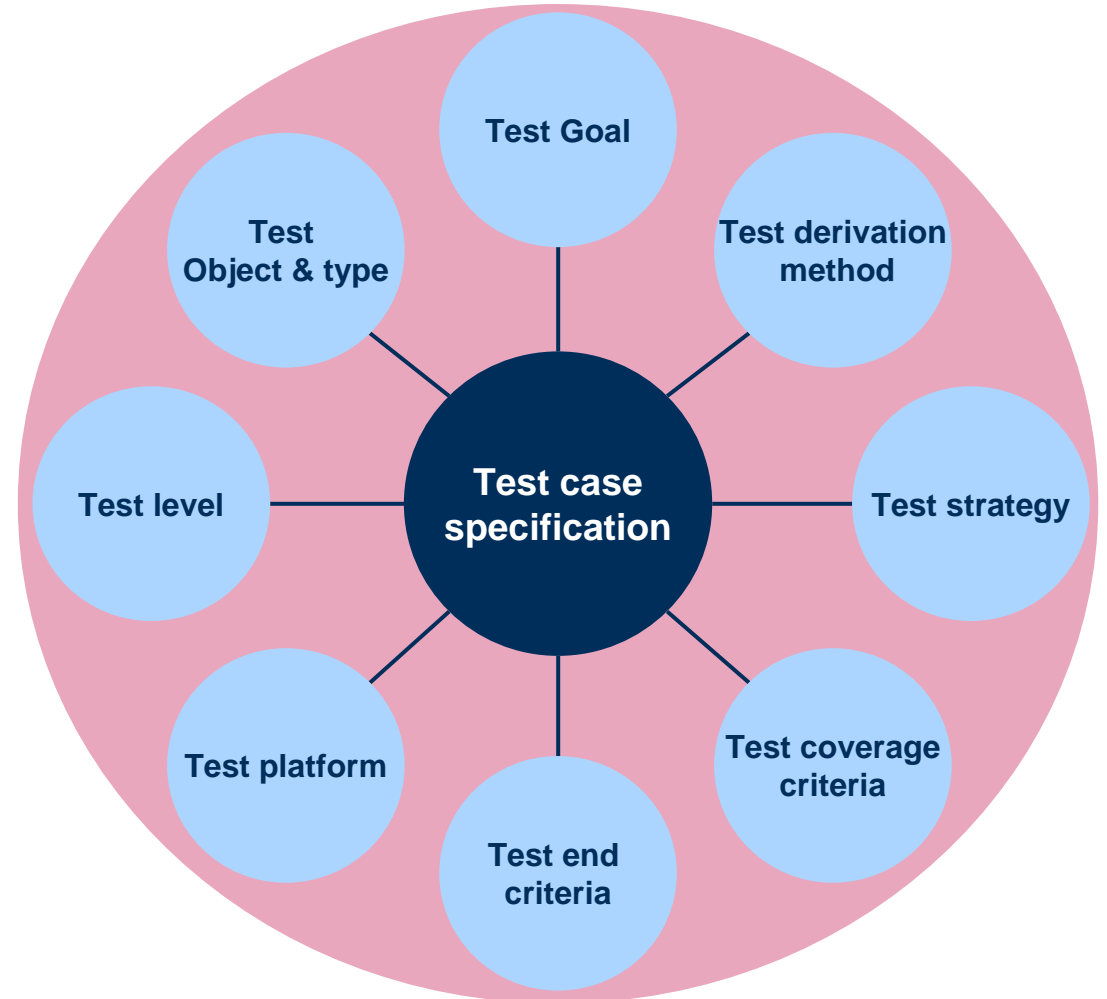
From Global to Scenario-based view



The **Verification concept** defines the approach of verifying of the system operations and the proof of the correct implementation of all requirements.

It defines what will be tested (**test objectives**), the purposes for which the test is intended to provide evidence of achievement (**test Goals**), the levels to which different system elements will be tested (**test Levels**), includes general specification of the test procedure to achieve the test objectives (**test strategies** and **the test derivation methods**, **test coverage** and **test end criteria** to be used, the test environments (**test Platforms**) used and a test matrix with detailed mapping connecting the testing performed to the system requirements.

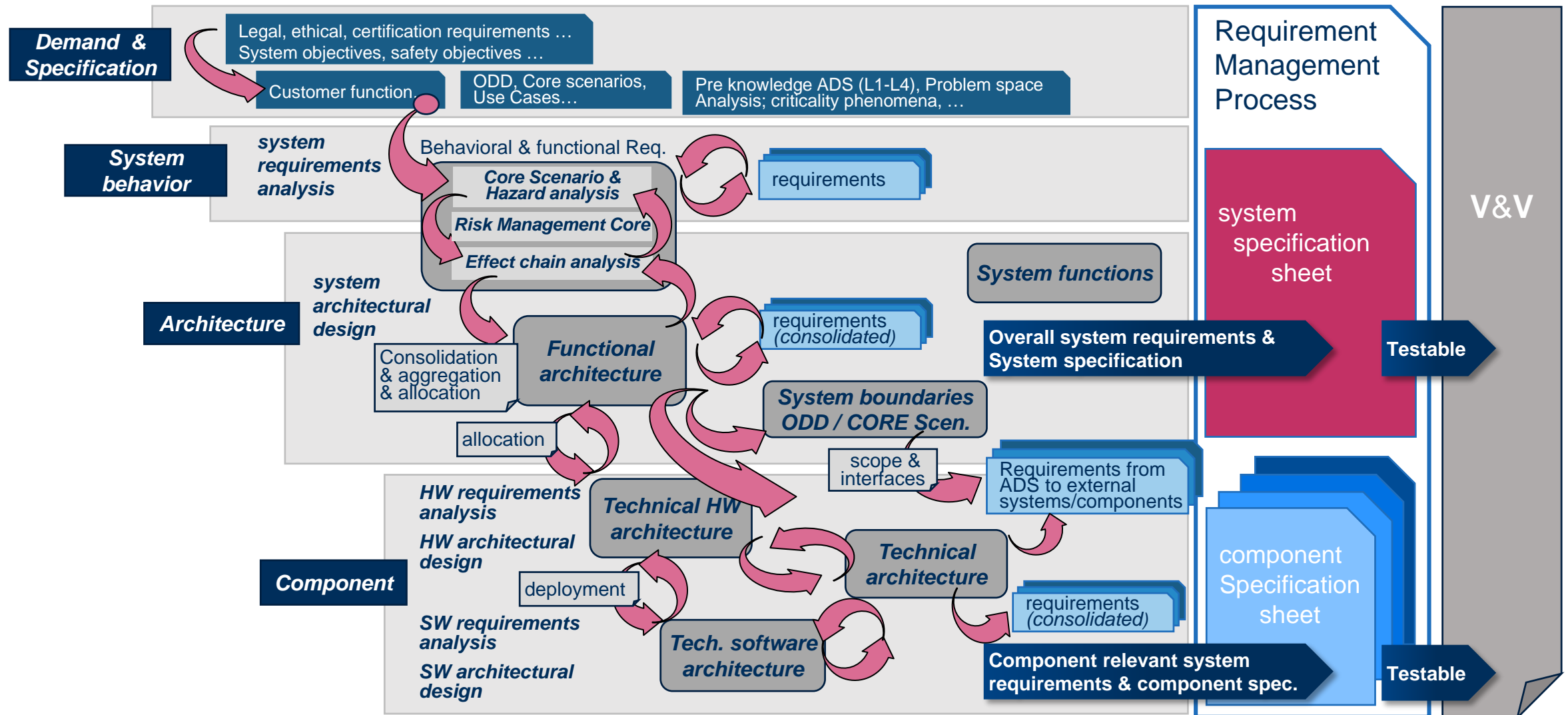
Furthermore, it ensures **conformity to the standards** ISO 26262 and 21448 and other common industrial ones.



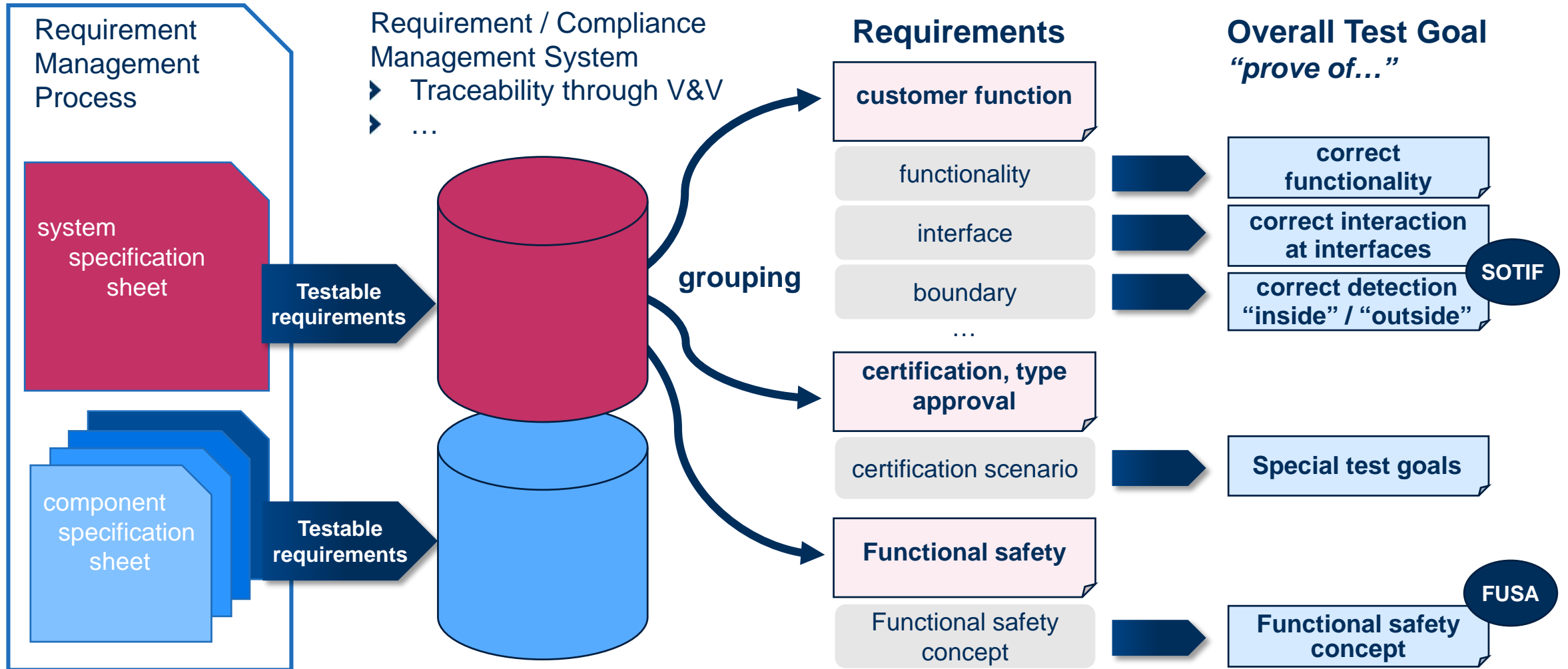
Detailed Test Case Specification

	Description	Planning
Initial test information	Test object (type), Test Goal, Test Levels & Plattform, Test Type, ..., responsibilities	X / -
Initial methodical info	CORE scenarios & relevant sub parameter space for test case coverage, scanning method, ...	x
Precondition	conditions which must be fulfilled before the test case is executed	x
Postcondition	conditions which are “ <i>valid</i> ” after the test case has been executed.	x
Test steps	<p>Test steps contain actions and usually also expected results.</p> <p>Init steps: steps necessary to establish a state in which test steps of type “run” can be executed, i.e. this is a preparation or setup for the “<i>run step</i>” execution.</p> <p>Run steps: main test steps of a test case. These establish the core testing activities for which the results have to be evaluated.</p> <p>Shutdown steps: if the test object has to be put (back) into a certain state after test execution, shutdown steps can be executed.</p>	x
Configuration parameter	Release level, parametrization of test object, (concrete) scenario parameter, ...	X/-
Pass / Fail criteria	Criteria that describes the expected result for an executed test. The description has to be in a form in which the test results can be judged as either “ <i>passed</i> ” or “ <i>failed</i> ”.	x
Abortion criteria	Criteria that describes the cancellation criteria apply to performed test level.	x
prioritization	State the set of rules you wish to apply to assign priority to test cases.	x
Linked requirements	links from a test case to those “ <i>requirements</i> ” which shall be tested by the test case.	x

Requirement Flow – inside ADS Design and to V&V



Requirement flow to “Test Planning”



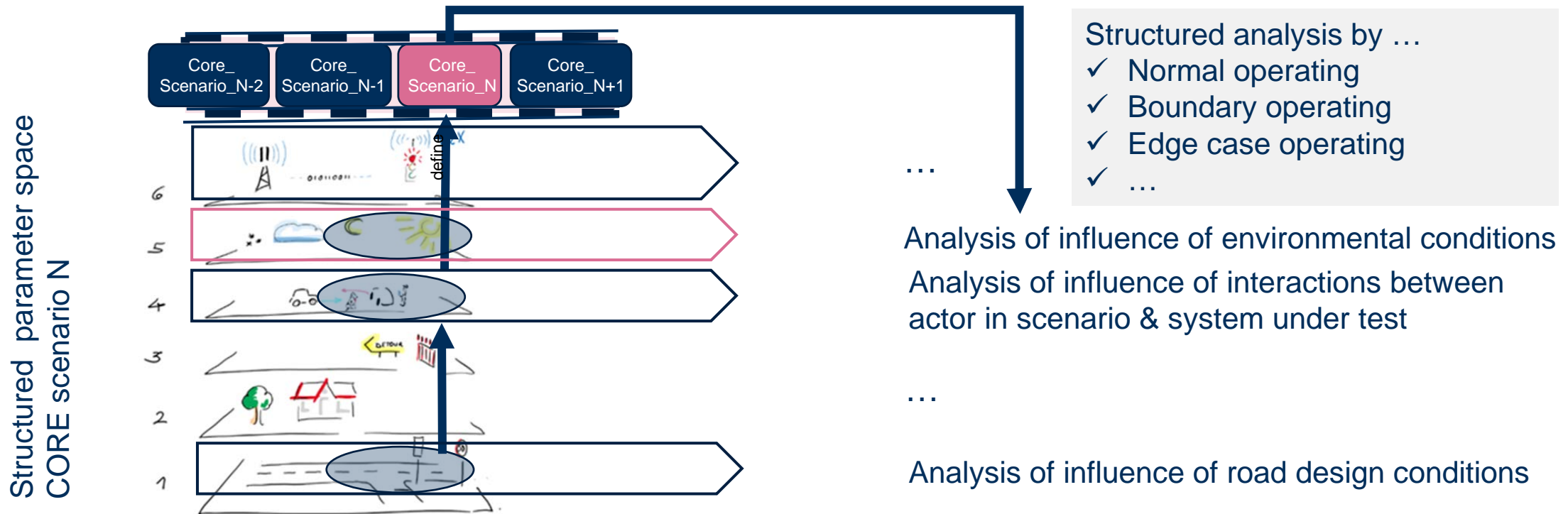
Test Goal and Test Case derivation in verification - Requirement based testing

Methods for test case derivation Test Goal (derived from Standards)	Requirement Analysis	Core Scenario analysis	Interface analysis
	<i>test case derivation based on requirements & specifications depending on normal & border operating, & error conditions.</i>	<i>test case derivation based on CORE scenario instances in varying parameters in layer (SUT & actors, weather, ...)</i>	<i>test case derivation based on external or internal interface specifications (e.g. HW/SW interface).</i>
SOTIF (exemplary)			
S.1: Proof that the system properly detects that it is “inside” ODD, where it is designed to perform.	X	X	
S.2: Proof that the system properly detects that it is “outside” ODD, where it is designed NOT to perform.	X	X	
Performance (exemplary)			
Prof that test object provides its required functionality with regard to time & up to limits of its tolerance range	X	X	
correct functionality (exemplary)			
Proof that implemented functionality is as specified (correct processing inputs & outputs).	X	X	
correct interaction at the interfaces			
Proof of (external & internal) interfaces as specified.	X		X

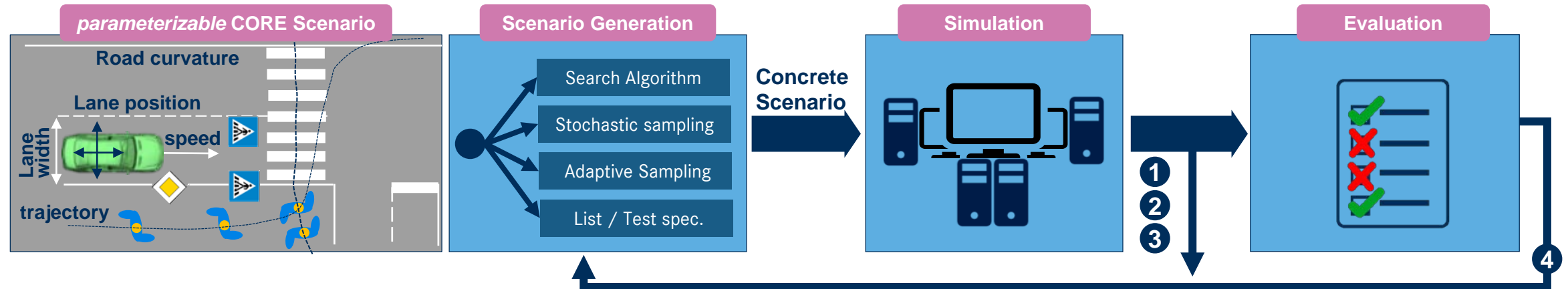
Scenario analysis / scenario-based verification / testing

The **test derivation method** analysis of CORE scenarios (instances) and their included use cases is a tool that enables a structured decomposition of parameter and treatment of analyzing their influence.

It could be used for generating test cases in verification for proving **general test goals** addressing “functionality”, “performance” or “usability”.

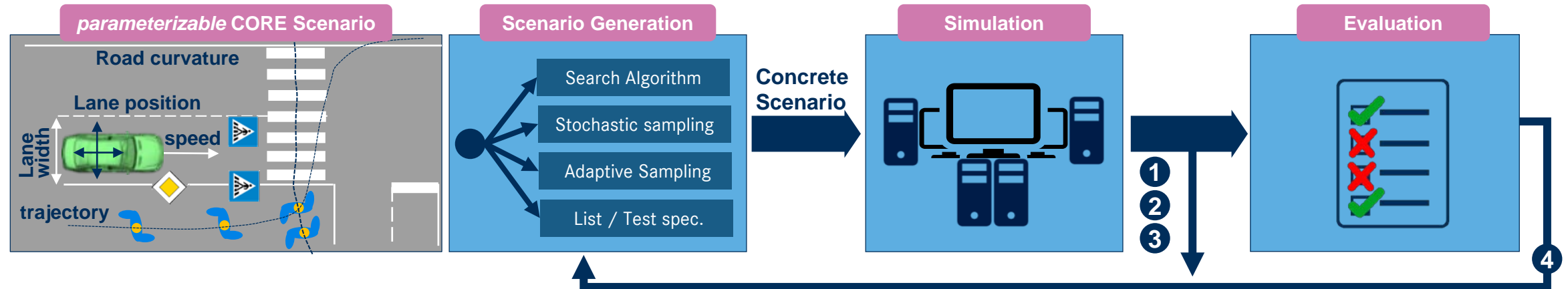


Scenario-based Testing - different characteristics, one approach



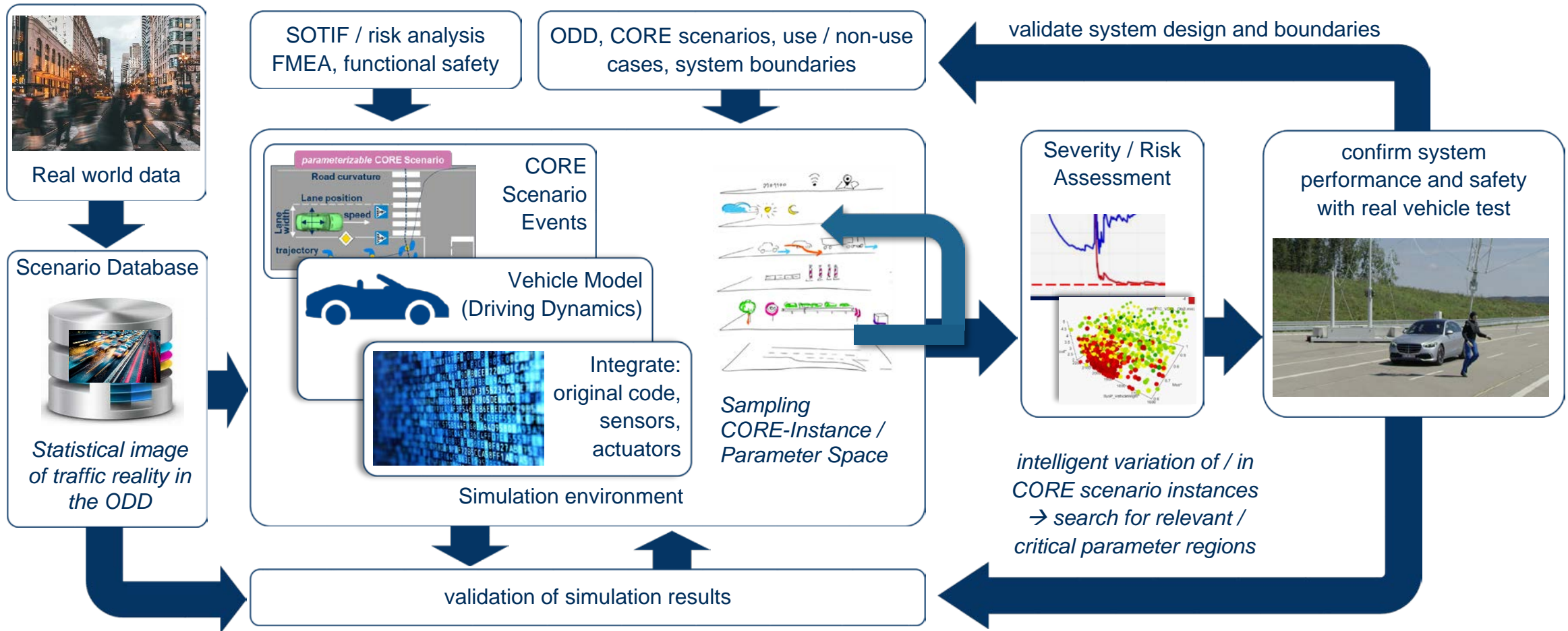
	Requirement-based Testing	Adaptive Sampling Testing	Monte Carlo-based Testing	Search-based Testing
Generation of sets of parameter for parametrize concrete scenarios	Concrete scenarios for fixed parameters, Requirement-based Scenario Creation	predefined plan for Sampling, systematic coverage of parameter Sub-space	Random sampling of the parameter space (open sampling loop)	Optimizer algorithm samples parameter space (closed sampling loop)
Number of scenarios	few	some <i>up to</i> large	(very) large	immense
Parameter distributions from real-world	-	-	Parameter distribution (Scenario Data-Base)	Parameter distribution (Scenario Data-Base)
Used in / for	Verification (of Requirement in related scenario)	Verification (of Requirement in related scenario)	Exploration of <i>unsafe</i> (areas of) unknowns	Residual Risk evaluation

Scenario-based Testing - fields of application



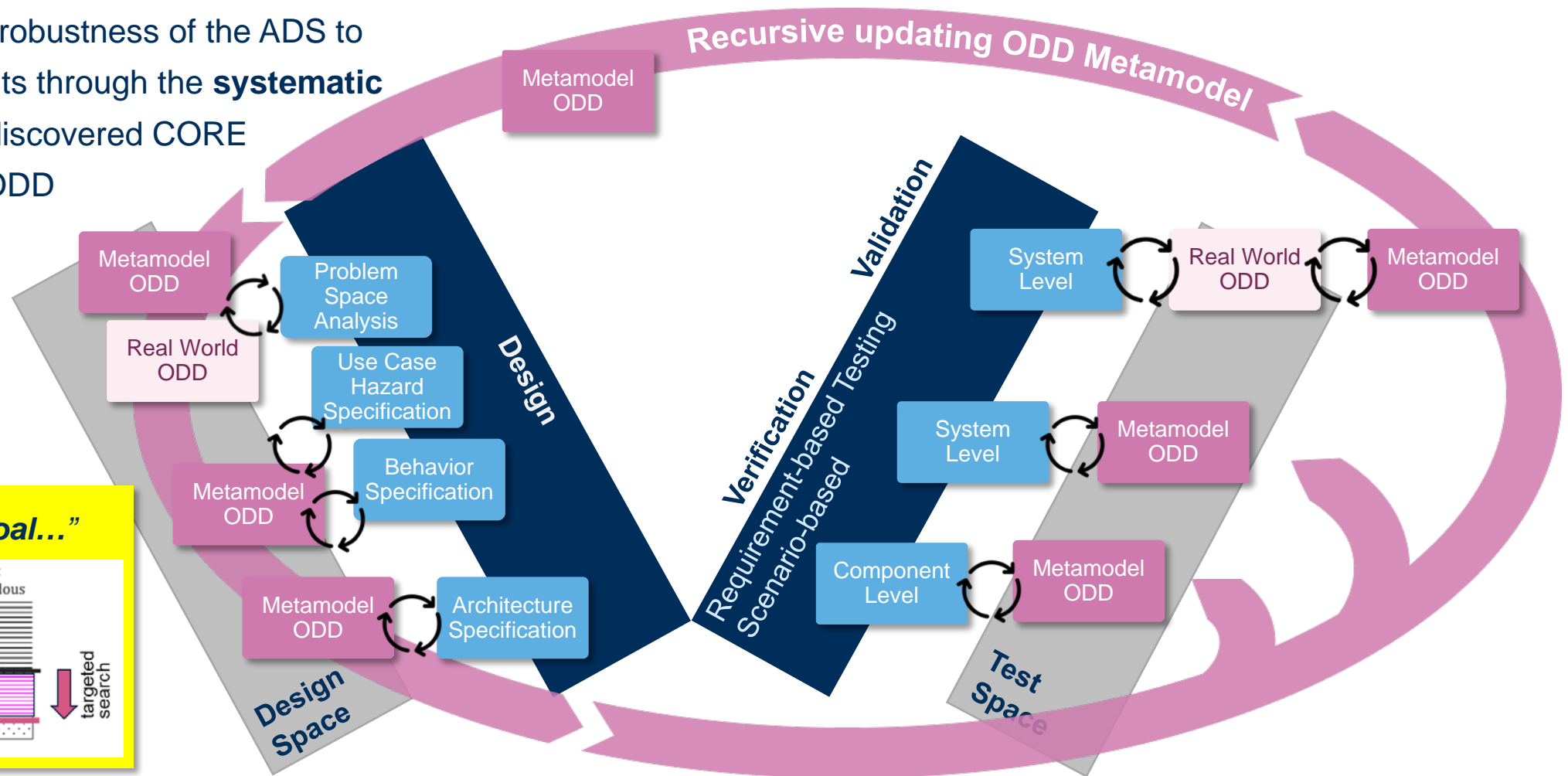
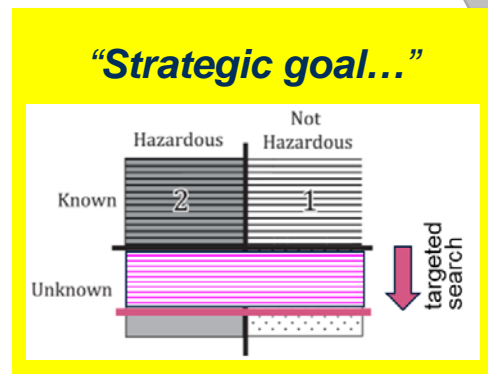
Test Object Platform	SIL	HIL	Proving Ground	Field Operational Test
Sense: Perception	-	(X)	X	X
Sense: Fusion	X	X	X	X
Plan: behavior	X	X	X	X
Sense / Plan / Act	X / X	X	X	X
Type of use	scenario-based closed loop tests scenario-based with random traffic closed loop	scenario-based open and closed- loop & integration tests	scenario-based open and closed- loop tests with ADS equipped vehicles	random real-world closed loop

Search-base testing – residual risk evaluation



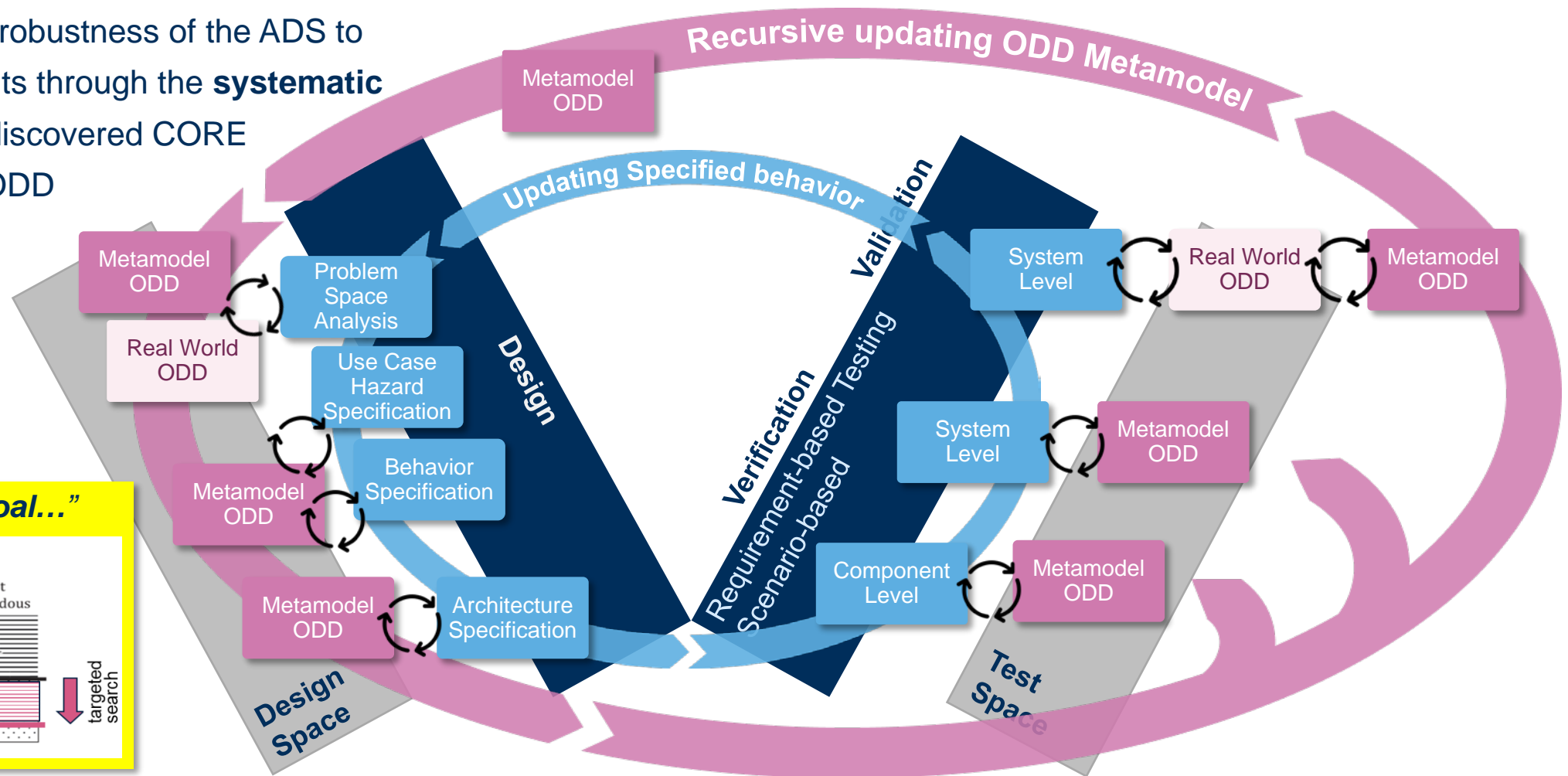
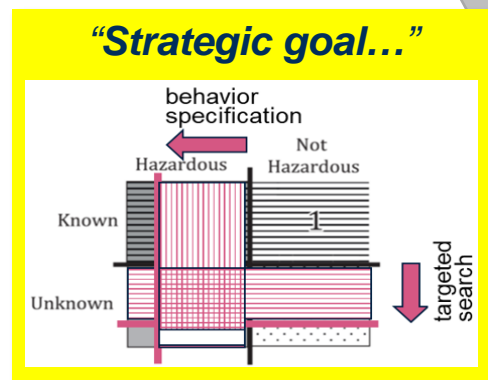
The ODD Metamodel

... increase the robustness of the ADS to hazardous events through the **systematic integration** of discovered CORE scenarios and ODD characteristics into the ODD Metamodel...

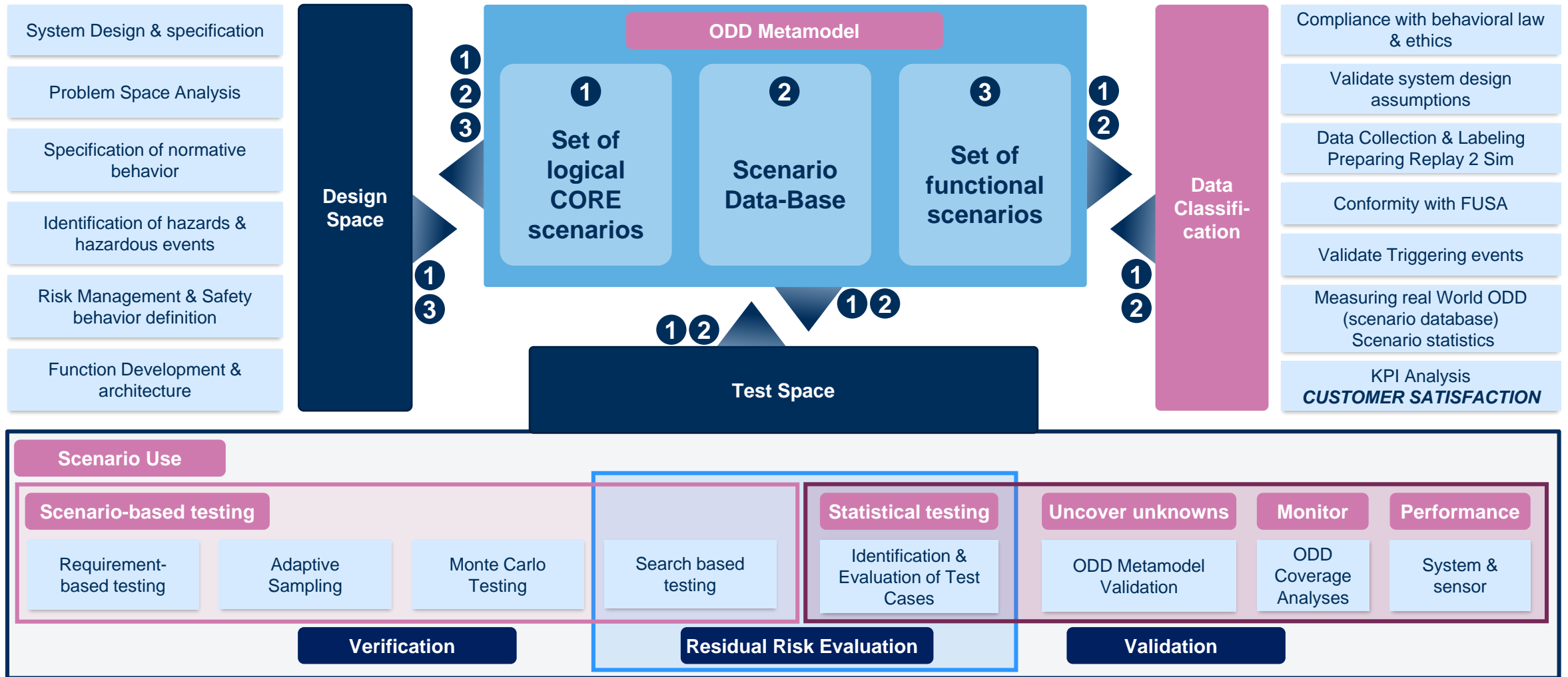


The ODD Metamodel

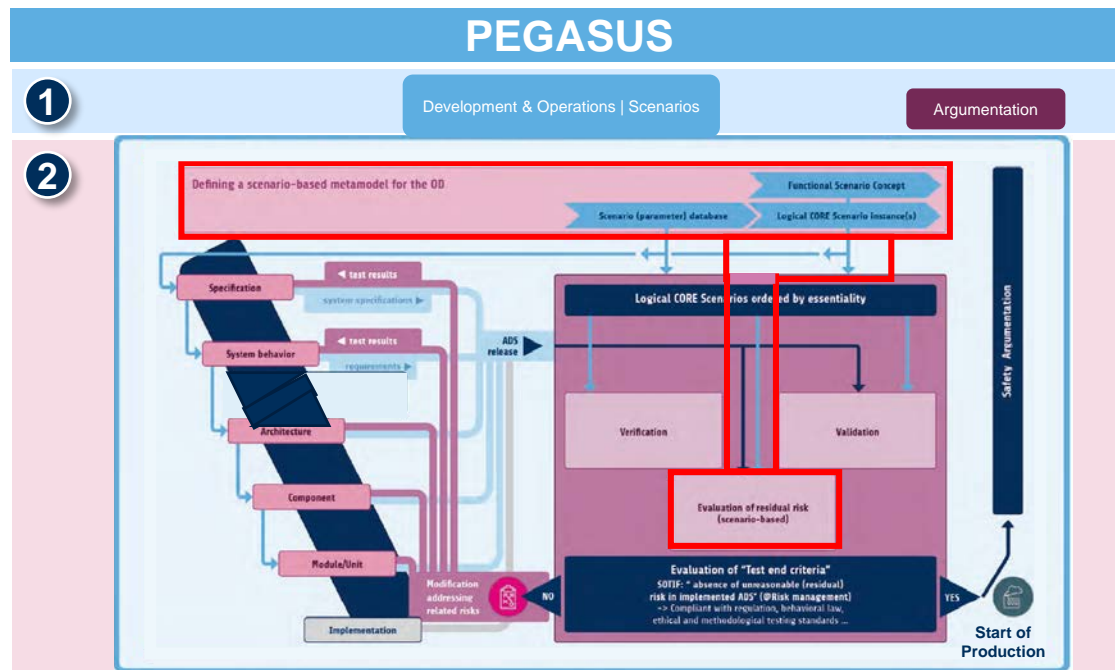
... increase the robustness of the ADS to hazardous events through the **systematic integration** of discovered CORE scenarios and ODD characteristics into the ODD Metamodel...



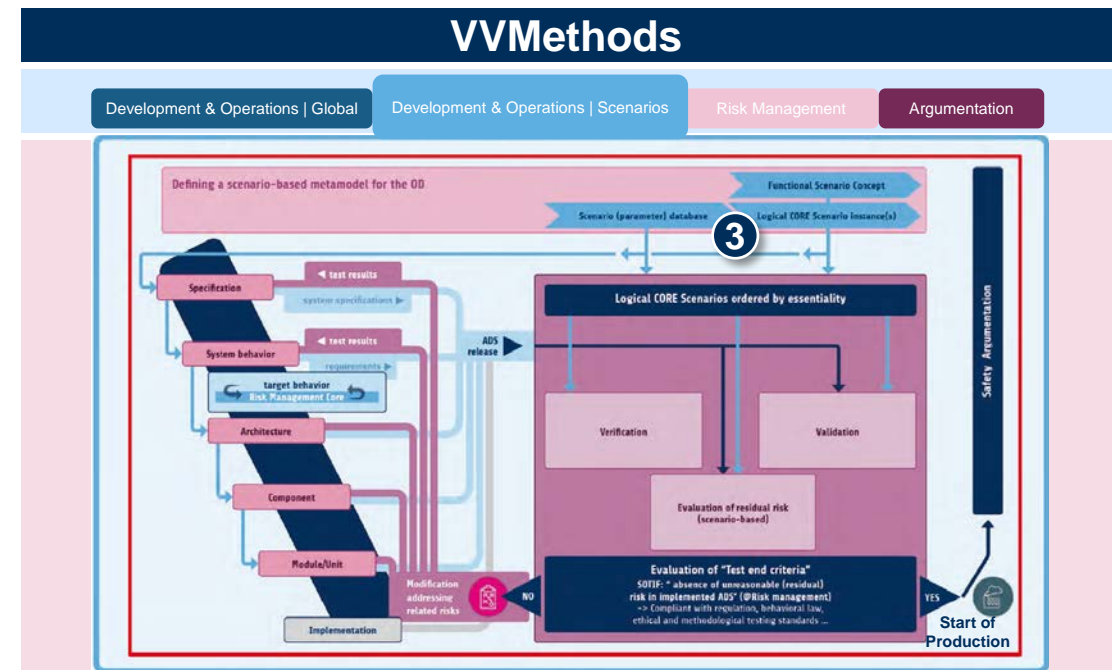
ODD Metamodel - one common, consistent ODD description – *“ADS is designed and tested on a valid model of the (real World) ODD”*



PEGASUS Approach – vs – VVMethods Approach



- 1 GSN-based argumentation
- 2 6 Layer Scenario model, Set of Logical Scenarios
- Scenario-based testing with risk evaluation in V&V
- Use of test instances: Simulation first, PG confirms simulation, endurance run assures stochastic aspects & complex situation



- Framework-based Argumentation including risk management
- 6 Layer Model & ODD Metamodel (set of CORE scenarios)
- Scenario-based behavior specification (ADF / ADS design)
- Scenario-based verification & validation
- Scenario-based risk evaluation in V&V
- Simulation first, PG confirms simulation, endurance run assures stochastic aspects & complex situation & validates Metamodel

Thank you!

Helmut Schittenhelm, Mercedes-Benz

helmut.schittenhelm@mercedes-benz.com



A project developed by the VDA Leitinitiative
autonomous and connected driving

Supported by:



on the basis of a decision
by the German Bundestag