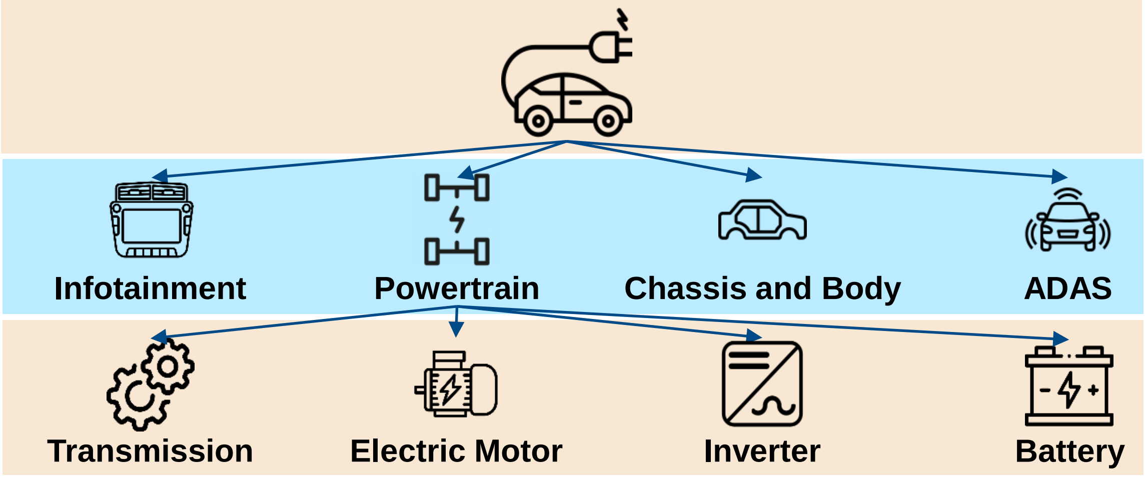
MathWorks
AUTOMOTIVE CONFERENCE 2024
Korea

모델 기반 설계에서의 ASPICE 준수방안

류성연 프로, MathWorks

MathWorks

Development of E/E Automotive Systems



Infotainment

Powertrain

Chassis and Body

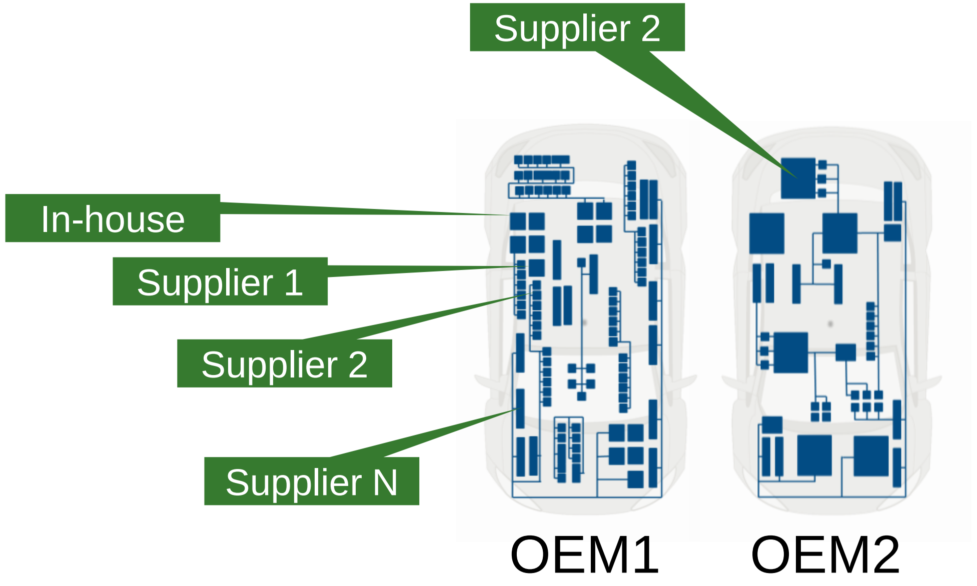
ADAS

Transmission

Electric Motor

Inverter

Battery



Supplier 2

In-house

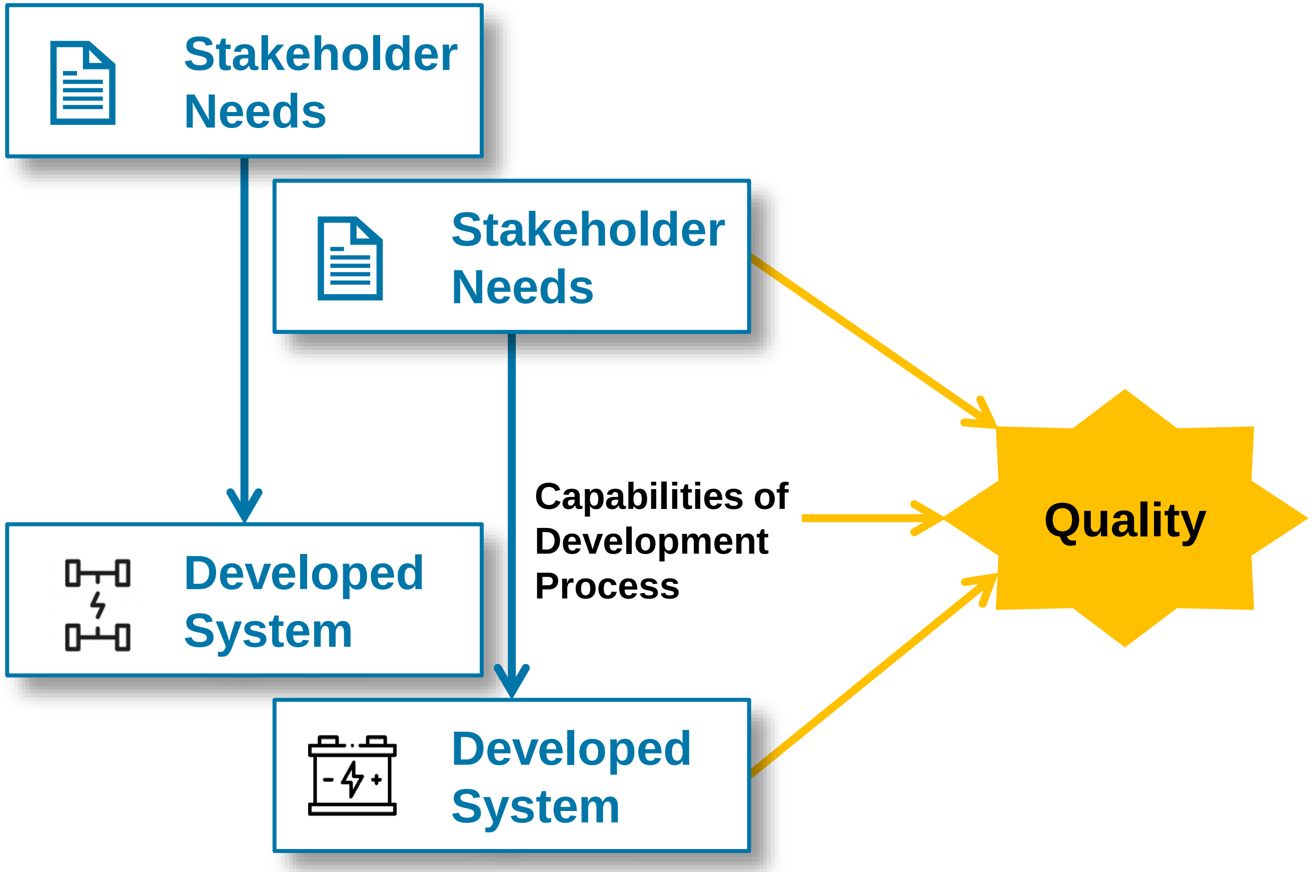
Supplier 1

Supplier 2

Supplier N

OEM1

OEM2



Stakeholder Needs

Stakeholder Needs

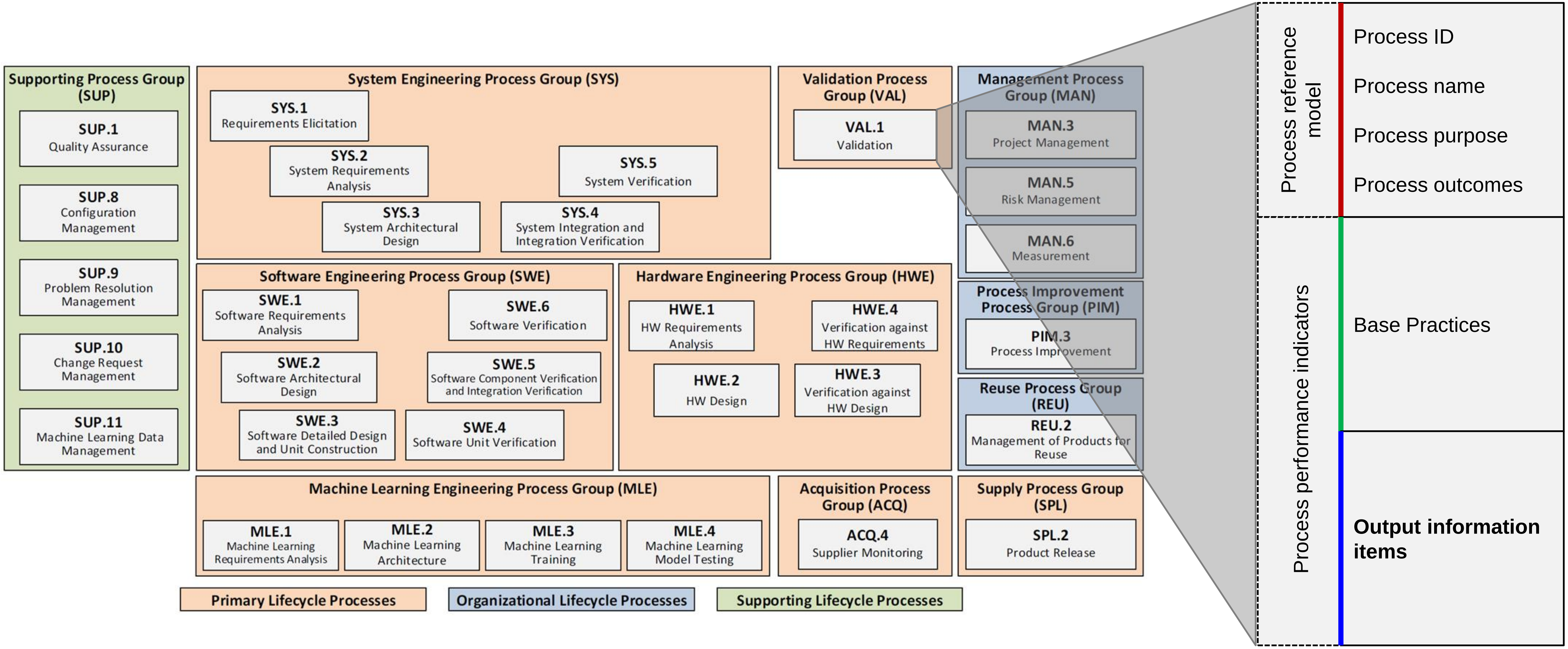
Developed System

Developed System

Capabilities of Development Process

Quality

Automotive SPICE® Process Reference Model



Supporting Process Group (SUP)

- SUP.1 Quality Assurance
- SUP.8 Configuration Management
- SUP.9 Problem Resolution Management
- SUP.10 Change Request Management
- SUP.11 Machine Learning Data Management

System Engineering Process Group (SYS)

- SYS.1 Requirements Elicitation
- SYS.2 System Requirements Analysis
- SYS.3 System Architectural Design
- SYS.4 System Integration and Integration Verification
- SYS.5 System Verification

Validation Process Group (VAL)

- VAL.1 Validation

Management Process Group (MAN)

- MAN.3 Project Management
- MAN.5 Risk Management
- MAN.6 Measurement

Software Engineering Process Group (SWE)

- SWE.1 Software Requirements Analysis
- SWE.2 Software Architectural Design
- SWE.3 Software Detailed Design and Unit Construction
- SWE.4 Software Unit Verification
- SWE.5 Software Component Verification and Integration Verification
- SWE.6 Software Verification

Hardware Engineering Process Group (HWE)

- HWE.1 HW Requirements Analysis
- HWE.2 HW Design
- HWE.3 Verification against HW Design
- HWE.4 Verification against HW Requirements

Process Improvement Process Group (PIM)

- PIM.3 Process Improvement

Reuse Process Group (REU)

- REU.2 Management of Products for Reuse

Machine Learning Engineering Process Group (MLE)

- MLE.1 Machine Learning Requirements Analysis
- MLE.2 Machine Learning Architecture
- MLE.3 Machine Learning Training
- MLE.4 Machine Learning Model Testing

Acquisition Process Group (ACQ)

- ACQ.4 Supplier Monitoring

Supply Process Group (SPL)

- SPL.2 Product Release

Primary Lifecycle Processes

Organizational Lifecycle Processes

Supporting Lifecycle Processes

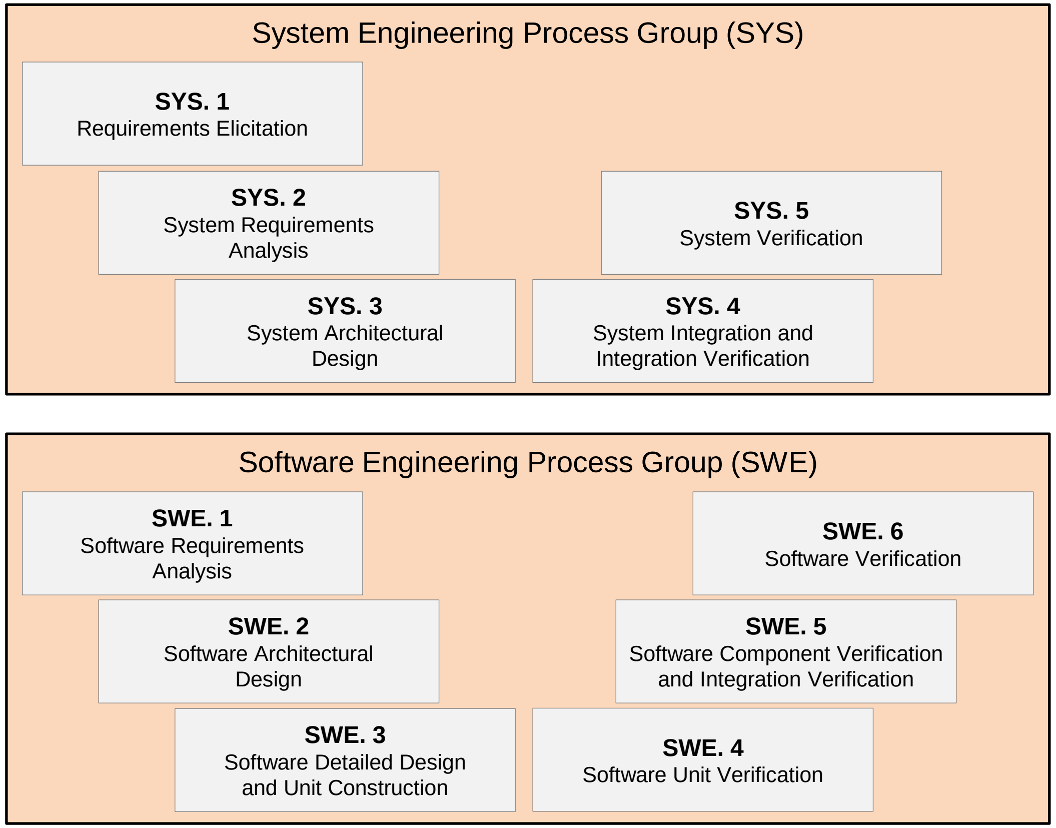
Process reference model

- Process ID
- Process name
- Process purpose
- Process outcomes

Process performance indicators

- Base Practices
- **Output information items**

Today’s Agenda



System Engineering Process Group (SYS)

SYS. 1
Requirements Elicitation

SYS. 2
System Requirements Analysis

SYS. 3
System Architectural Design

SYS. 4
System Integration and Integration Verification

SYS. 5
System Verification

Software Engineering Process Group (SWE)

SWE. 1
Software Requirements Analysis

SWE. 2
Software Architectural Design

SWE. 3
Software Detailed Design and Unit Construction

SWE. 4
Software Unit Verification

SWE. 5
Software Component Verification and Integration Verification

SWE. 6
Software Verification

For E/E Automotive Systems Development…

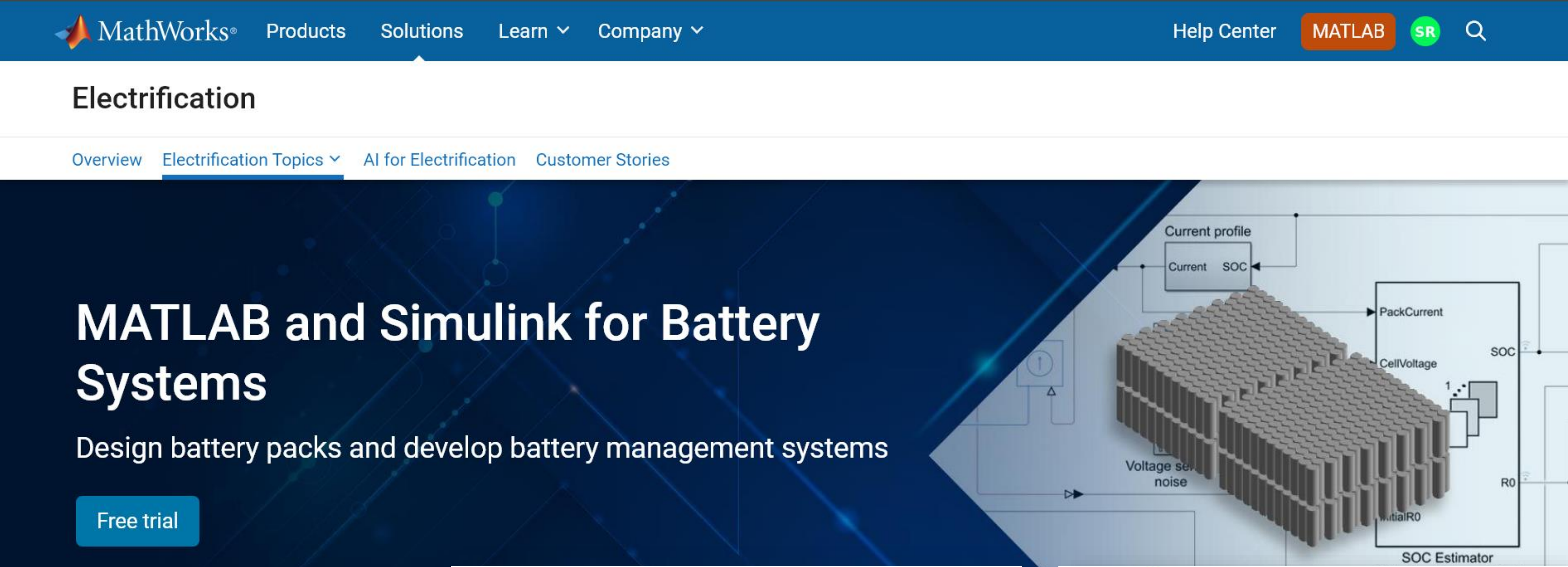
- Many automotive standards for production

Enhance efficiency through MBD

ASPICE

AUTOSAR

ISO26262



Electrification

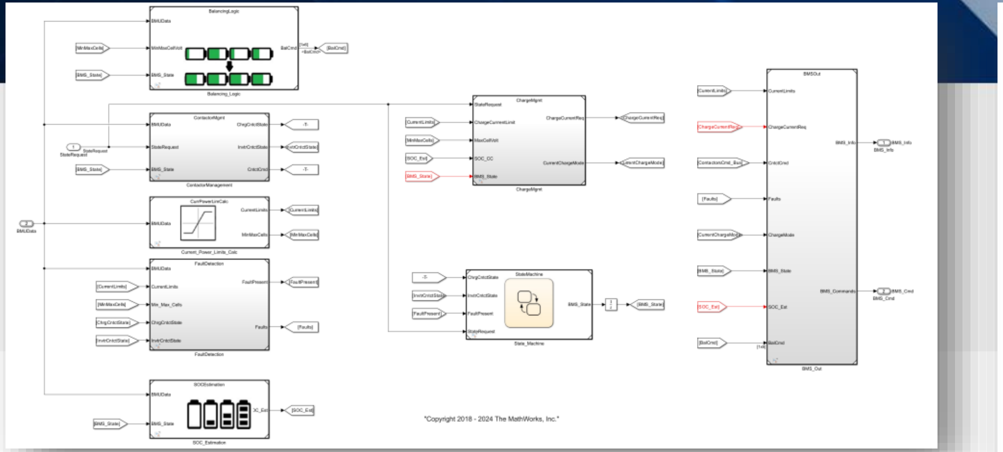
Overview | Electrification Topics | AI for Electrification | Customer Stories

MATLAB and Simulink for Battery Systems

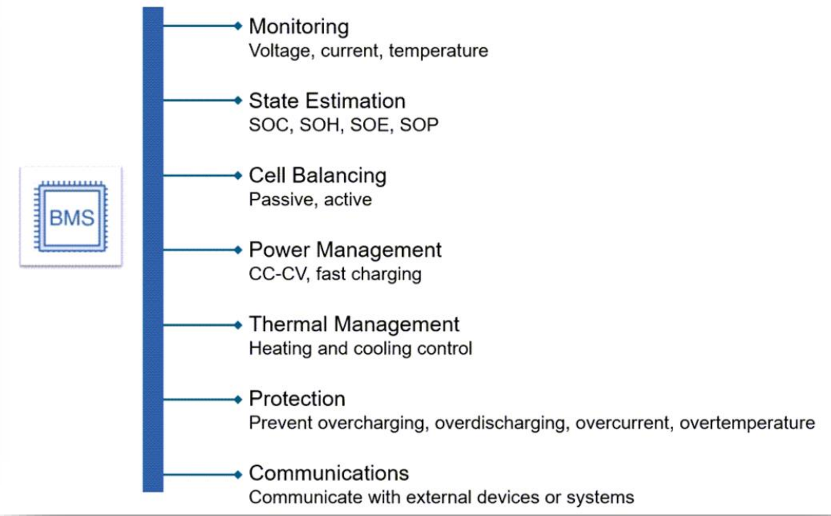
Design battery packs and develop battery management systems

Free trial

Demo application:



"Copyright 2018 - 2024 The MathWorks, Inc."



Monitoring
Voltage, current, temperature

State Estimation
SOC, SOH, SOE, SOP

Cell Balancing
Passive, active

Power Management
CC-CV, fast charging

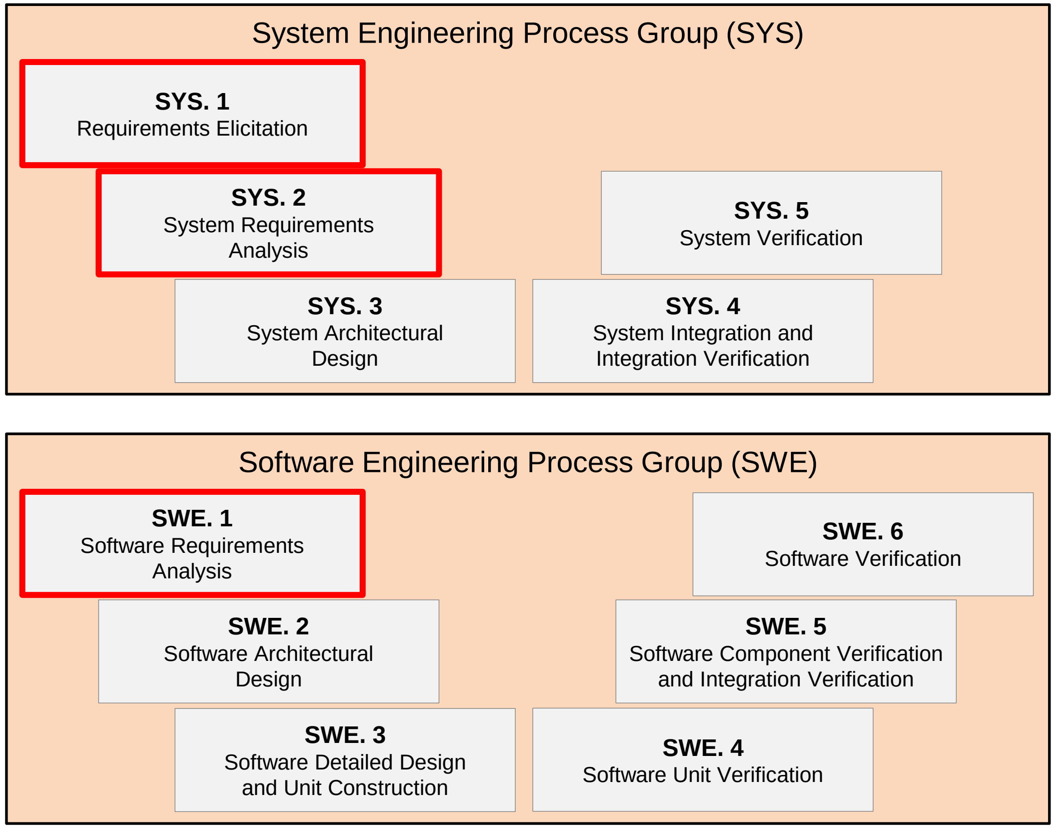
Thermal Management
Heating and cooling control

Protection
Prevent overcharging, overdischarging, overcurrent, overtemperature

Communications
Communicate with external devices or systems

https://www.mathworks.com/solutions/electrification/battery-systems.html

Requirements Management



System Engineering Process Group (SYS)

SYS. 1
Requirements Elicitation

SYS. 2
System Requirements Analysis

SYS. 3
System Architectural Design

SYS. 4
System Integration and Integration Verification

SYS. 5
System Verification

Software Engineering Process Group (SWE)

SWE. 1
Software Requirements Analysis

SWE. 2
Software Architectural Design

SWE. 3
Software Detailed Design and Unit Construction

SWE. 4
Software Unit Verification

SWE. 5
Software Component Verification and Integration Verification

SWE. 6
Software Verification

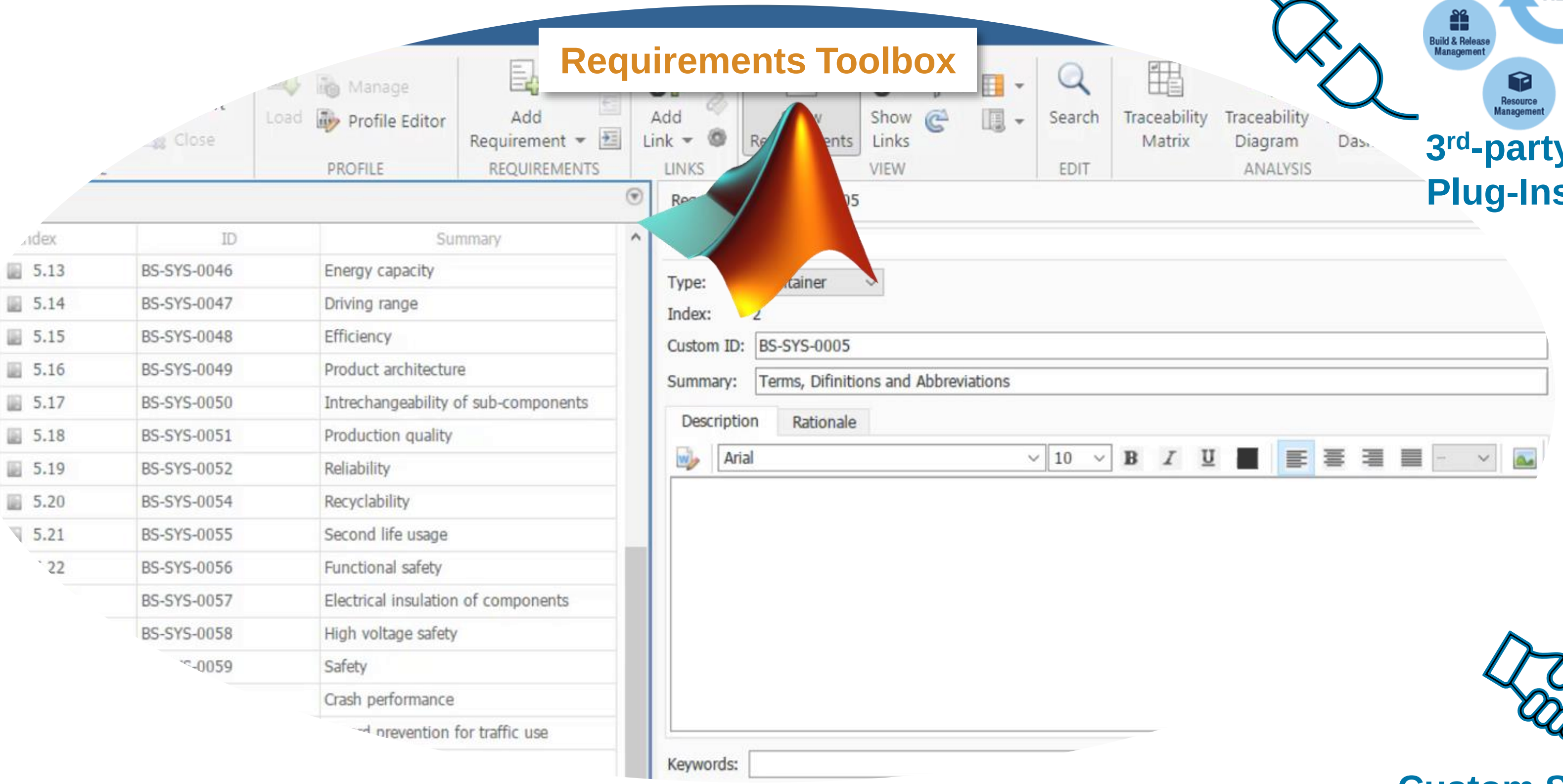
Why Traceability Matters for ASPICE

Digital Thread

- **Completeness** and **Consistency** are the top challenges
  - **Completeness:** all required functionality is defined
  - **Consistency:** requirements do not conflict
- Ensure application is complete, fully tested, and meets customer requirements
- Understand the impact of requirement changes to implementation and test

Connect the Requirements Toolbox with External Sources and Tools

Import-Export & Roundtrips



Requirements Toolbox

Index	ID	Summary
5.13	BS-SYS-0046	Energy capacity
5.14	BS-SYS-0047	Driving range
5.15	BS-SYS-0048	Efficiency
5.16	BS-SYS-0049	Product architecture
5.17	BS-SYS-0050	Intrechangeability of sub-components
5.18	BS-SYS-0051	Production quality
5.19	BS-SYS-0052	Reliability
5.20	BS-SYS-0054	Recyclability
5.21	BS-SYS-0055	Second life usage
22	BS-SYS-0056	Functional safety
	BS-SYS-0057	Electrical insulation of components
	BS-SYS-0058	High voltage safety
	-0059	Safety
		Crash performance
		prevention for traffic use

Type: tainer

Index: 2

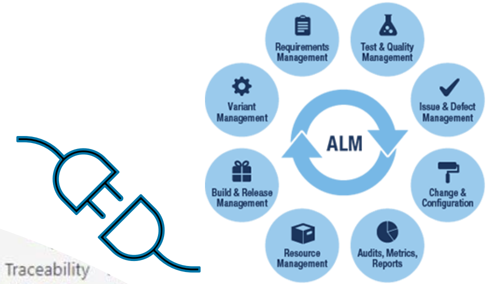
Custom ID: BS-SYS-0005

Summary: Terms, Difinitions and Abbreviations

Description | Rationale

Arial 10

Keywords:

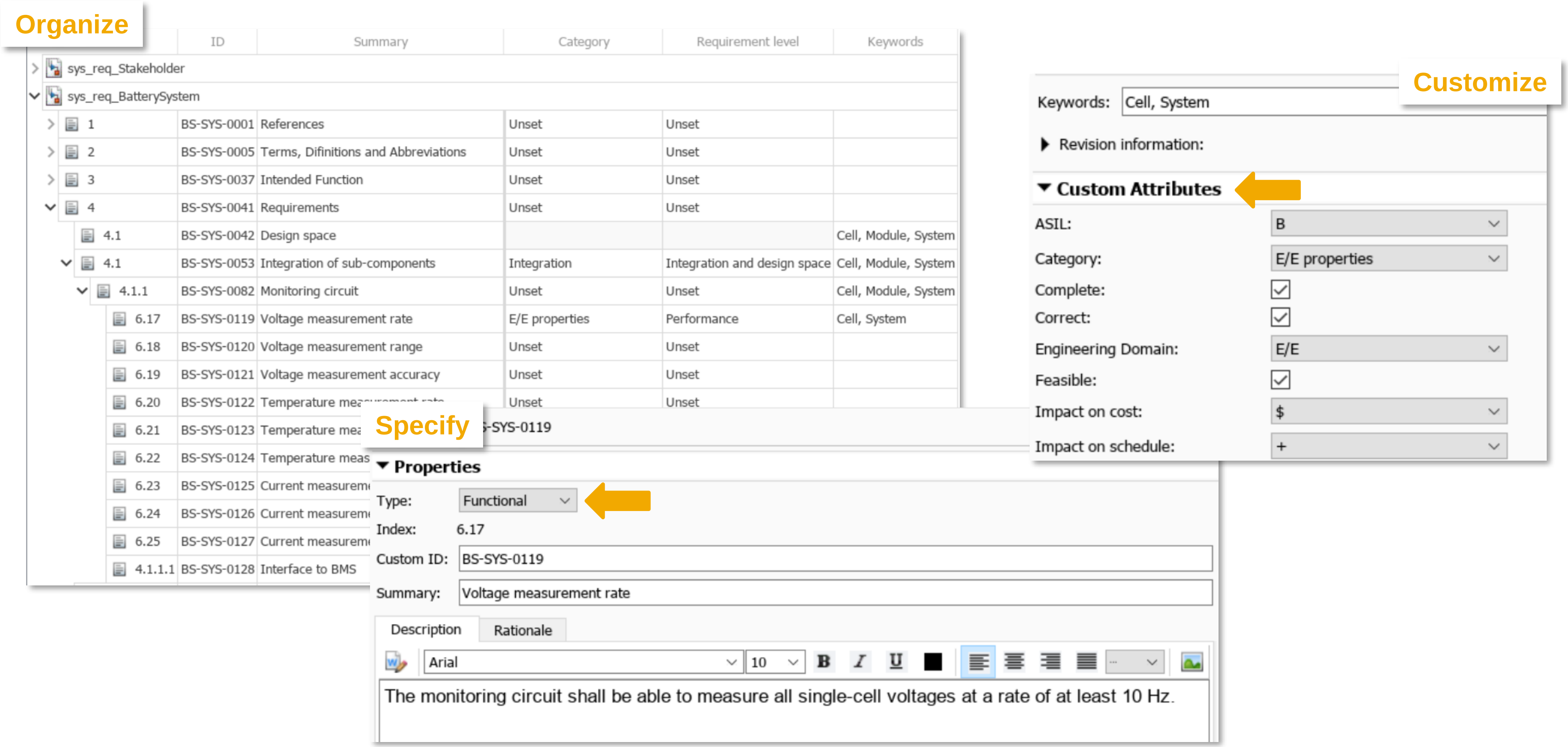


3rd-party Plug-Ins

Custom Solutions supported by MathWorks Consulting Services

Organize, Specify and Customize Requirements

Requirements Toolbox



Organize

	ID	Summary	Category	Requirement level	Keywords
sys_req_Stakeholder					
sys_req_BatterySystem					
1	BS-SYS-0001	References	Unset	Unset	
2	BS-SYS-0005	Terms, Difinitions and Abbreviations	Unset	Unset	
3	BS-SYS-0037	Intended Function	Unset	Unset	
4	BS-SYS-0041	Requirements	Unset	Unset	
4.1	BS-SYS-0042	Design space			Cell, Module, System
4.1	BS-SYS-0053	Integration of sub-components	Integration	Integration and design space	Cell, Module, System
4.1.1	BS-SYS-0082	Monitoring circuit	Unset	Unset	Cell, Module, System
6.17	BS-SYS-0119	Voltage measurement rate	E/E properties	Performance	Cell, System
6.18	BS-SYS-0120	Voltage measurement range	Unset	Unset	
6.19	BS-SYS-0121	Voltage measurement accuracy	Unset	Unset	
6.20	BS-SYS-0122	Temperature measurement rate	Unset	Unset	
6.21	BS-SYS-0123	Temperature mea			
6.22	BS-SYS-0124	Temperature meas			
6.23	BS-SYS-0125	Current measurem			
6.24	BS-SYS-0126	Current measurem			
6.25	BS-SYS-0127	Current measurem			
4.1.1.1	BS-SYS-0128	Interface to BMS			

Specify

S-SYS-0119

Properties

Type: Functional

Index: 6.17

Custom ID: BS-SYS-0119

Summary: Voltage measurement rate

The monitoring circuit shall be able to measure all single-cell voltages at a rate of at least 10 Hz.

Customize

Keywords: Cell, System

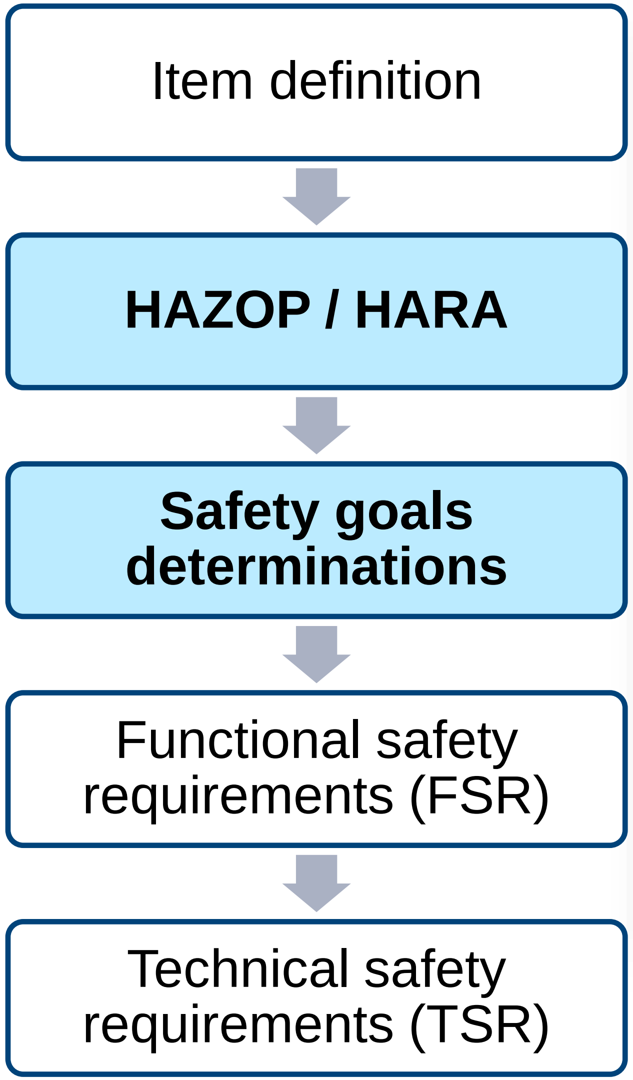
Revision information:

Custom Attributes

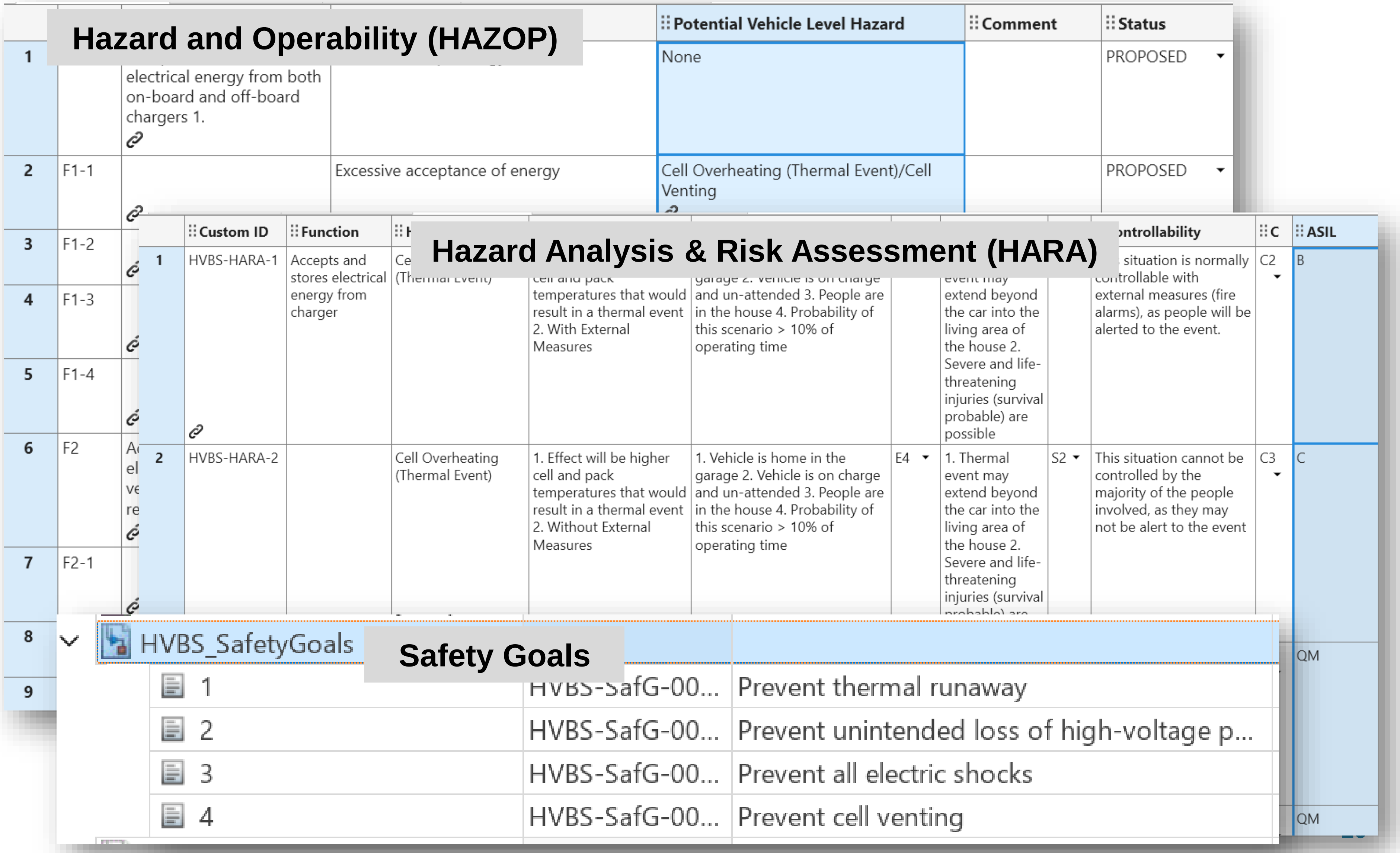
ASIL:	B
Category:	E/E properties
Complete:	☑
Correct:	☑
Engineering Domain:	E/E
Feasible:	☑
Impact on cost:	$
Impact on schedule:	+

Functional Safety Requirements from Concept Phase

Simulink Fault Analyzer



Item definition → HAZOP / HARA → Safety goals determinations → Functional safety requirements (FSR) → Technical safety requirements (TSR)



Hazard and Operability (HAZOP)

				Potential Vehicle Level Hazard	Comment	Status
1		electrical energy from both on-board and off-board chargers 1.		None		PROPOSED
2	F1-1		Excessive acceptance of energy	Cell Overheating (Thermal Event)/Cell Venting		PROPOSED
3	F1-2					
4	F1-3					
5	F1-4					
6	F2	Ac el ve re				
7	F2-1					
8						
9						

Hazard Analysis & Risk Assessment (HARA)

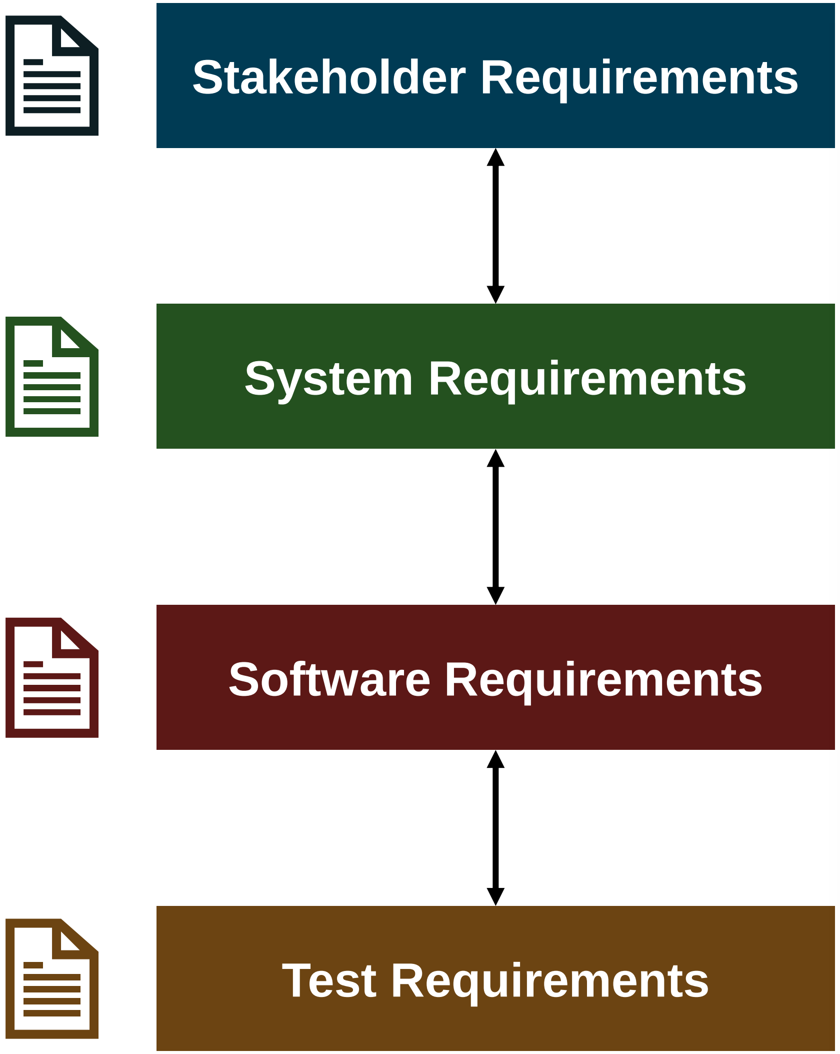
	Custom ID	Function	H						ontrollability	C	ASIL
1	HVBS-HARA-1	Accepts and stores electrical energy from charger	Ce (Thermal Event)	cell and pack temperatures that would result in a thermal event 2. With External Measures	garage 2. Vehicle is on charge and un-attended 3. People are in the house 4. Probability of this scenario > 10% of operating time		event may extend beyond the car into the living area of the house 2. Severe and life-threatening injuries (survival probable) are possible		situation is normally controllable with external measures (fire alarms), as people will be alerted to the event.	C2	B
2	HVBS-HARA-2		Cell Overheating (Thermal Event)	1. Effect will be higher cell and pack temperatures that would result in a thermal event 2. Without External Measures	1. Vehicle is home in the garage 2. Vehicle is on charge and un-attended 3. People are in the house 4. Probability of this scenario > 10% of operating time	E4	1. Thermal event may extend beyond the car into the living area of the house 2. Severe and life-threatening injuries (survival probable) are	S2	This situation cannot be controlled by the majority of the people involved, as they may not be alert to the event	C3	C
											QM
											QM

Safety Goals

HVBS_SafetyGoals

1	HVBS-SafG-00...	Prevent thermal runaway
2	HVBS-SafG-00...	Prevent unintended loss of high-voltage p...
3	HVBS-SafG-00...	Prevent all electric shocks
4	HVBS-SafG-00...	Prevent cell venting

Elicit and Elaborate Requirements through Bi-directional Links

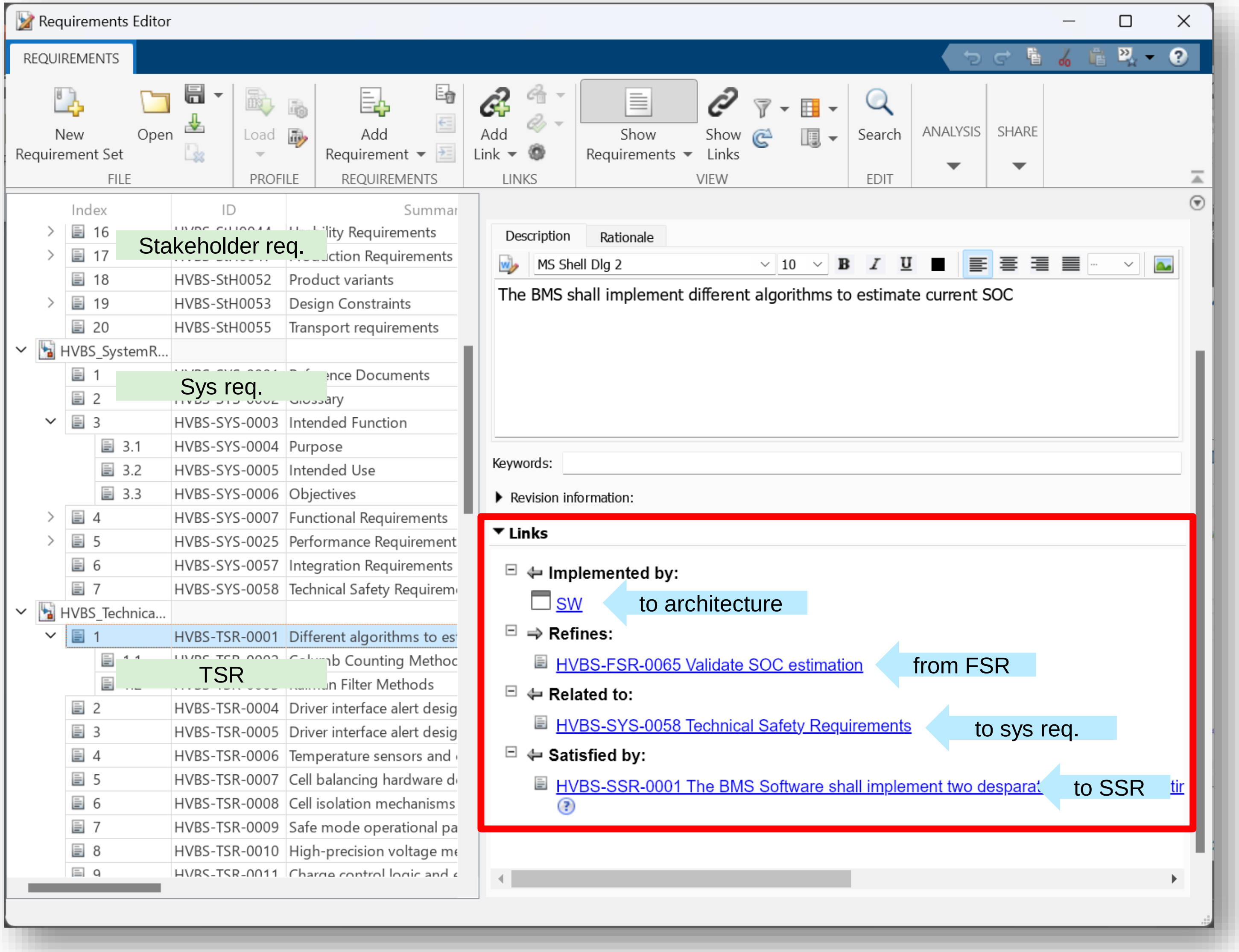


Stakeholder Requirements

System Requirements

Software Requirements

Test Requirements



Index	ID	Summar
16	HVBS-StH0044	Usability Requirements
17	HVBS-StH0047	Production Requirements
18	HVBS-StH0052	Product variants
19	HVBS-StH0053	Design Constraints
20	HVBS-StH0055	Transport requirements
HVBS_SystemR...		
1	HVBS-SYS-0001	Reference Documents
2	HVBS-SYS-0002	Glossary
3	HVBS-SYS-0003	Intended Function
3.1	HVBS-SYS-0004	Purpose
3.2	HVBS-SYS-0005	Intended Use
3.3	HVBS-SYS-0006	Objectives
4	HVBS-SYS-0007	Functional Requirements
5	HVBS-SYS-0025	Performance Requirement
6	HVBS-SYS-0057	Integration Requirements
7	HVBS-SYS-0058	Technical Safety Requireme
HVBS_Technica...		
1	HVBS-TSR-0001	Different algorithms to est
1.1	HVBS-TSR-0002	Columb Counting Methoc
1.2	HVBS-TSR-0003	Kalman Filter Methods
2	HVBS-TSR-0004	Driver interface alert desig
3	HVBS-TSR-0005	Driver interface alert desig
4	HVBS-TSR-0006	Temperature sensors and
5	HVBS-TSR-0007	Cell balancing hardware d
6	HVBS-TSR-0008	Cell isolation mechanisms
7	HVBS-TSR-0009	Safe mode operational pa
8	HVBS-TSR-0010	High-precision voltage me
9	HVBS-TSR-0011	Charge control logic and

Stakeholder req.

Sys req.

TSR

Description | Rationale

MS Shell Dlg 2 | 10

The BMS shall implement different algorithms to estimate current SOC

Keywords:

Revision information:

Links

Implemented by:

SW — to architecture

Refines:

HVBS-FSR-0065 Validate SOC estimation — from FSR

Related to:

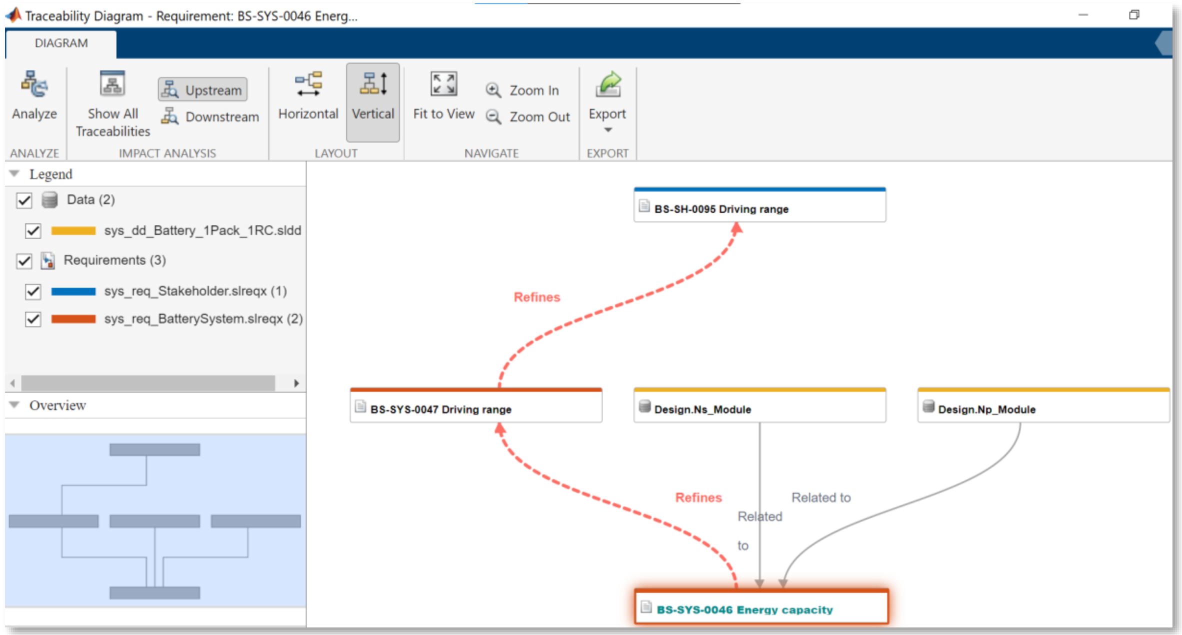
HVBS-SYS-0058 Technical Safety Requirements — to sys req.

Satisfied by:

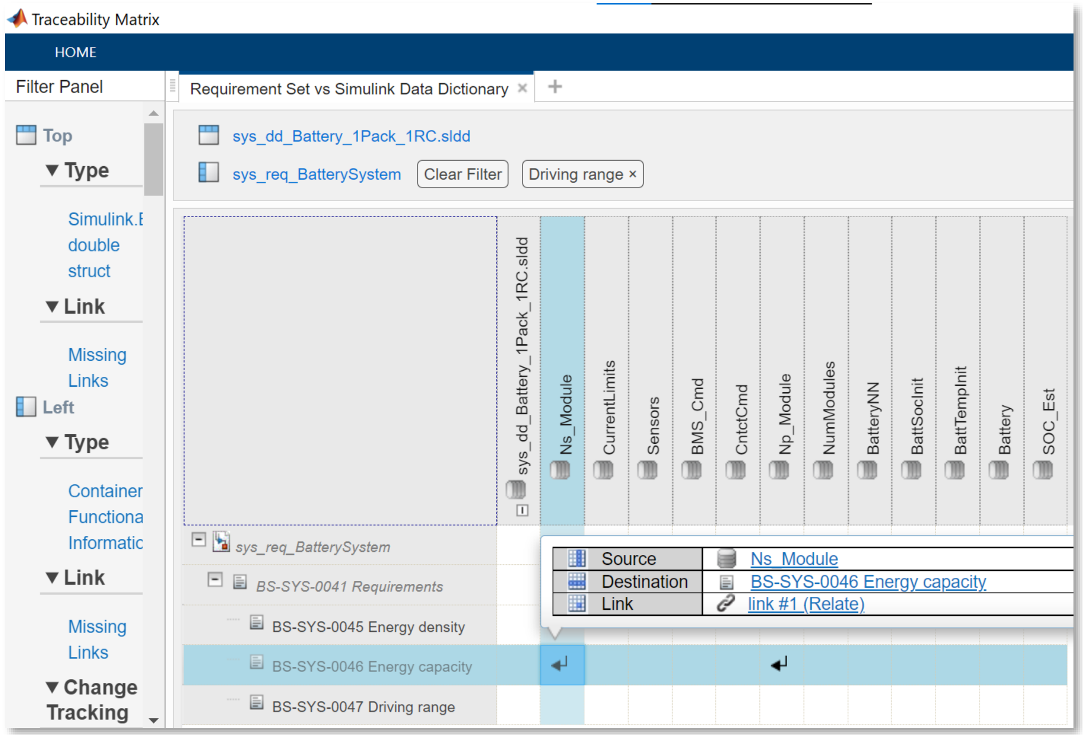
HVBS-SSR-0001 The BMS Software shall implement two desparat — to SSR

Use Traceability Diagrams and Matrixes to Check for Consistency and Completeness

Traceability Diagrams



Traceability Matrix

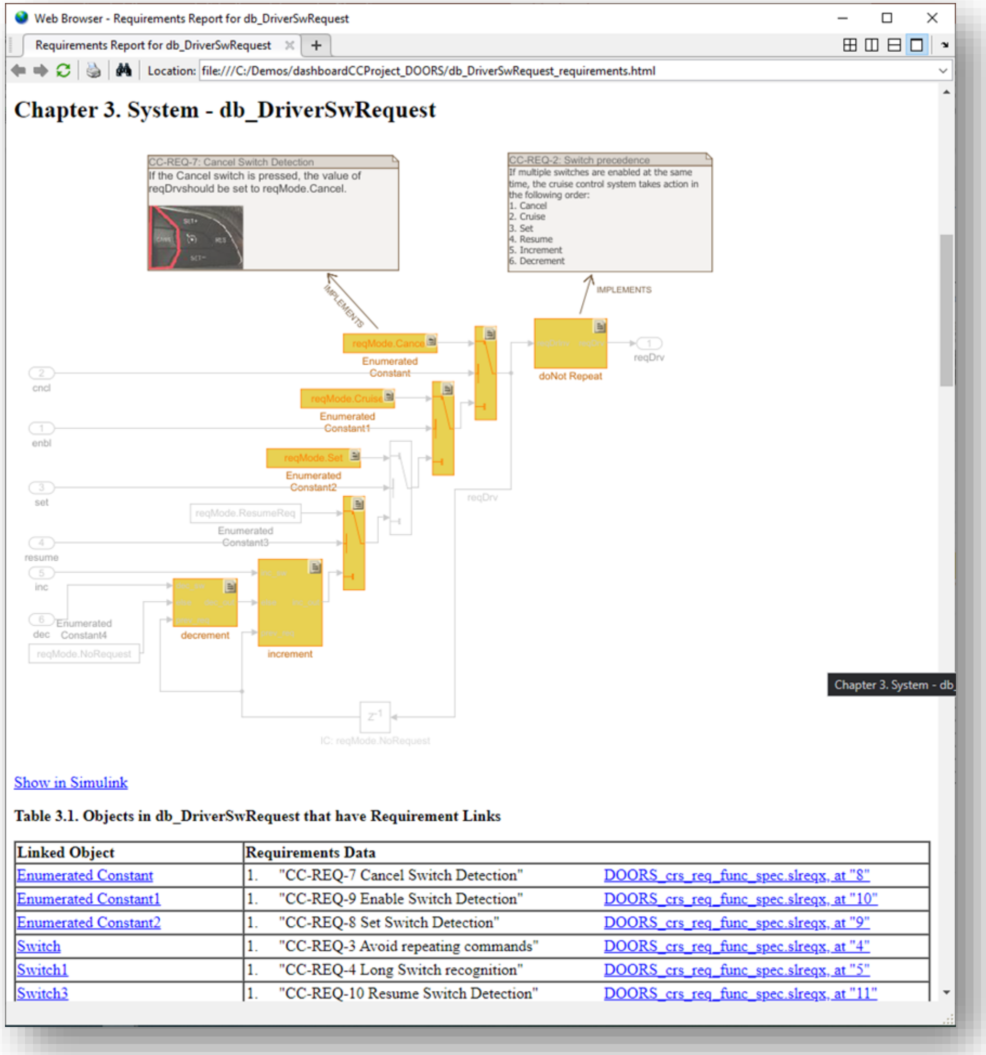


Source	Ns_Module
Destination	BS-SYS-0046 Energy capacity
Link	link #1 (Relate)

Requirements Traceability Report

Simulink Report Generator

- Provides overview of model objects linked with requirements
  - Traceability to high level requirements
  - Required for A-SPICE, CMMI, DO-178B, DO-254, IEC 61508, ISO 26262 etc.
  - Helps find objects with incorrect, incomplete, ambiguous or missing requirements



Chapter 3. System - db_DriverSwRequest

CC-REQ-7: Cancel Switch Detection

If the Cancel switch is pressed, the value of reqDrvshould be set to reqMode.Cancel.

CC-REQ-2: Switch precedence

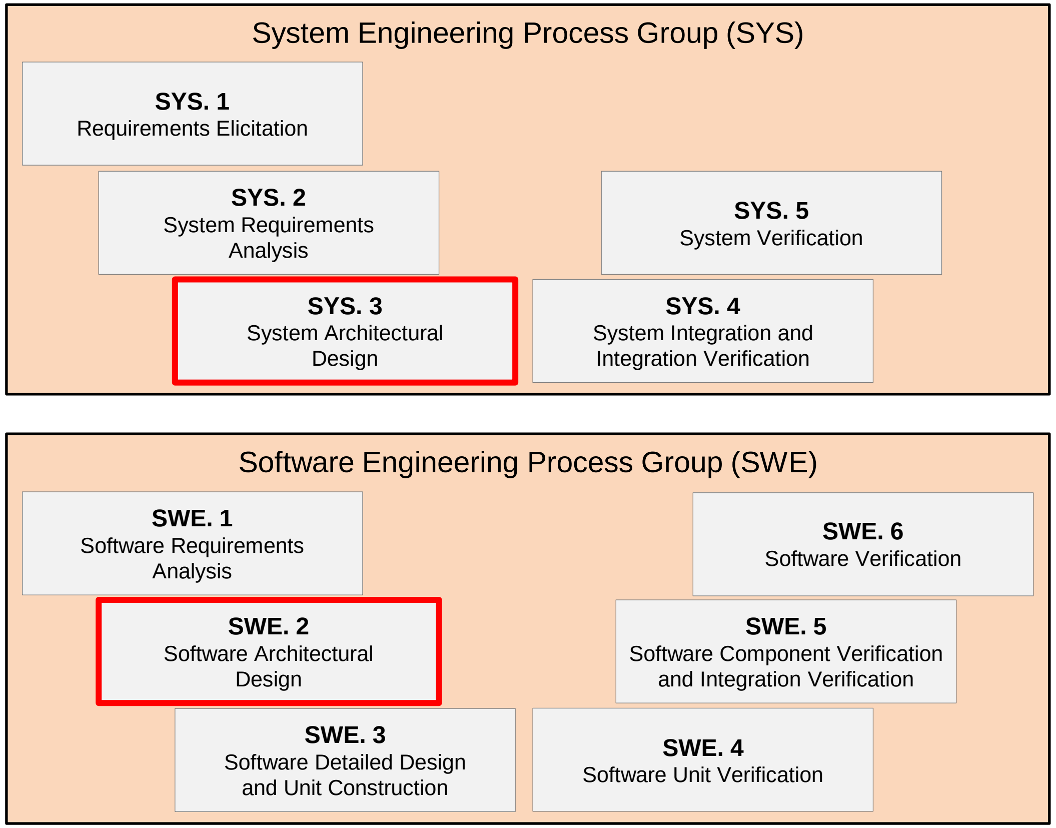
If multiple switches are enabled at the same time, the cruise control system takes action in the following order:
1. Cancel
2. Cruise
3. Set
4. Resume
5. Increment
6. Decrement

Show in Simulink

Table 3.1. Objects in db_DriverSwRequest that have Requirement Links

Linked Object	Requirements Data	
Enumerated Constant	1. "CC-REQ-7 Cancel Switch Detection"	DOORS_crs_req_func_spec.slreqx, at "8"
Enumerated Constant1	1. "CC-REQ-9 Enable Switch Detection"	DOORS_crs_req_func_spec.slreqx, at "10"
Enumerated Constant2	1. "CC-REQ-8 Set Switch Detection"	DOORS_crs_req_func_spec.slreqx, at "9"
Switch	1. "CC-REQ-3 Avoid repeating commands"	DOORS_crs_req_func_spec.slreqx, at "4"
Switch1	1. "CC-REQ-4 Long Switch recognition"	DOORS_crs_req_func_spec.slreqx, at "5"
Switch3	1. "CC-REQ-10 Resume Switch Detection"	DOORS_crs_req_func_spec.slreqx, at "11"

Architecture Design



System Engineering Process Group (SYS)

SYS. 1
Requirements Elicitation

SYS. 2
System Requirements Analysis

SYS. 3
System Architectural Design

SYS. 4
System Integration and Integration Verification

SYS. 5
System Verification

Software Engineering Process Group (SWE)

SWE. 1
Software Requirements Analysis

SWE. 2
Software Architectural Design

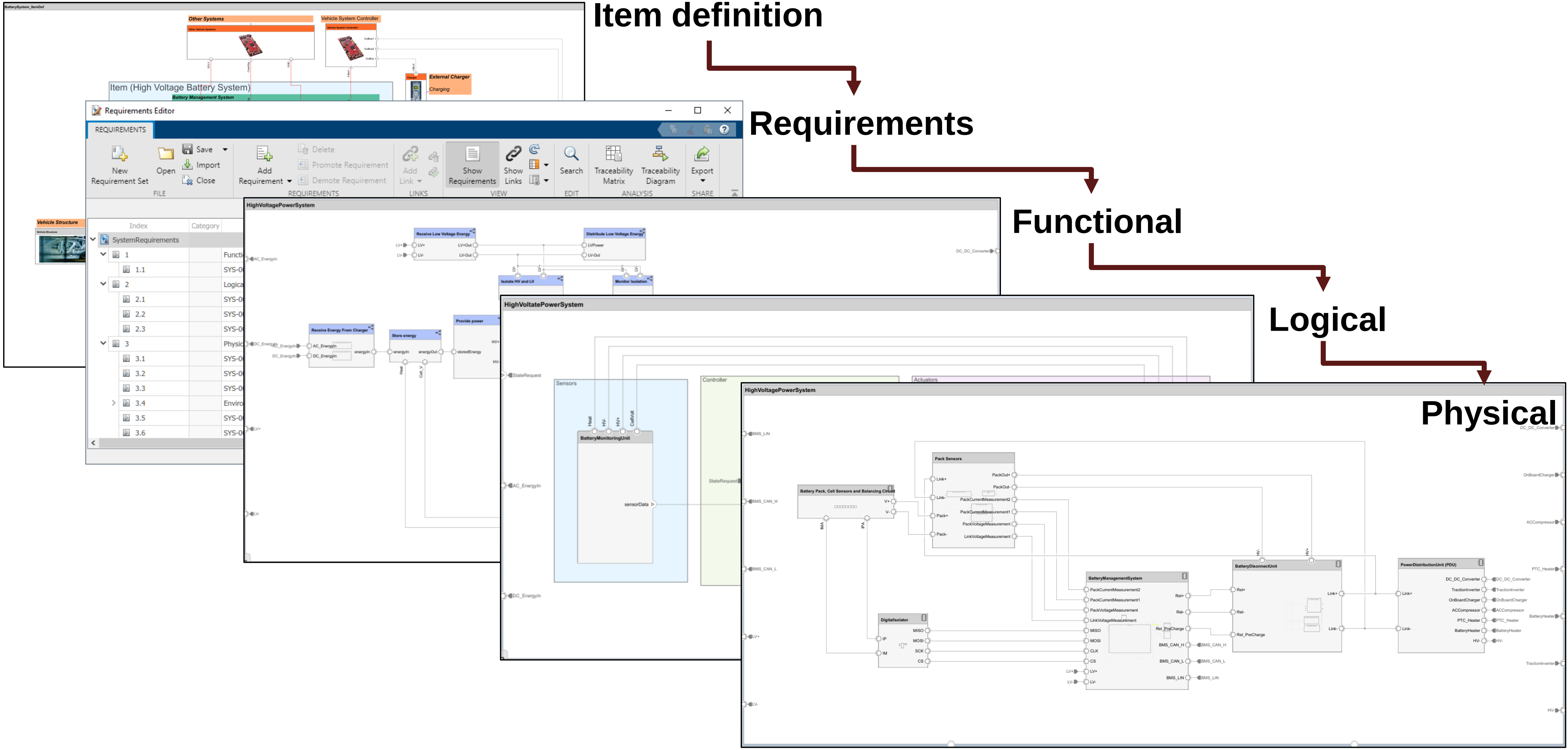
SWE. 3
Software Detailed Design and Unit Construction

SWE. 4
Software Unit Verification

SWE. 5
Software Component Verification and Integration Verification

SWE. 6
Software Verification

Develop Architectural Design Models with System Composer



Index	Category
SystemRequirements	
1	Functi
1.1	SYS-0
2	Logica
2.1	SYS-0
2.2	SYS-0
2.3	SYS-0
3	Physic
3.1	SYS-0
3.2	SYS-0
3.3	SYS-0
3.4	Enviro
3.5	SYS-0
3.6	SYS-0

Item definition

Requirements

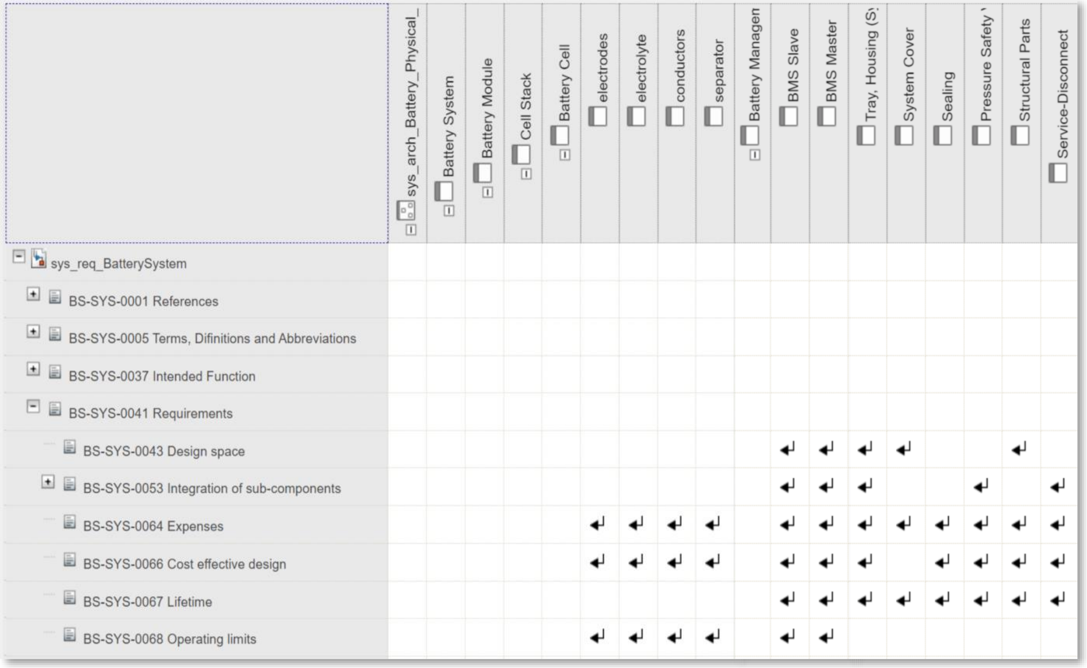
Functional

Logical

Physical

Ensure Consistency with Tool Support for Bidirectional Traceability

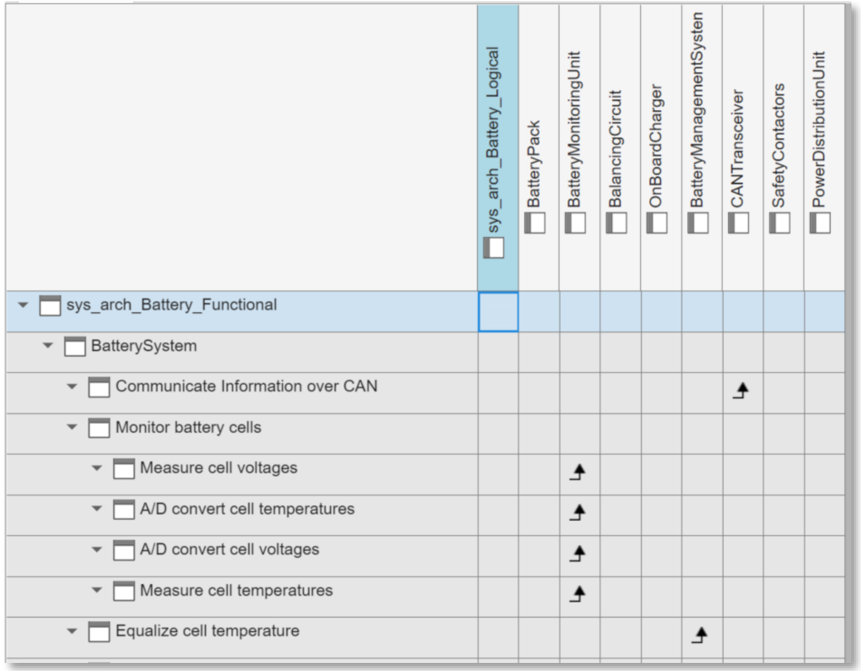
Requirements ↔ Architecture



	sys_arch_Battery_Physical_	Battery System	Battery Module	Cell Stack	Battery Cell	electrodes	electrolyte	conductors	separator	Battery Managem	BMS Slave	BMS Master	Tray, Housing (S	System Cover	Sealing	Pressure Safety '	Structural Parts	Service-Disconnect
sys_req_BatterySystem																		
BS-SYS-0001 References																		
BS-SYS-0005 Terms, Difinitions and Abbreviations																		
BS-SYS-0037 Intended Function																		
BS-SYS-0041 Requirements																		
BS-SYS-0043 Design space											↲	↲	↲	↲			↲	
BS-SYS-0053 Integration of sub-components											↲	↲	↲			↲		↲
BS-SYS-0064 Expenses						↲	↲	↲	↲		↲	↲	↲	↲	↲	↲	↲	↲
BS-SYS-0066 Cost effective design						↲	↲	↲	↲		↲	↲	↲		↲	↲	↲	↲
BS-SYS-0067 Lifetime											↲	↲	↲	↲	↲	↲	↲	↲
BS-SYS-0068 Operating limits						↲	↲	↲	↲		↲	↲						

Requirements Editor

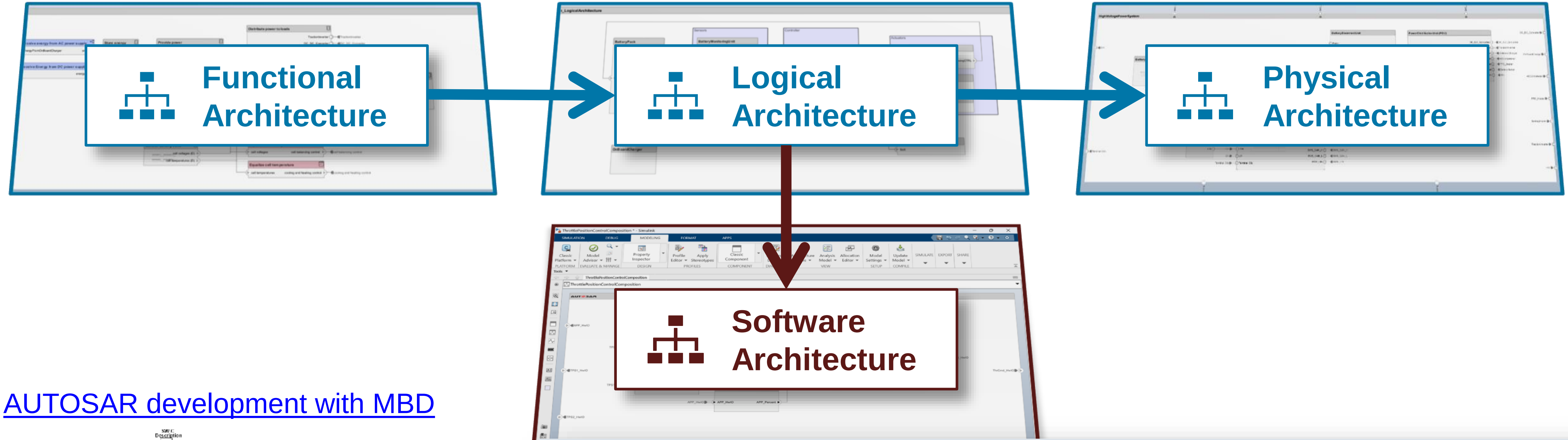
Architecture ↔ Architecture

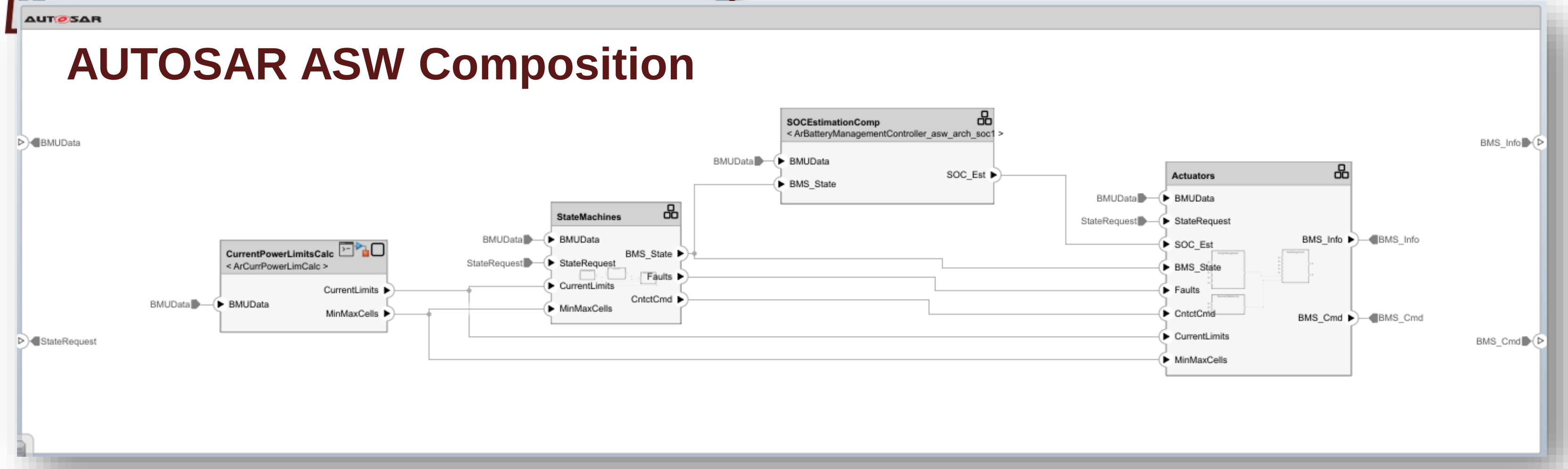


	sys_arch_Battery_Logical	BatteryPack	BatteryMonitoringUnit	BalancingCircuit	OnBoardCharger	BatteryManagementSysten	CANTransceiver	SafetyContactors	PowerDistributionUnit
sys_arch_Battery_Functional									
BatterySystem									
Communicate Information over CAN							↑		
Monitor battery cells									
Measure cell voltages			↑						
A/D convert cell temperatures			↑						
A/D convert cell voltages			↑						
Measure cell temperatures			↑						
Equalize cell temperature						↑			

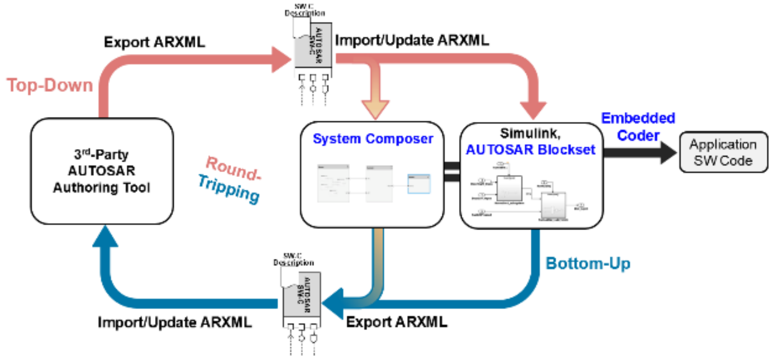
Allocation Editor

Software Architectural Design Models with System Composer

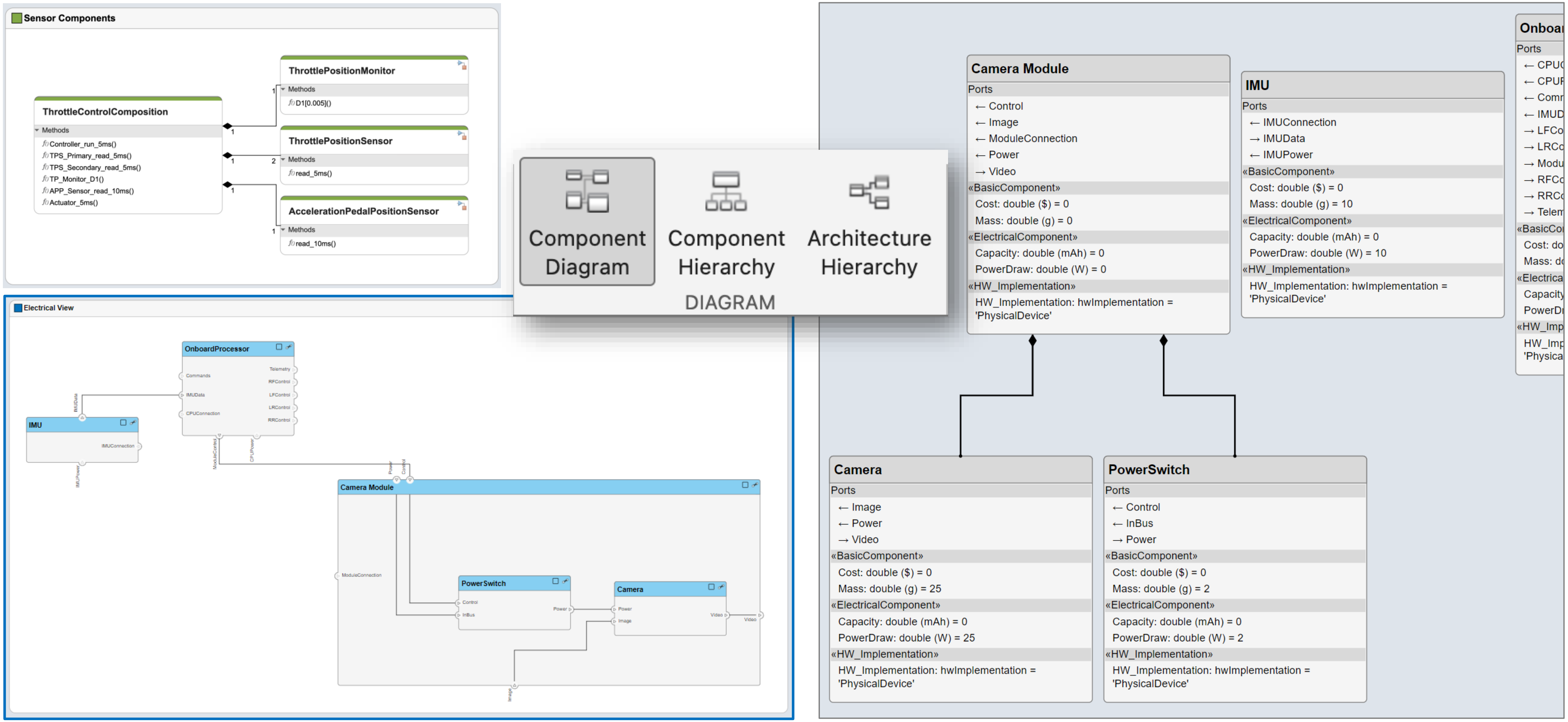




- [AUTOSAR development with MBD]()

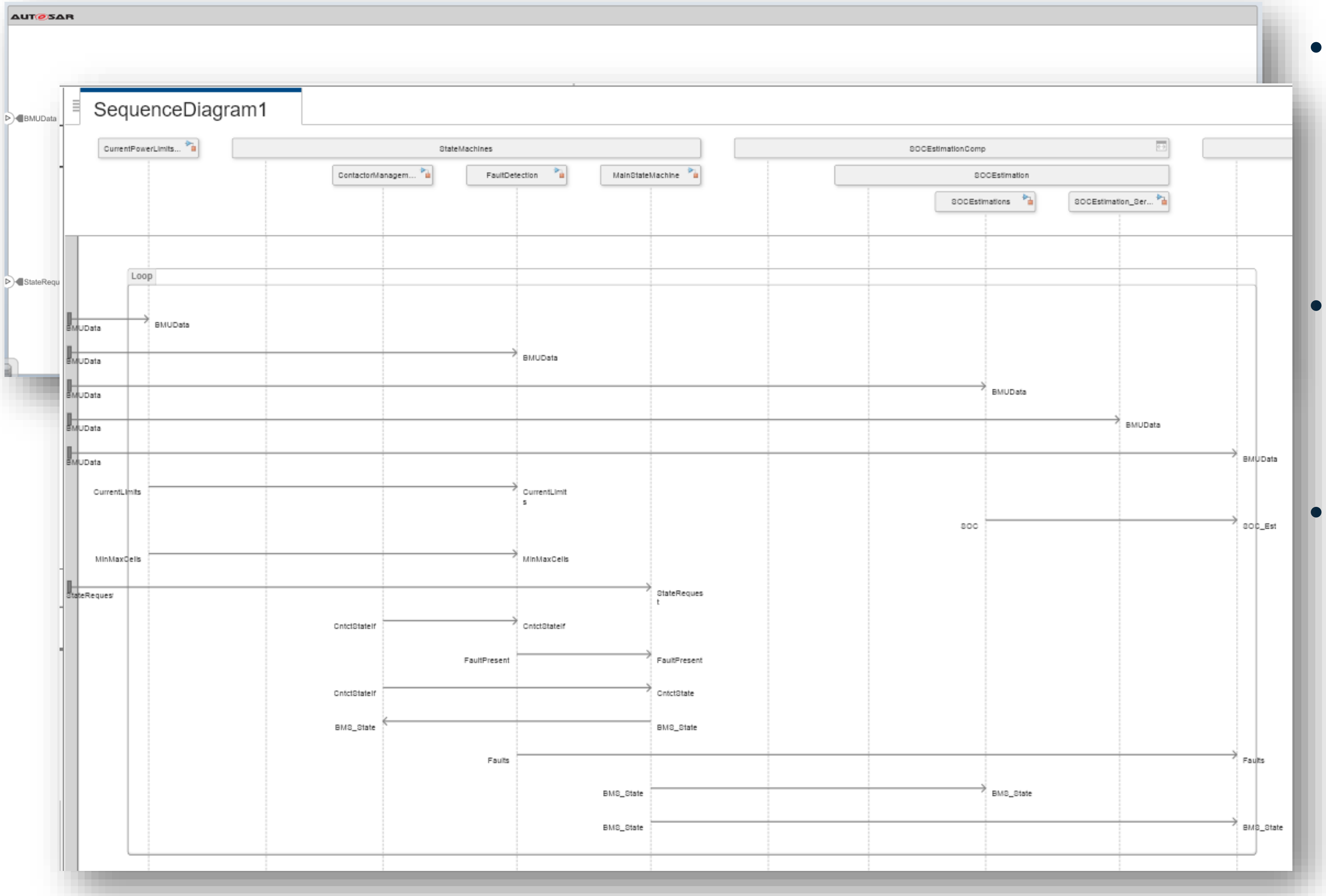


Analyze System Architecture with Autogenerated Custom Views



Describe Dynamic Behavior using **Sequence Diagram**

- Describe system behavior as interactions between components through message exchanges

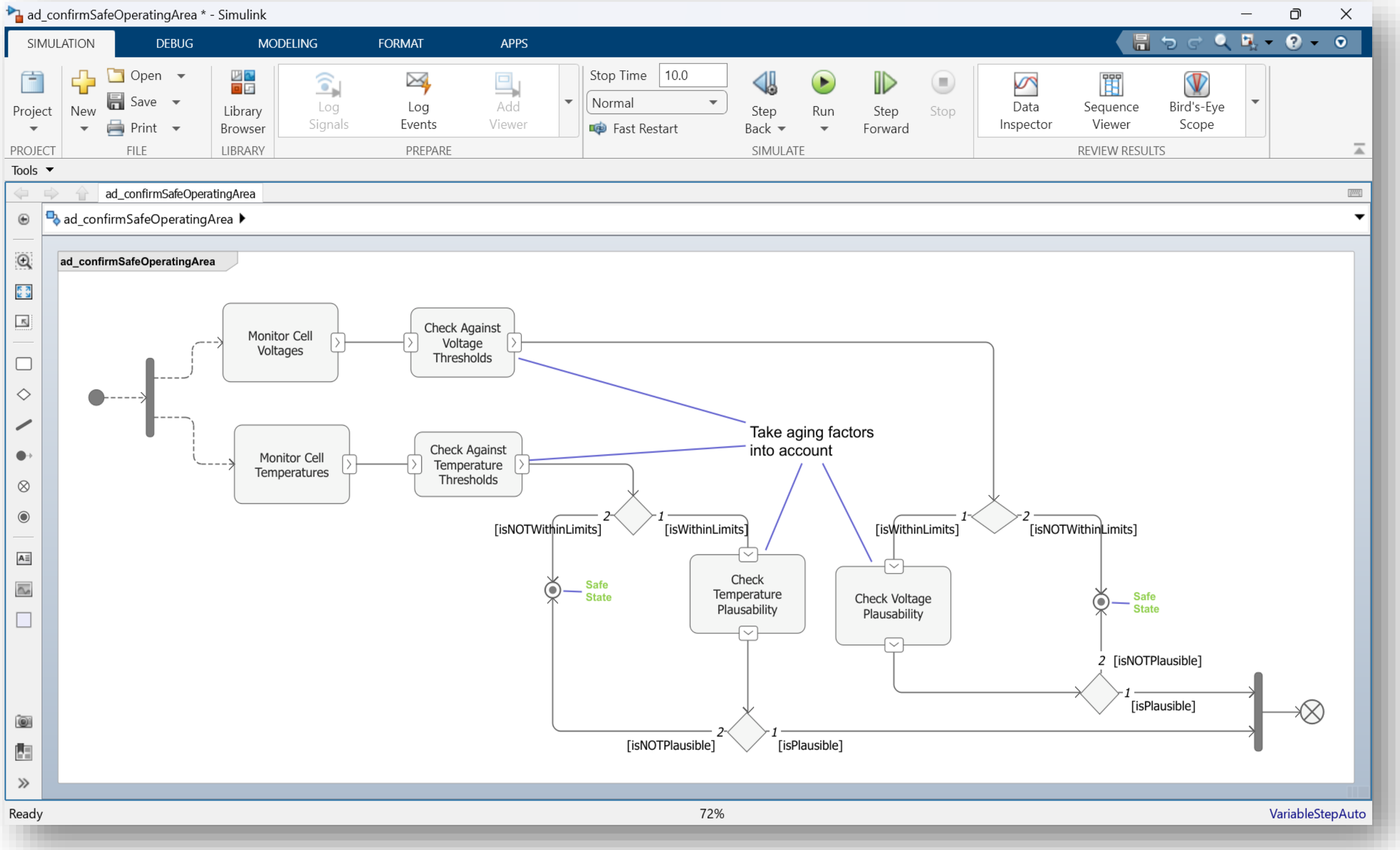


- Create lifelines to represent components, add messages between lifelines and use message labels to describe interactions.
- Describe client-server interactions, and gates connecting to root architecture ports.
- Simulate, and validate sequence diagrams to verify the system design.

https://www.mathworks.com/help/systemcomposer/ug/simulate-sequence-diagrams.html

Describe System Behavior using **Activity Diagrams**

- Validate (via simulation) system behaviors defined as a controlled flow of actions



Flexibly model processes with

- Serial, parallel, iterative actions
- Dynamic decisions
- Hierarchies (or sub-processes)
- Custom logic (via MATLAB functions)

Token (objects being processed)

- Support all Simulink data types

Support simulation features

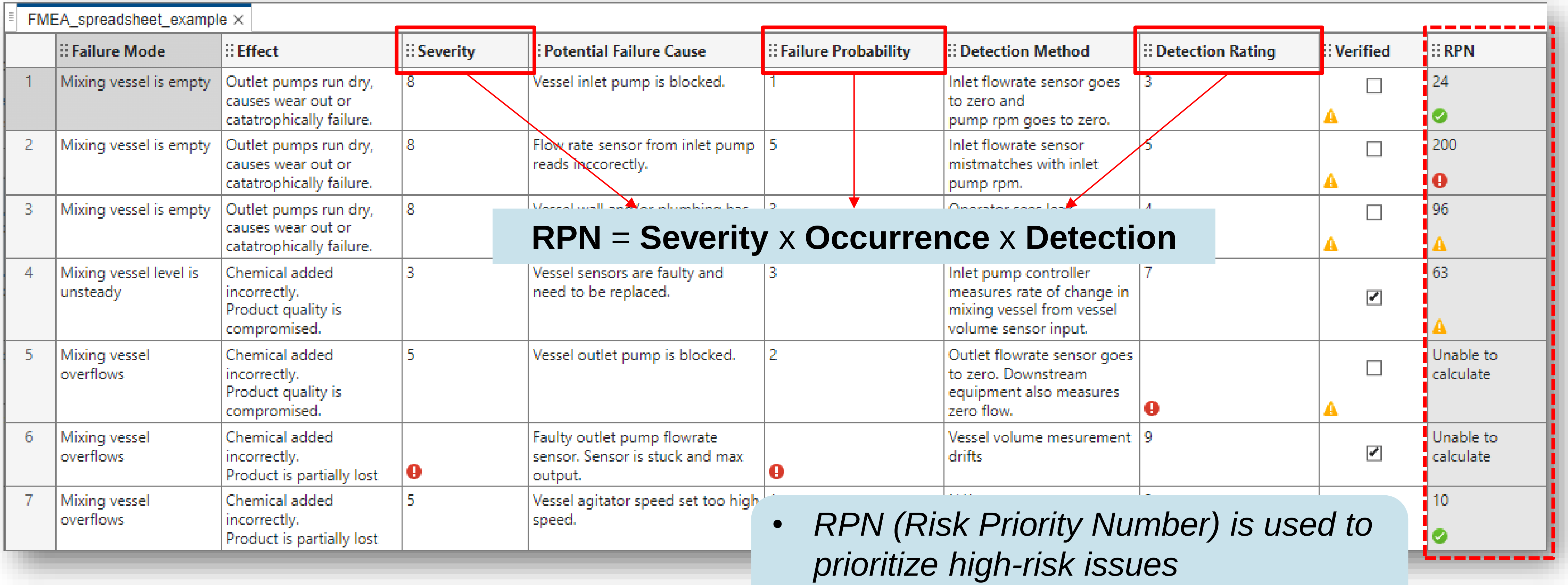
- SDI
- Event animation
- Debugger (value label, breakpoint, step back etc.)

https://www.mathworks.com/help/systemcomposer/ug/author-activity-diagram-for-mobile-robot-example.html

System / SW Failure Mode and Effects Analysis (FMEA)

Simulink Fault Analyzer

- FMEA is to support hazard identification and prevention for the ASIL level

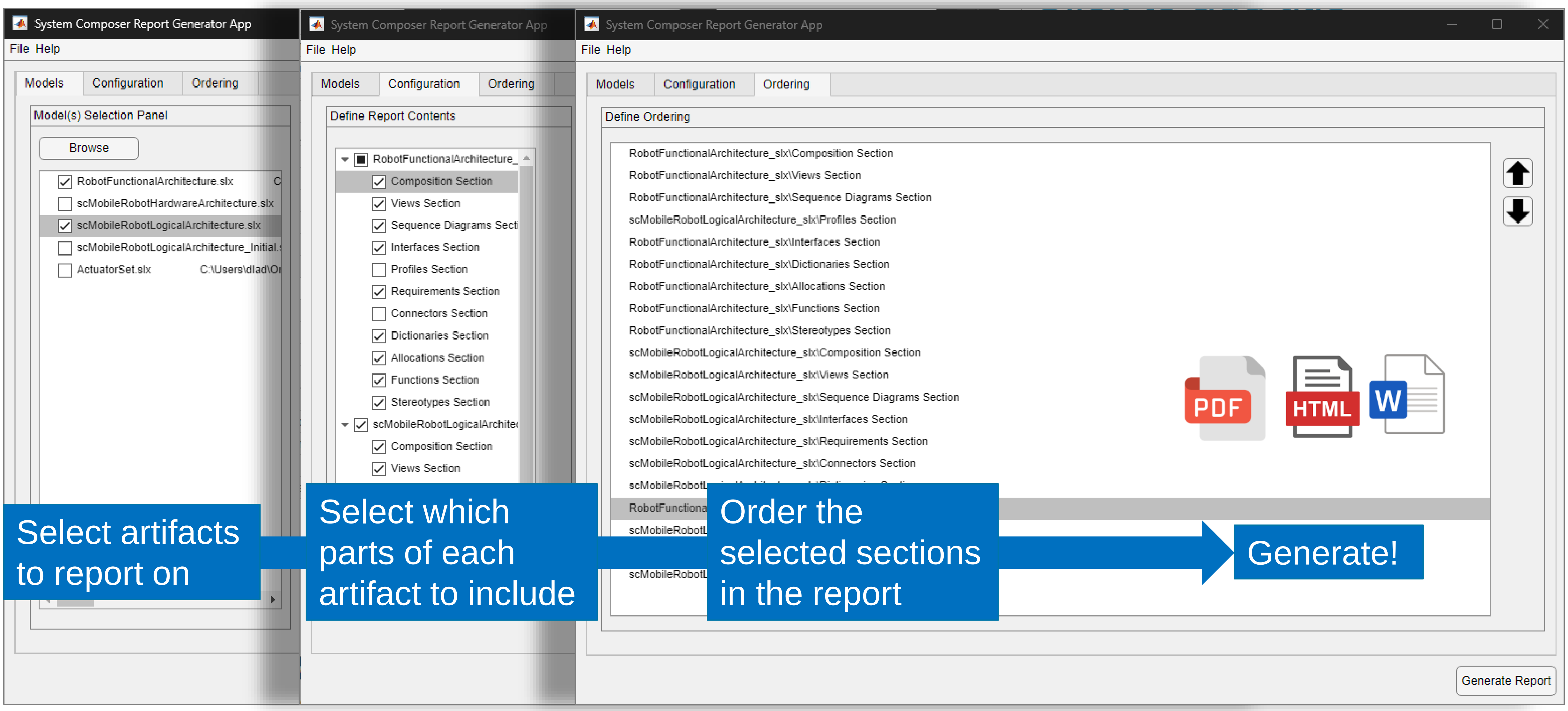


	Failure Mode	Effect	Severity	Potential Failure Cause	Failure Probability	Detection Method	Detection Rating	Verified	RPN
1	Mixing vessel is empty	Outlet pumps run dry, causes wear out or catatrophically failure.	8	Vessel inlet pump is blocked.	1	Inlet flowrate sensor goes to zero and pump rpm goes to zero.	3	☐ ⚠	24 ✓
2	Mixing vessel is empty	Outlet pumps run dry, causes wear out or catatrophically failure.	8	Flow rate sensor from inlet pump reads inccorectly.	5	Inlet flowrate sensor mistmatches with inlet pump rpm.	5	☐ ⚠	200 ❗
3	Mixing vessel is empty	Outlet pumps run dry, causes wear out or catatrophically failure.	8	Vessel wall and/or plumbing has	3	Operator sees leak	4	☐ ⚠	96 ⚠
4	Mixing vessel level is unsteady	Chemical added incorrectly. Product quality is compromised.	3	Vessel sensors are faulty and need to be replaced.	3	Inlet pump controller measures rate of change in mixing vessel from vessel volume sensor input.	7	☑	63 ⚠
5	Mixing vessel overflows	Chemical added incorrectly. Product quality is compromised.	5	Vessel outlet pump is blocked.	2	Outlet flowrate sensor goes to zero. Downstream equipment also measures zero flow.	❗	☐ ⚠	Unable to calculate
6	Mixing vessel overflows	Chemical added incorrectly. Product is partially lost	❗	Faulty outlet pump flowrate sensor. Sensor is stuck and max output.	❗	Vessel volume mesurement drifts	9	☑	Unable to calculate
7	Mixing vessel overflows	Chemical added incorrectly. Product is partially lost	5	Vessel agitator speed set too high speed.	1	N/A	2		10 ✓

RPN = Severity x Occurrence x Detection

- *RPN (Risk Priority Number) is used to prioritize high-risk issues*
- *RPN threshold determines which failure mode requires corrective action*

The “System Composer Report Generator App” offers fast automated reports with basic customization



Model(s) Selection Panel

Browse

☑ RobotFunctionalArchitecture.slx
□ scMobileRobotHardwareArchitecture.slx
☑ scMobileRobotLogicalArchitecture.slx
□ scMobileRobotLogicalArchitecture_Initial.s
□ ActuatorSet.slx C:\Users\dlad\Or

Define Report Contents

■ RobotFunctionalArchitecture_
- ☑ Composition Section
- ☑ Views Section
- ☑ Sequence Diagrams Secti
- ☑ Interfaces Section
- □ Profiles Section
- ☑ Requirements Section
- □ Connectors Section
- ☑ Dictionaries Section
- ☑ Allocations Section
- ☑ Functions Section
- ☑ Stereotypes Section

☑ scMobileRobotLogicalArchitec
- ☑ Composition Section
- ☑ Views Section

Define Ordering

RobotFunctionalArchitecture_slx\Composition Section
RobotFunctionalArchitecture_slx\Views Section
RobotFunctionalArchitecture_slx\Sequence Diagrams Section
scMobileRobotLogicalArchitecture_slx\Profiles Section
RobotFunctionalArchitecture_slx\Interfaces Section
RobotFunctionalArchitecture_slx\Dictionaries Section
RobotFunctionalArchitecture_slx\Allocations Section
RobotFunctionalArchitecture_slx\Functions Section
RobotFunctionalArchitecture_slx\Stereotypes Section
scMobileRobotLogicalArchitecture_slx\Composition Section
scMobileRobotLogicalArchitecture_slx\Views Section
scMobileRobotLogicalArchitecture_slx\Sequence Diagrams Section
scMobileRobotLogicalArchitecture_slx\Interfaces Section
scMobileRobotLogicalArchitecture_slx\Requirements Section
scMobileRobotLogicalArchitecture_slx\Connectors Section
scMobileRobotL
RobotFunctiona
scMobileRobotL
scMobileRobotL

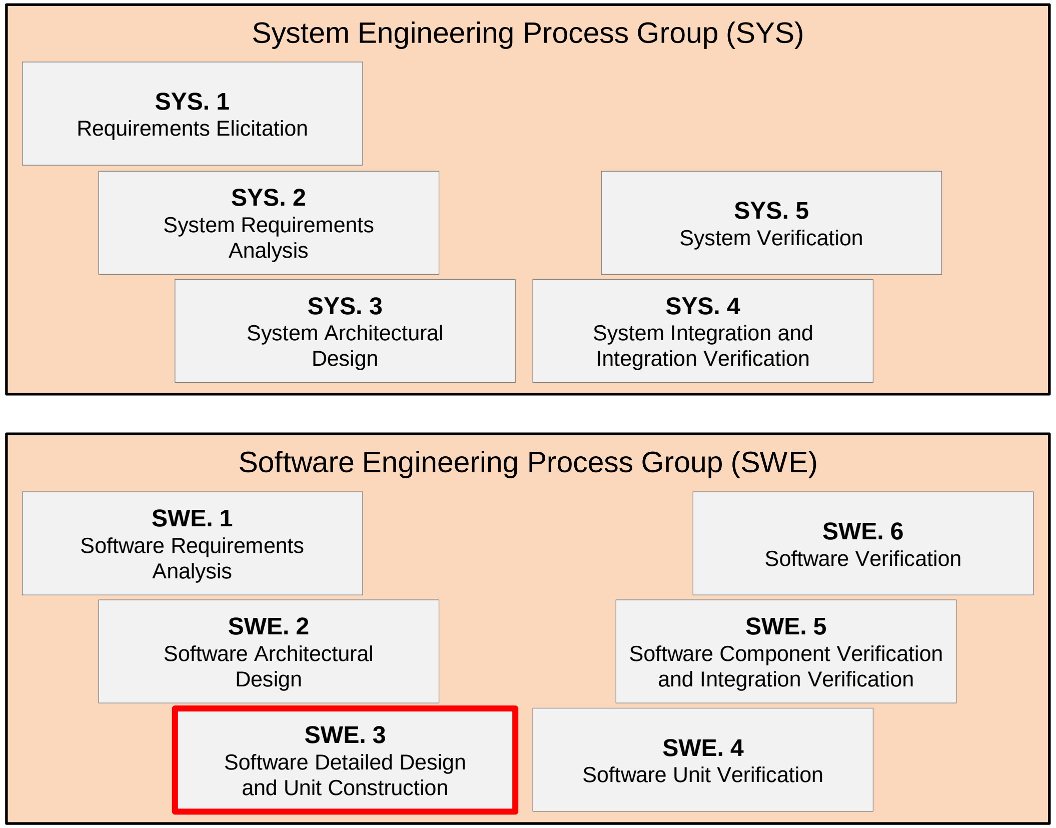
Select artifacts to report on

Select which parts of each artifact to include

Order the selected sections in the report

Generate!

Software Detailed Design and Unit Construction



System Engineering Process Group (SYS)

SYS. 1
Requirements Elicitation

SYS. 2
System Requirements Analysis

SYS. 3
System Architectural Design

SYS. 4
System Integration and Integration Verification

SYS. 5
System Verification

Software Engineering Process Group (SWE)

SWE. 1
Software Requirements Analysis

SWE. 2
Software Architectural Design

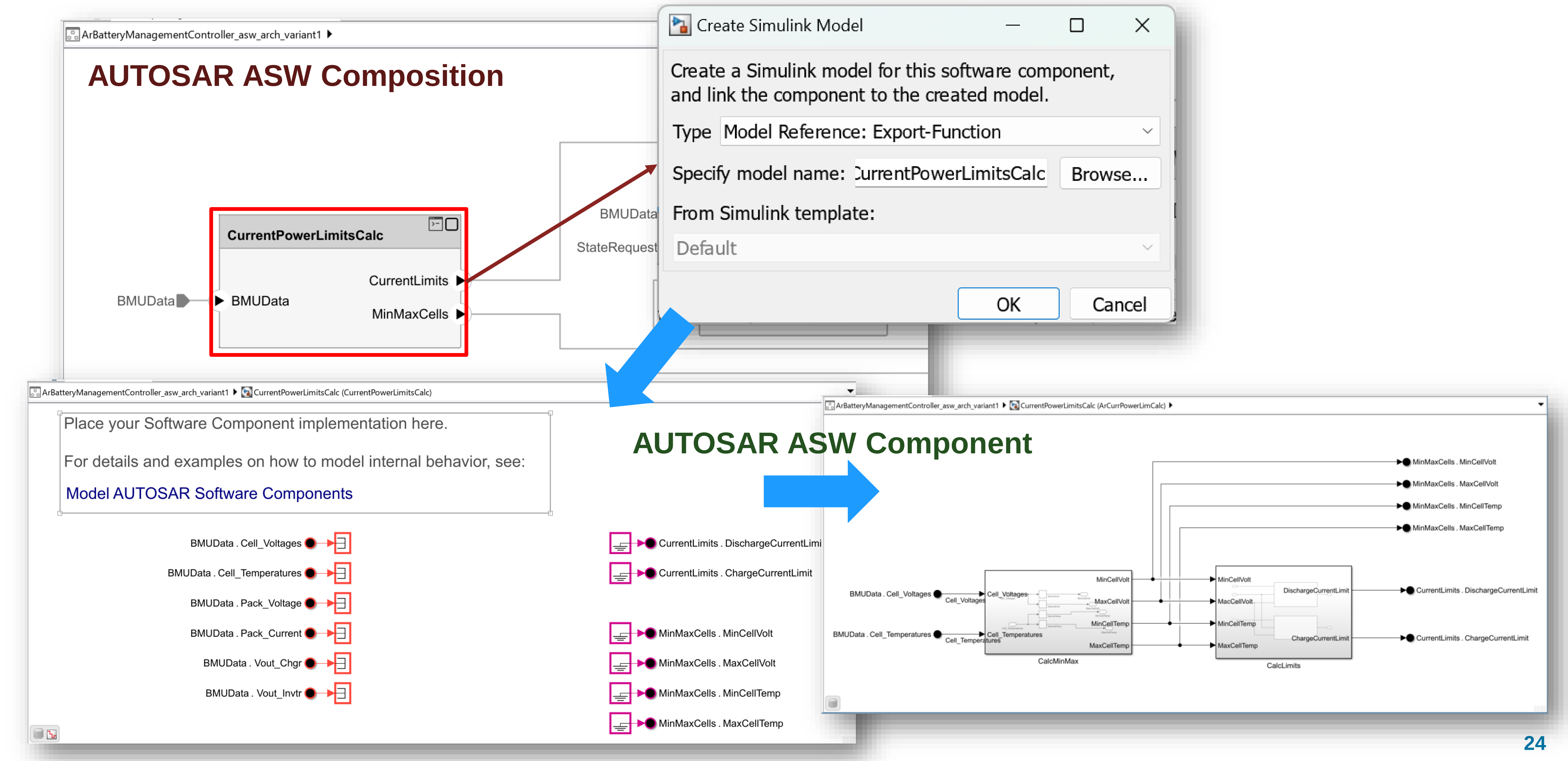
SWE. 3
Software Detailed Design and Unit Construction

SWE. 4
Software Unit Verification

SWE. 5
Software Component Verification and Integration Verification

SWE. 6
Software Verification

Software Detailed Design Seamlessly from Software Architecture



AUTOSAR ASW Composition

AUTOSAR ASW Component

Create Simulink Model

Create a Simulink model for this software component, and link the component to the created model.

Type: Model Reference: Export-Function

Specify model name: CurrentPowerLimitsCalc

From Simulink template: Default

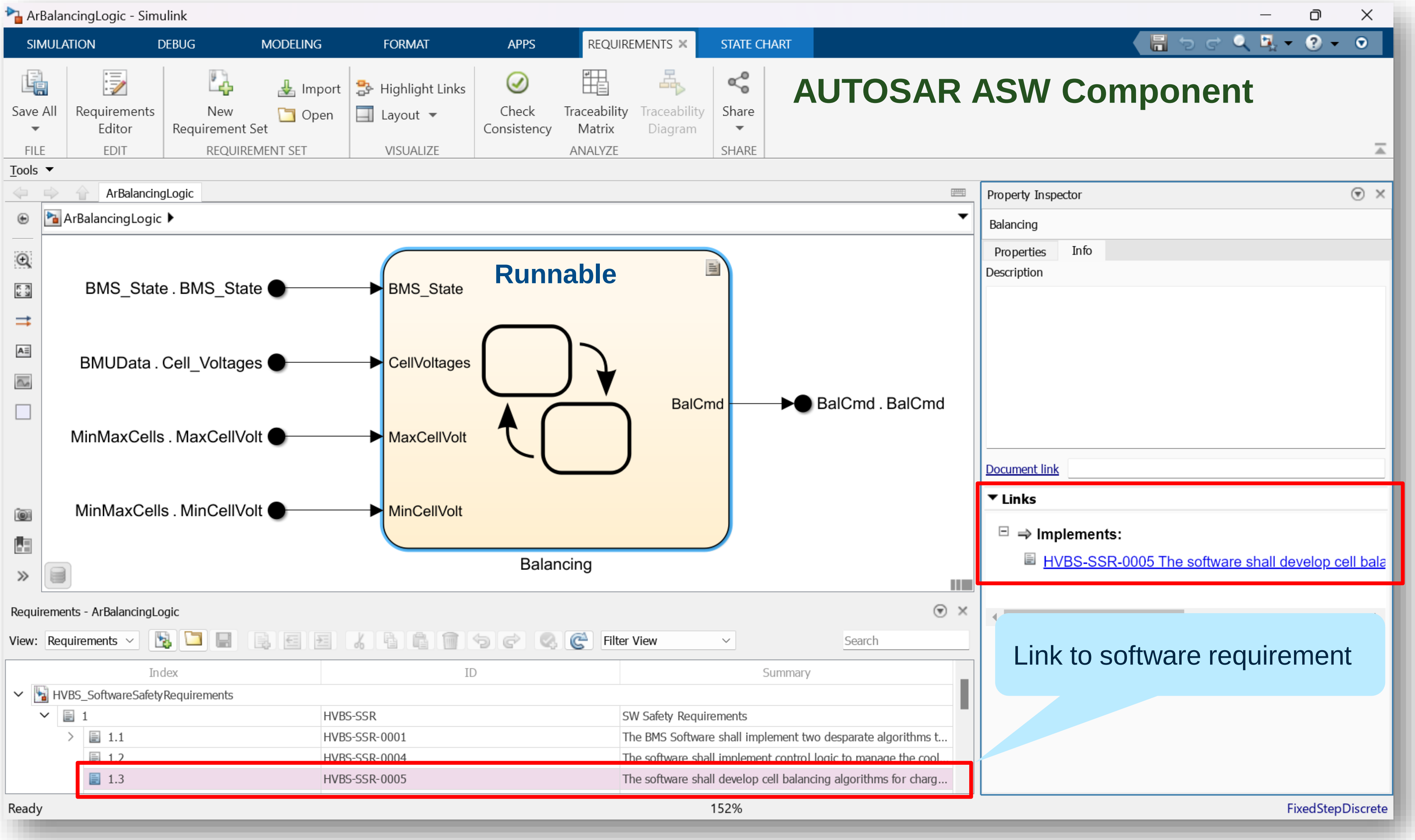
Place your Software Component implementation here.

For details and examples on how to model internal behavior, see:

Model AUTOSAR Software Components

ArBatteryManagementController_asw_arch_variant1
CurrentPowerLimitsCalc
BMUData
CurrentLimits
MinMaxCells
StateRequest
Browse...
OK
Cancel
ArBatteryManagementController_asw_arch_variant1 ▸ CurrentPowerLimitsCalc (CurrentPowerLimitsCalc)
BMUData . Cell_Voltages
BMUData . Cell_Temperatures
BMUData . Pack_Voltage
BMUData . Pack_Current
BMUData . Vout_Chgr
BMUData . Vout_Invtr
CurrentLimits . DischargeCurrentLimi
CurrentLimits . ChargeCurrentLimit
MinMaxCells . MinCellVolt
MinMaxCells . MaxCellVolt
MinMaxCells . MinCellTemp
MinMaxCells . MaxCellTemp
ArBatteryManagementController_asw_arch_variant1 ▸ CurrentPowerLimitsCalc (ArCurrPowerLimCalc) ▸
Cell_Voltages
Cell_Temperatures
MinCellVolt
MaxCellVolt
MinCellTemp
MaxCellTemp
CalcMinMax
MacCellVolt
DischargeCurrentLimit
ChargeCurrentLimit
CalcLimits
CurrentLimits . DischargeCurrentLimit
CurrentLimits . ChargeCurrentLimit

Software Detailed Design



AUTOSAR ASW Component

Runnable

Property Inspector: Balancing

Description

Links

Implements:

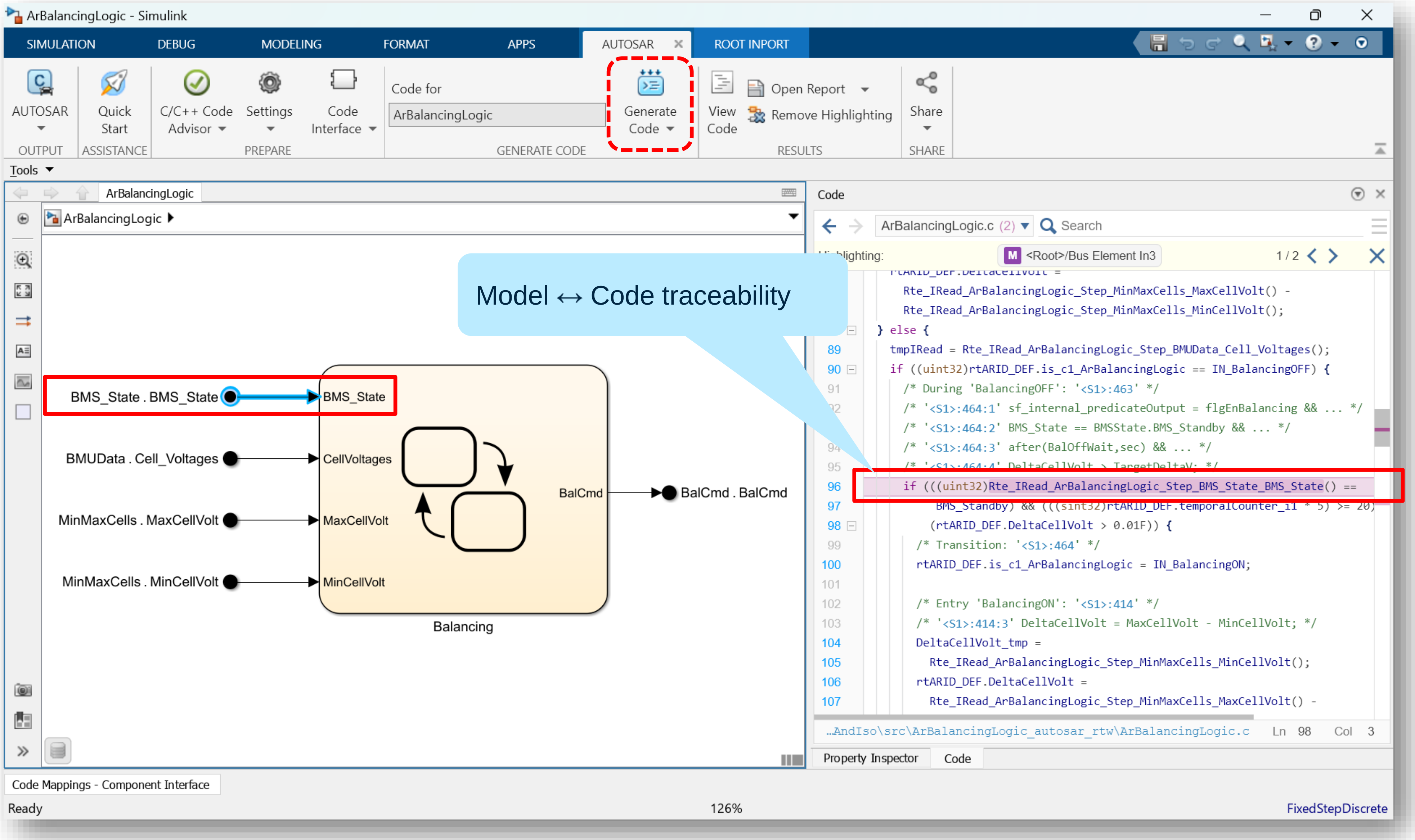
HVBS-SSR-0005 The software shall develop cell bala

Link to software requirement

View: Requirements

Index	ID	Summary
HVBS_SoftwareSafetyRequirements		
1	HVBS-SSR	SW Safety Requirements
1.1	HVBS-SSR-0001	The BMS Software shall implement two desparate algorithms t...
1.2	HVBS-SSR-0004	The software shall implement control logic to manage the cool
1.3	HVBS-SSR-0005	The software shall develop cell balancing algorithms for charg...

Code Generation Software Detailed Design



Model ↔ Code traceability

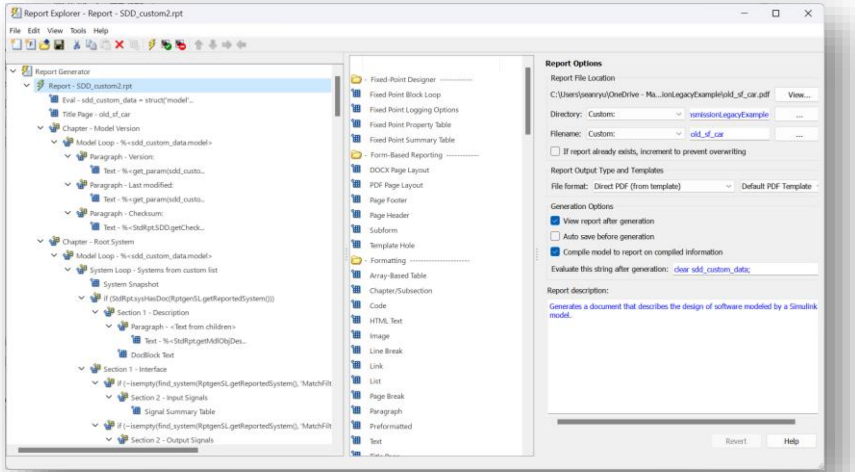
```
    rtARID_DEF.DeltaCellVolt =
      Rte_IRead_ArBalancingLogic_Step_MinMaxCells_MaxCellVolt() -
      Rte_IRead_ArBalancingLogic_Step_MinMaxCells_MinCellVolt();
  } else {
    tmpIRead = Rte_IRead_ArBalancingLogic_Step_BMUData_Cell_Voltages();
    if ((uint32)rtARID_DEF.is_c1_ArBalancingLogic == IN_BalancingOFF) {
      /* During 'BalancingOFF': '<S1>:463' */
      /* '<S1>:464:1' sf_internal_predicateOutput = flgEnBalancing && ... */
      /* '<S1>:464:2' BMS_State == BMSState.BMS_Standby && ... */
      /* '<S1>:464:3' after(BalOffWait,sec) && ... */
      /* '<S1>:464:4' DeltaCellVolt > TargetDeltaV; */
      if (((uint32)Rte_IRead_ArBalancingLogic_Step_BMS_State_BMS_State() ==
            BMS_Standby) && (((sint32)rtARID_DEF.temporalCounter_i1 * 5) >= 20)
          (rtARID_DEF.DeltaCellVolt > 0.01F)) {
        /* Transition: '<S1>:464' */
        rtARID_DEF.is_c1_ArBalancingLogic = IN_BalancingON;

        /* Entry 'BalancingON': '<S1>:414' */
        /* '<S1>:414:3' DeltaCellVolt = MaxCellVolt - MinCellVolt; */
        DeltaCellVolt_tmp =
          Rte_IRead_ArBalancingLogic_Step_MinMaxCells_MinCellVolt();
        rtARID_DEF.DeltaCellVolt =
          Rte_IRead_ArBalancingLogic_Step_MinMaxCells_MaxCellVolt() -
```

..AndIso\src\ArBalancingLogic_autosar_rtw\ArBalancingLogic.c Ln 98 Col 3

Detailed Design Description form Software Unit

Simulink Report Generator

- Report Explorer



- Report Generator
  - Report - SDD_custom2.rpt
    - Eval - sdd_custom_data = struct('model'...
    - Title Page - old_sf_car
    - Chapter - Model Version
      - Model Loop - %<sdd_custom_data.model>
        - Paragraph - Version:
          - Text - %<get_param(sdd_custo...
        - Paragraph - Last modified:
          - Text - %<get_param(sdd_custo...
        - Paragraph - Checksum:
          - Text - %<StdRpt.SDD.getCheck...
    - Chapter - Root System
      - Model Loop - %<sdd_custom_data.model>
        - System Loop - Systems from custom list
          - System Snapshot
          - if (StdRpt.sysHasDoc(RptgenSL.getReportedSystem()))
            - Section 1 - Description
              - Paragraph - <Text from children>
                - Text - %<StdRpt.getMdlObjDes...
              - DocBlock Text
          - Section 1 - Interface
            - if (~isempty(find_system(RptgenSL.getReportedSystem(), 'MatchFilt
              - Section 2 - Input Signals
                - Signal Summary Table
            - if (~isempty(find_system(RptgenSL.getReportedSystem(), 'MatchFilt
              - Section 2 - Output Signals

- Fixed-Point Designer
  - Fixed Point Block Loop
  - Fixed Point Logging Options
  - Fixed Point Property Table
  - Fixed Point Summary Table
- Form-Based Reporting
  - DOCX Page Layout
  - PDF Page Layout
  - Page Footer
  - Page Header
  - Subform
  - Template Hole
- Formatting
  - Array-Based Table
  - Chapter/Subsection
  - Code
  - HTML Text
  - Image
  - Line Break
  - Link
  - List
  - Page Break
  - Paragraph
  - Preformatted
  - Text

Report Options

Report File Location

C:\Users\seanryu\OneDrive - Ma...ionLegacyExample\old_sf_car.pdf [View...]

Directory: Custom: ısmissionLegacyExample [...]

Filename: Custom: old_sf_car [...]

□ If report already exists, increment to prevent overwriting

Report Output Type and Templates

File format: Direct PDF (from template) | Default PDF Template

Generation Options

☑ View report after generation

□ Auto save before generation

☑ Compile model to report on compiled information

Evaluate this string after generation: clear sdd_custom_data;

Report description:

Generates a document that describes the design of software modeled by a Simulink model.

OR

- Report APIs

- Import API functions

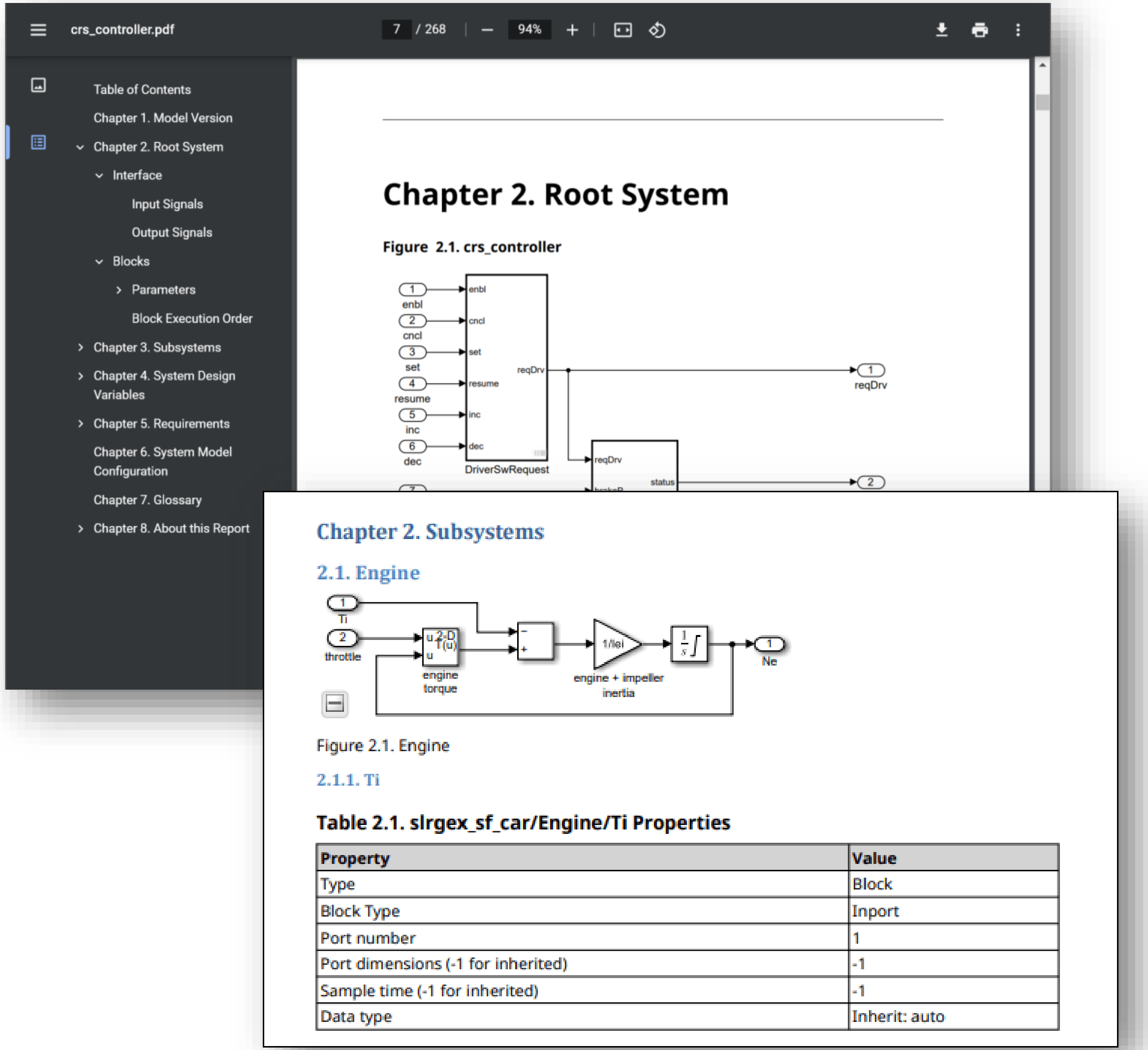
```
import slreportgen.report.*
import slreportgen.finder.*
import mlreportgen.report.*
```

- Add a title page

```
tp = TitlePage;
tp.Title = upper(get_param(model,'Name'));
tp.Subtitle = 'System Design Description';
tp.Author = 'MathWorks';
tp.Image = Diagram(model); append(rpt,tp);
```

- Add a chapter for subsystems

```
ch = Chapter("Title","Subsystems");
sysdiagFinder = SystemDiagramFinder(model);
sysdiagFinder.IncludeRoot = false;  append(rpt,ch);
```



- Table of Contents
- Chapter 1. Model Version
- Chapter 2. Root System
  - Interface
    - Input Signals
    - Output Signals
  - Blocks
    - Parameters
    - Block Execution Order
- Chapter 3. Subsystems
- Chapter 4. System Design Variables
- Chapter 5. Requirements
- Chapter 6. System Model Configuration
- Chapter 7. Glossary
- Chapter 8. About this Report

Chapter 2. Root System

Figure 2.1. crs_controller

Chapter 2. Subsystems

2.1. Engine

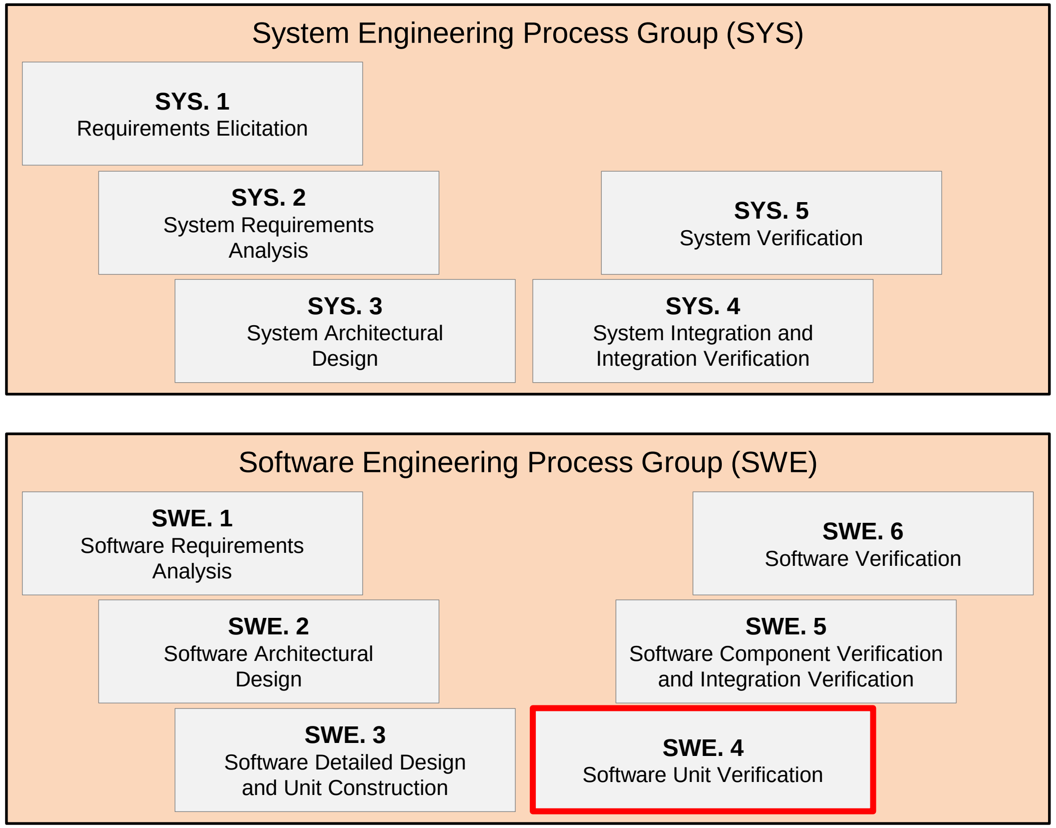
Figure 2.1. Engine

2.1.1. Ti

Table 2.1. slrgex_sf_car/Engine/Ti Properties

Property	Value
Type	Block
Block Type	Inport
Port number	1
Port dimensions (-1 for inherited)	-1
Sample time (-1 for inherited)	-1
Data type	Inherit: auto

Software Unit Verification



System Engineering Process Group (SYS)

SYS. 1
Requirements Elicitation

SYS. 2
System Requirements Analysis

SYS. 3
System Architectural Design

SYS. 4
System Integration and Integration Verification

SYS. 5
System Verification

Software Engineering Process Group (SWE)

SWE. 1
Software Requirements Analysis

SWE. 2
Software Architectural Design

SWE. 3
Software Detailed Design and Unit Construction

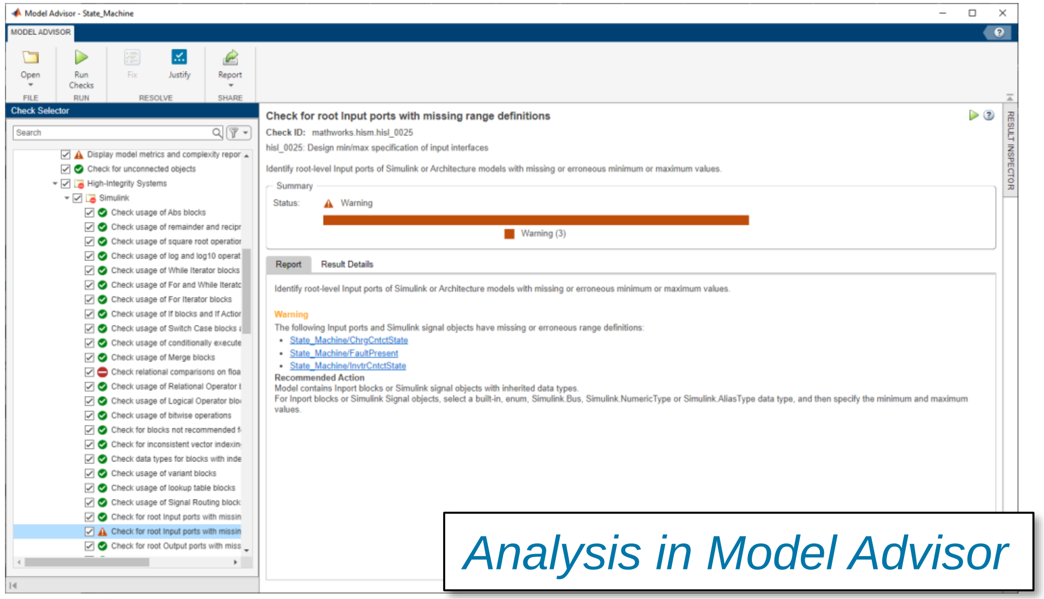
SWE. 4
Software Unit Verification

SWE. 5
Software Component Verification and Integration Verification

SWE. 6
Software Verification

Perform Static Verification of Software Units

Simulink Check



- ☑ Display model metrics and complexity repor
- ☑ Check for unconnected objects
- ☑ High-Integrity Systems
  - ☑ Simulink
    - ☑ Check usage of Abs blocks
    - ☑ Check usage of remainder and recipr
    - ☑ Check usage of square root operatior
    - ☑ Check usage of log and log10 operat
    - ☑ Check usage of While Iterator blocks
    - ☑ Check usage of For and While Iteratc
    - ☑ Check usage of For Iterator blocks
    - ☑ Check usage of If blocks and If Actior
    - ☑ Check usage of Switch Case blocks a
    - ☑ Check usage of conditionally execute
    - ☑ Check usage of Merge blocks
    - ☑ Check relational comparisons on floa
    - ☑ Check usage of Relational Operator b
    - ☑ Check usage of Logical Operator blo
    - ☑ Check usage of bitwise operations
    - ☑ Check for blocks not recommended f
    - ☑ Check for inconsistent vector indexin
    - ☑ Check data types for blocks with inde
    - ☑ Check usage of variant blocks
    - ☑ Check usage of lookup table blocks
    - ☑ Check usage of Signal Routing block
    - ☑ Check for root Input ports with missin
    - ☑ Check for root Input ports with missin
    - ☑ Check for root Output ports with miss

Check for root Input ports with missing range definitions

Check ID: mathworks.hism.hisl_0025

hisl_0025: Design min/max specification of input interfaces

Identify root-level Input ports of Simulink or Architecture models with missing or erroneous minimum or maximum values.

Summary

Status: Warning

Warning (3)

Report | Result Details

Identify root-level Input ports of Simulink or Architecture models with missing or erroneous minimum or maximum values.

Warning

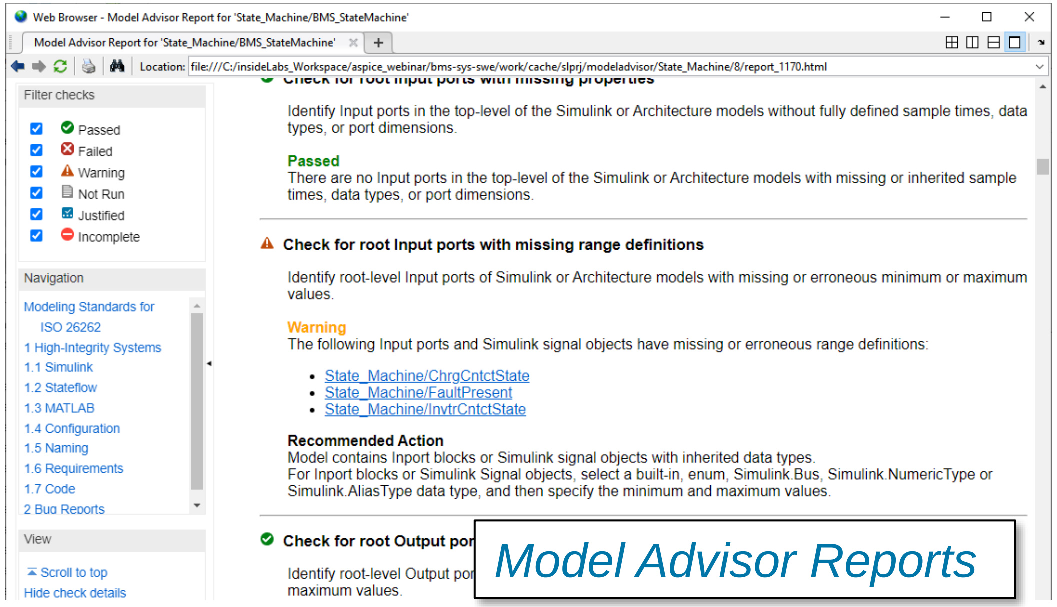
The following Input ports and Simulink signal objects have missing or erroneous range definitions:

- State_Machine/ChrgCntctState
- State_Machine/FaultPresent
- State_Machine/InvtrCntctState

Recommended Action

Model contains Inport blocks or Simulink signal objects with inherited data types.
For Inport blocks or Simulink Signal objects, select a built-in, enum, Simulink.Bus, Simulink.NumericType or Simulink.AliasType data type, and then specify the minimum and maximum values.

Analysis in Model Advisor



Web Browser - Model Advisor Report for 'State_Machine/BMS_StateMachine'
Model Advisor Report for 'State_Machine/BMS_StateMachine'
Location: file:///C:/insideLabs_Workspace/aspice_webinar/bms-sys-swe/work/cache/slprj/modeladvisor/State_Machine/8/report_1170.html

Filter checks

- ☑ Passed
- ☑ Failed
- ☑ Warning
- ☑ Not Run
- ☑ Justified
- ☑ Incomplete

Navigation

- Modeling Standards for ISO 26262
- 1 High-Integrity Systems
- 1.1 Simulink
- 1.2 Stateflow
- 1.3 MATLAB
- 1.4 Configuration
- 1.5 Naming
- 1.6 Requirements
- 1.7 Code
- 2 Bug Reports

View

- Scroll to top
- Hide check details

Check for root Input ports with missing properties

Identify Input ports in the top-level of the Simulink or Architecture models without fully defined sample times, data types, or port dimensions.

Passed

There are no Input ports in the top-level of the Simulink or Architecture models with missing or inherited sample times, data types, or port dimensions.

Check for root Input ports with missing range definitions

Identify root-level Input ports of Simulink or Architecture models with missing or erroneous minimum or maximum values.

Warning

The following Input ports and Simulink signal objects have missing or erroneous range definitions:

- State_Machine/ChrgCntctState
- State_Machine/FaultPresent
- State_Machine/InvtrCntctState

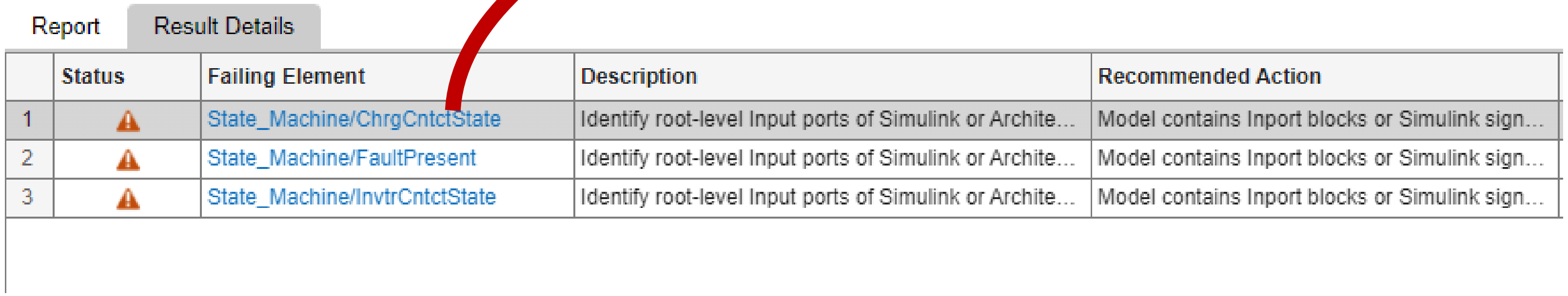
Recommended Action

Model contains Inport blocks or Simulink signal objects with inherited data types.
For Inport blocks or Simulink Signal objects, select a built-in, enum, Simulink.Bus, Simulink.NumericType or Simulink.AliasType data type, and then specify the minimum and maximum values.

Check for root Output por

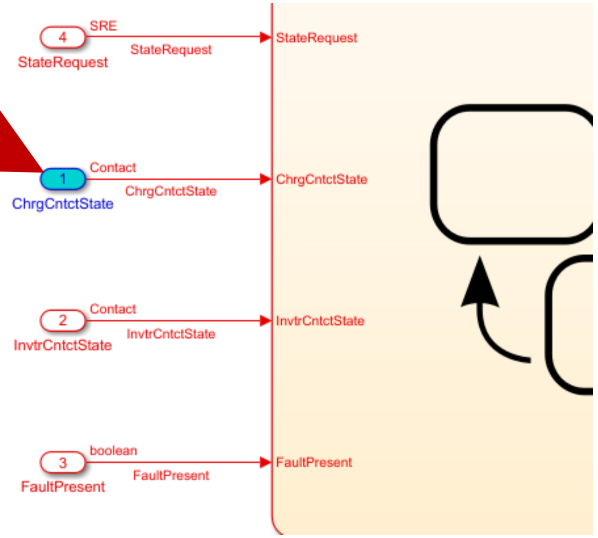
Identify root-level Output por
maximum values.

Model Advisor Reports



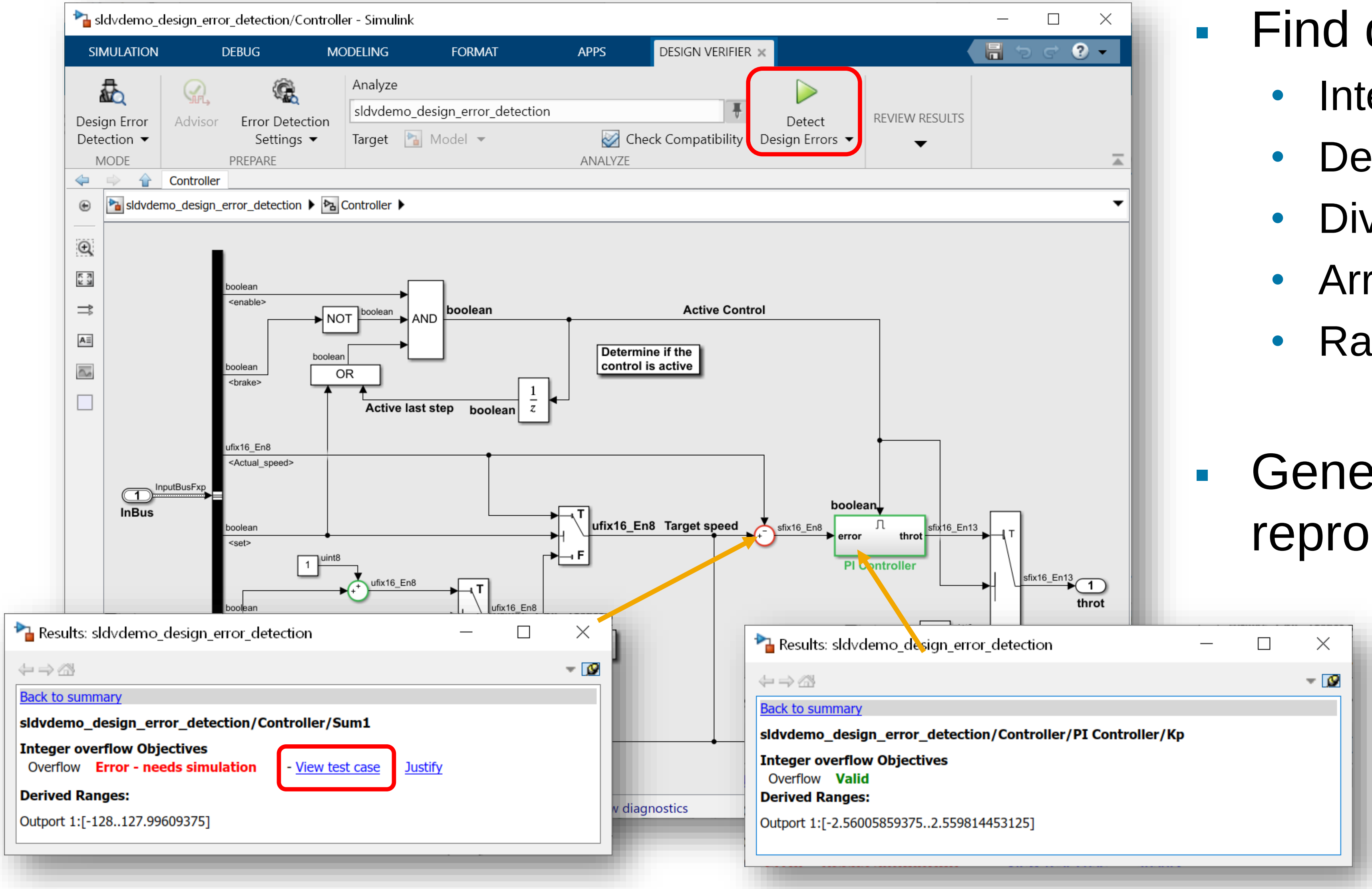
Report | Result Details

	Status	Failing Element	Description	Recommended Action
1	⚠	State_Machine/ChrgCntctState	Identify root-level Input ports of Simulink or Archite…	Model contains Inport blocks or Simulink sign…
2	⚠	State_Machine/FaultPresent	Identify root-level Input ports of Simulink or Archite…	Model contains Inport blocks or Simulink sign…
3	⚠	State_Machine/InvtrCntctState	Identify root-level Input ports of Simulink or Archite…	Model contains Inport blocks or Simulink sign…



Perform Static Verification of Software Units

Simulink Design Verifier



Results: sldvdemo_design_error_detection

Back to summary

sldvdemo_design_error_detection/Controller/Sum1

Integer overflow Objectives

Overflow **Error - needs simulation** - View test case Justify

Derived Ranges:

Outport 1:[-128..127.99609375]

Results: sldvdemo_design_error_detection

Back to summary

sldvdemo_design_error_detection/Controller/PI Controller/Kp

Integer overflow Objectives

Overflow **Valid**

Derived Ranges:

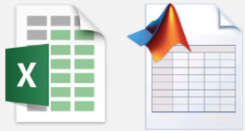
Outport 1:[-2.56005859375..2.559814453125]

- Find design errors
  - Integer overflow
  - Dead Logic
  - Division by zero
  - Array out-of-bounds
  - Range violations
- Generate counter example to reproduce error

Automation of Software Unit Testing using Simulink Test

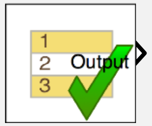
Test Case

Inputs

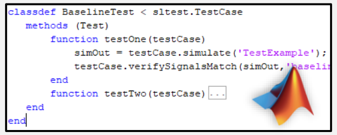


Data file (input)

Signal Editor

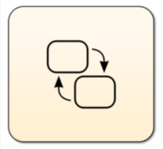


Test Sequence



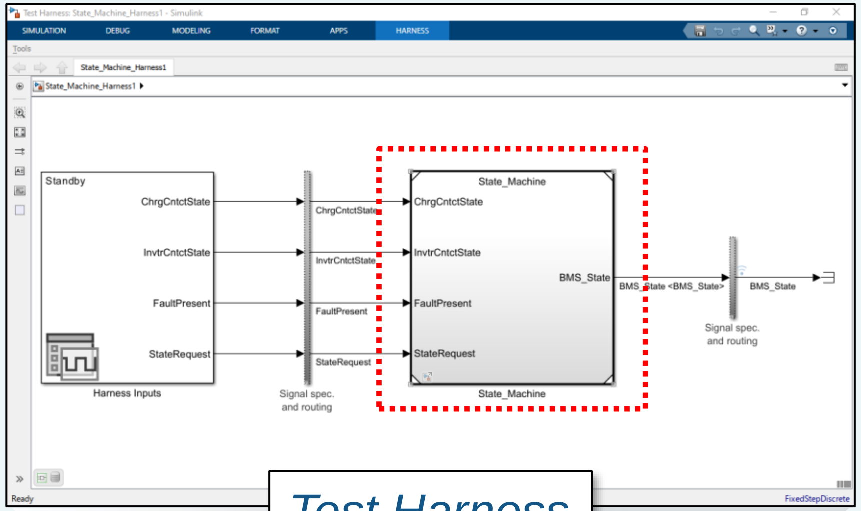
```
classdef BaselineTest < sltest.TestCase
    methods (Test)
        function testOne(testCase)
            simOut = testCase.simulate('TestExample');
            testCase.verifySignalsMatch(simOut,'baseli
        end
        function testTwo(testCase)...
    end
end
```

MATLAB Code

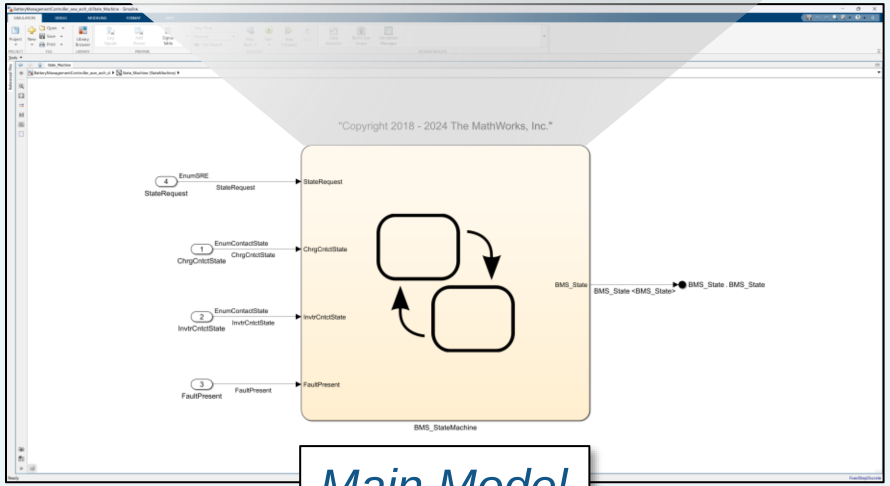


Stateflow

and more!

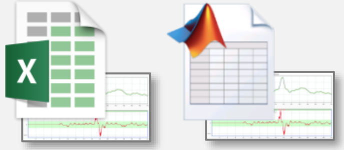


Test Harness

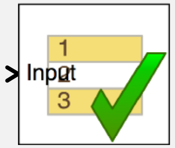


Main Model

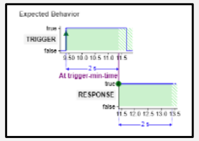
Assessments



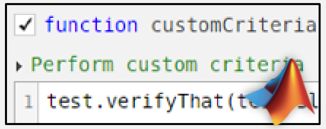
Data file (baseline)



Test Assessment



Temporal Assessment



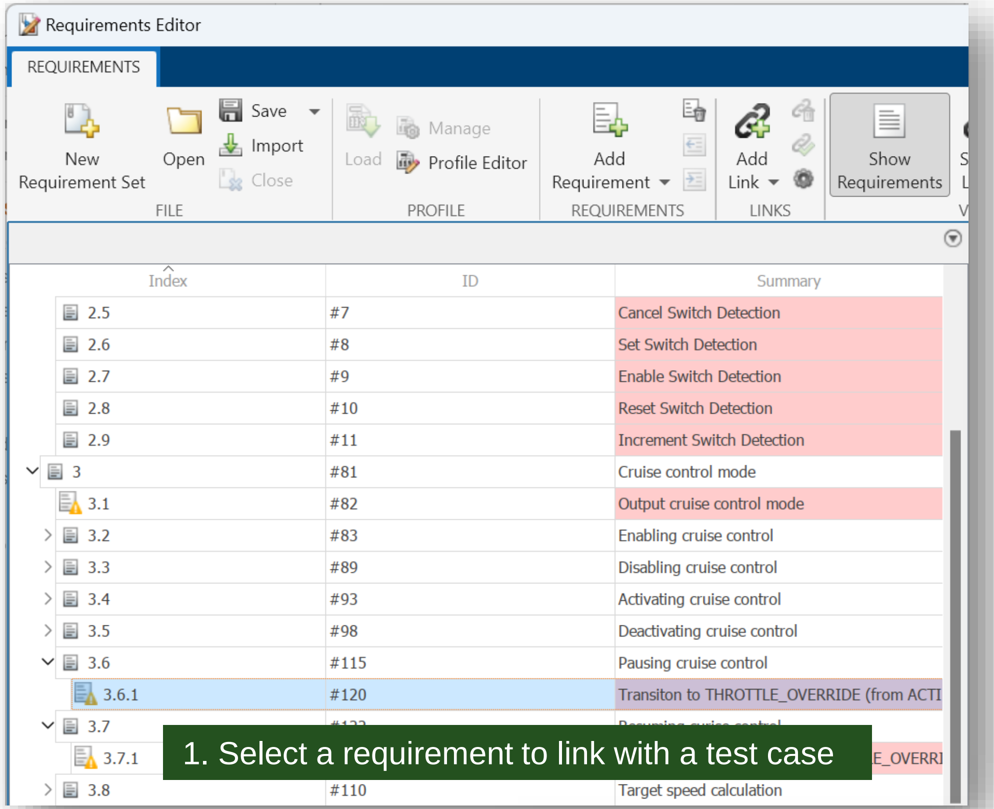
```
☑ function customCriteria
▸ Perform custom criteria
1 test.verifyThat(te
```

MATLAB Code

and more!

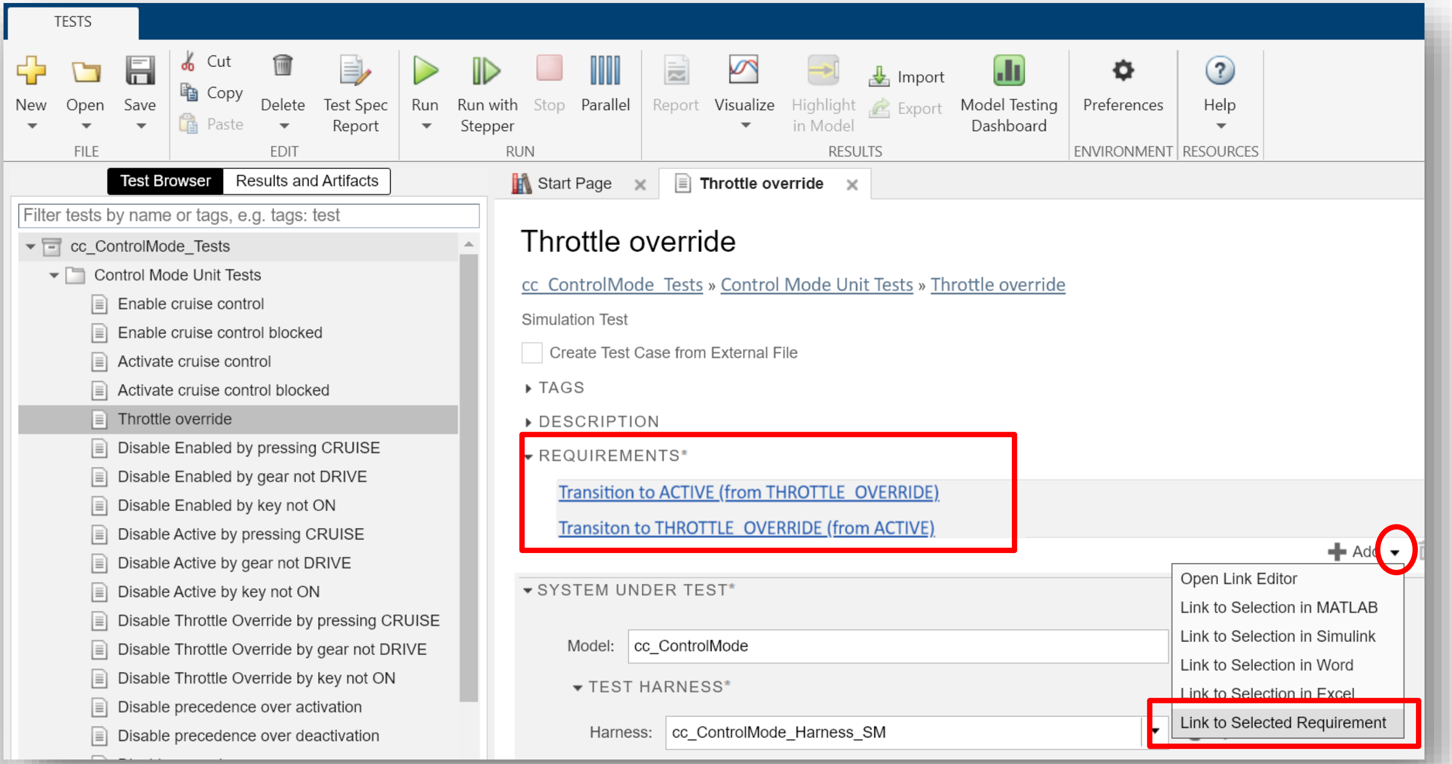
Track Verifications from Requirements

Requirements Based Testing



Index	ID	Summary
2.5	#7	Cancel Switch Detection
2.6	#8	Set Switch Detection
2.7	#9	Enable Switch Detection
2.8	#10	Reset Switch Detection
2.9	#11	Increment Switch Detection
3	#81	Cruise control mode
3.1	#82	Output cruise control mode
3.2	#83	Enabling cruise control
3.3	#89	Disabling cruise control
3.4	#93	Activating cruise control
3.5	#98	Deactivating cruise control
3.6	#115	Pausing cruise control
3.6.1	#120	Transiton to THROTTLE_OVERRIDE (from ACTI
3.7	[illegible]	Resuming cruise control
3.7.1	[illegible]	E_OVERRI
3.8	#110	Target speed calculation

1. Select a requirement to link with a test case



- cc_ControlMode_Tests
  - Control Mode Unit Tests
    - Enable cruise control
    - Enable cruise control blocked
    - Activate cruise control
    - Activate cruise control blocked
    - Throttle override
    - Disable Enabled by pressing CRUISE
    - Disable Enabled by gear not DRIVE
    - Disable Enabled by key not ON
    - Disable Active by pressing CRUISE
    - Disable Active by gear not DRIVE
    - Disable Active by key not ON
    - Disable Throttle Override by pressing CRUISE
    - Disable Throttle Override by gear not DRIVE
    - Disable Throttle Override by key not ON
    - Disable precedence over activation
    - Disable precedence over deactivation

Throttle override

cc_ControlMode_Tests » Control Mode Unit Tests » Throttle override

Simulation Test

□ Create Test Case from External File

TAGS

DESCRIPTION

REQUIREMENTS*

Transition to ACTIVE (from THROTTLE_OVERRIDE)

Transiton to THROTTLE_OVERRIDE (from ACTIVE)

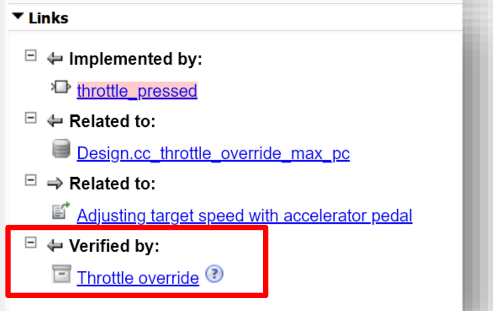
SYSTEM UNDER TEST*

Model: cc_ControlMode

TEST HARNESS*

Harness: cc_ControlMode_Harness_SM

2. Link the selected requirement in the test case



Links

Implemented by:

throttle_pressed

Related to:

Design.cc_throttle_override_max_pc

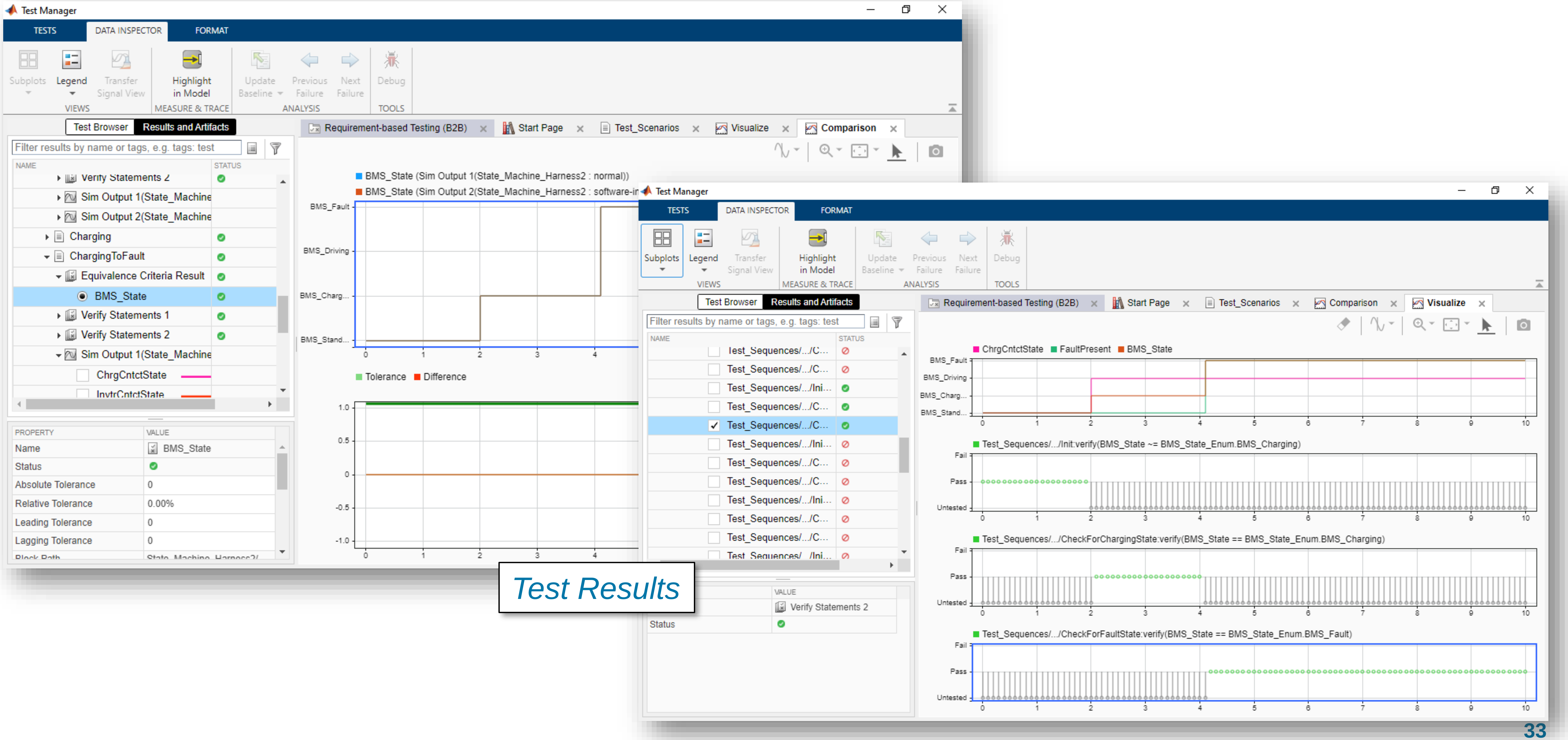
Related to:

Adjusting target speed with accelerator pedal

Verified by:

Throttle override

Test Software Units – Interactive Analysis of Results


Test Manager
TESTS
DATA INSPECTOR
FORMAT
Subplots
Legend
Transfer Signal View
VIEWS
Highlight in Model
MEASURE & TRACE
Update Baseline
Previous Failure
Next Failure
ANALYSIS
Debug
TOOLS
Test Browser
Results and Artifacts
Filter results by name or tags, e.g. tags: test
NAME
STATUS
Requirement-based Testing (B2B)
Start Page
Test_Scenarios
Visualize
Comparison
BMS_State (Sim Output 1(State_Machine_Harness2 : normal))
BMS_State (Sim Output 2(State_Machine_Harness2 : software-in
BMS_Fault
BMS_Driving
BMS_Charg...
BMS_Stand...
0
1
2
3
4
Tolerance
Difference
1.0
0.5
0
-0.5
-1.0
ChrgCntctState
FaultPresent
BMS_State
Test_Sequences/.../Init:verify(BMS_State ~= BMS_State_Enum.BMS_Charging)
Test_Sequences/.../CheckForChargingState:verify(BMS_State == BMS_State_Enum.BMS_Charging)
Test_Sequences/.../CheckForFaultState:verify(BMS_State == BMS_State_Enum.BMS_Fault)
Fail
Pass
Untested
5
6
7
8
9
10

NAME	STATUS
Verify Statements 2	✓
Sim Output 1(State_Machine	
Sim Output 2(State_Machine	
Charging	✓
ChargingToFault	✓
Equivalence Criteria Result	✓
● BMS_State	✓
Verify Statements 1	✓
Verify Statements 2	✓
Sim Output 1(State_Machine	
□ ChrgCntctState	
□ InvtrCntctState	

PROPERTY	VALUE
Name	BMS_State
Status	✓
Absolute Tolerance	0
Relative Tolerance	0.00%
Leading Tolerance	0
Lagging Tolerance	0
Block Path	State_Machine_Harness2/

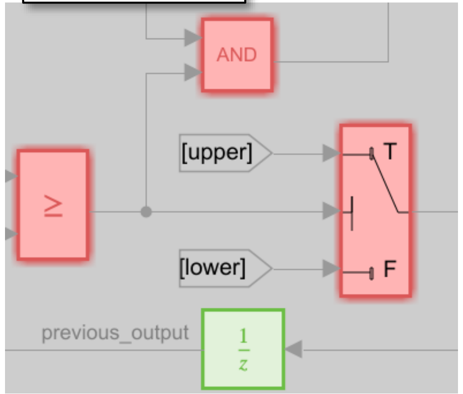
NAME	STATUS
□ Test_Sequences/.../C...	⊘
□ Test_Sequences/.../C...	⊘
□ Test_Sequences/.../Ini...	✓
□ Test_Sequences/.../C...	✓
☑ Test_Sequences/.../C...	✓
□ Test_Sequences/.../Ini...	⊘
□ Test_Sequences/.../C...	⊘
□ Test_Sequences/.../C...	⊘
□ Test_Sequences/.../Ini...	⊘
□ Test_Sequences/.../C...	⊘
□ Test_Sequences/.../C...	⊘
□ Test_Sequences/.../Ini...	⊘

	VALUE
	Verify Statements 2
Status	✓

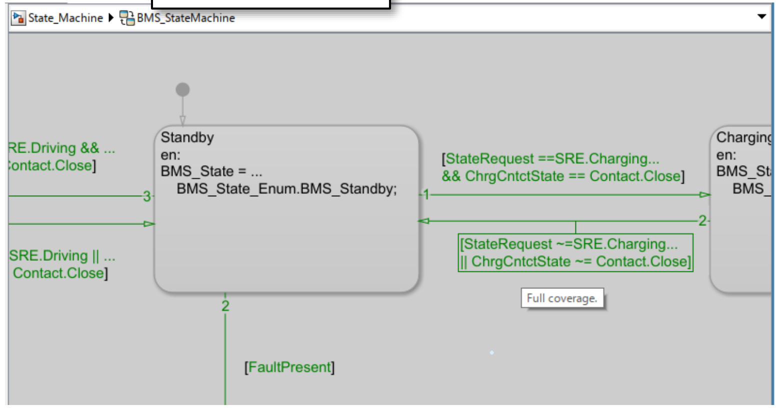
Test Results

Test Software Units – Structural Coverage

Simulink

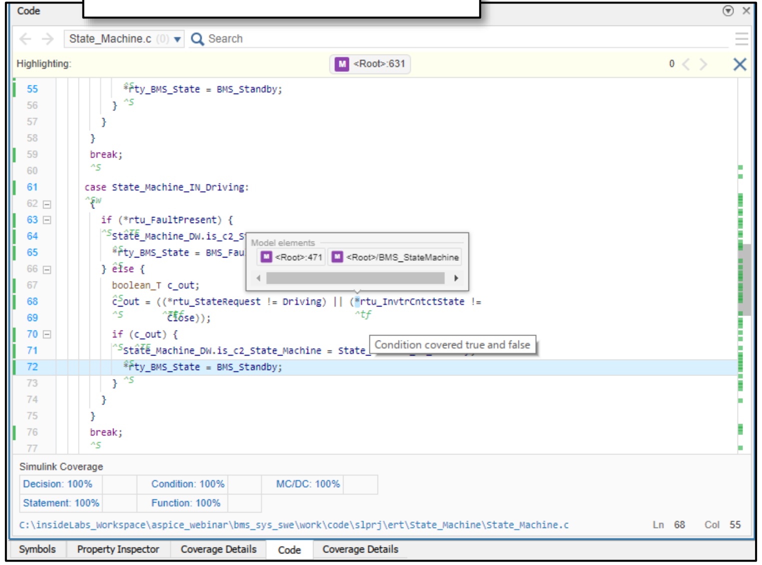


Stateflow


State_Machine ▸ BMS_StateMachine
Standby
en:
BMS_State = ...
BMS_State_Enum.BMS_Standby;
RE.Driving && ...
ontact.Close]
SRE.Driving || ...
Contact.Close]
[StateRequest ==SRE.Charging...
&& ChrgCntctState == Contact.Close]
[StateRequest ~=SRE.Charging...
|| ChrgCntctState ~= Contact.Close]
Full coverage.
Charging
en:
BMS_St
BMS_
[FaultPresent]
1
2
3
2

- Identify testing gaps
- Missing requirements
- Unintended Functionality

Generated Code



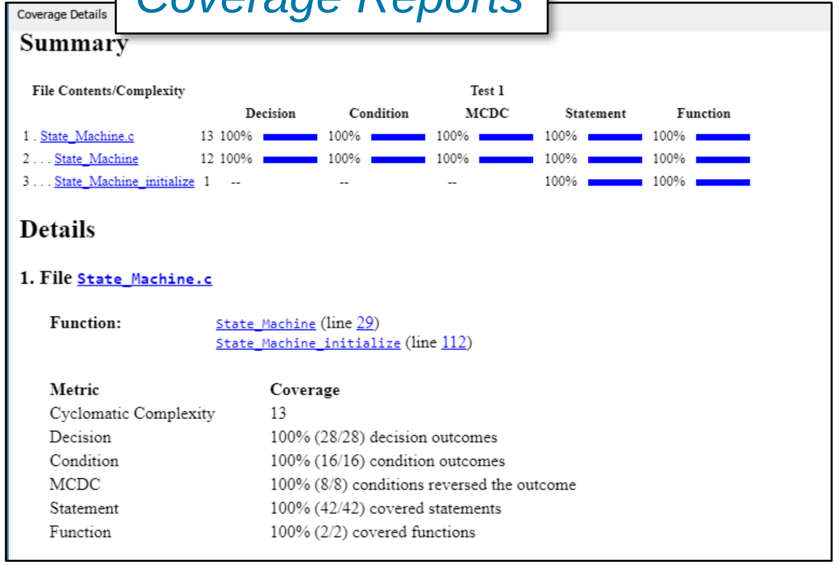
```
          *rty_BMS_State = BMS_Standby;
        }
      }
    }
    break;

   case State_Machine_IN_Driving:
    {
     if (*rtu_FaultPresent) {
      State_Machine_DW.is_c2_S
      *rty_BMS_State = BMS_Fau
     } else {
      boolean_T c_out;
      c_out = ((*rtu_StateRequest != Driving) || (*rtu_InvtrCntctState !=
               Close));
      if (c_out) {
       State_Machine_DW.is_c2_State_Machine = State_
        *rty_BMS_State = BMS_Standby;
       }
     }
    }
    break;
```

Simulink Coverage

Decision: 100%	Condition: 100%	MC/DC: 100%
Statement: 100%	Function: 100%	

C:\insideLabs_Workspace\aspice_webinar\bms_sys_swe\work\code\slprj\ert\State_Machine\State_Machine.c Ln 68 Col 55

Coverage Reports



Summary

File Contents/Complexity		Test 1 Decision	Condition	MCDC	Statement	Function
1 . State_Machine.c	13	100%	100%	100%	100%	100%
2 . . . State_Machine	12	100%	100%	100%	100%	100%
3 . . . State_Machine_initialize	1	--	--	--	100%	100%

Details

1. File State_Machine.c

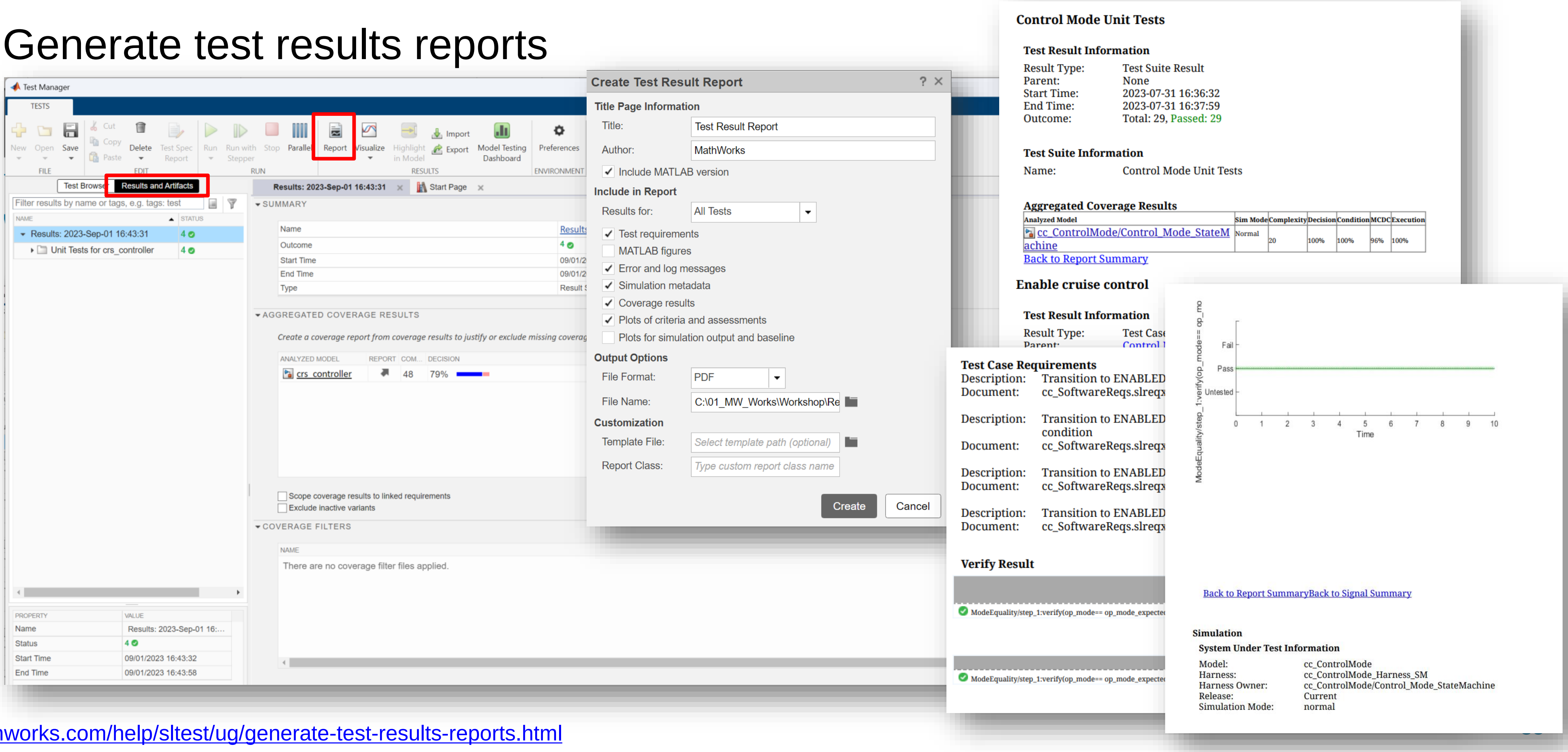
Function: State_Machine (line 29)
State_Machine_initialize (line 112)

Metric	Coverage
Cyclomatic Complexity	13
Decision	100% (28/28) decision outcomes
Condition	100% (16/16) condition outcomes
MCDC	100% (8/8) conditions reversed the outcome
Statement	100% (42/42) covered statements
Function	100% (2/2) covered functions

Reporting Test Results

Generate Test Results Reports

- Generate test results reports



Filter results by name or tags, e.g. tags: test

NAME	STATUS
Results: 2023-Sep-01 16:43:31	4 ✓
Unit Tests for crs_controller	4 ✓

PROPERTY	VALUE
Name	Results: 2023-Sep-01 16:...
Status	4 ✓
Start Time	09/01/2023 16:43:32
End Time	09/01/2023 16:43:58

SUMMARY

Name	Results
Outcome	4 ✓
Start Time	09/01/2
End Time	09/01/2
Type	Result S

AGGREGATED COVERAGE RESULTS

Create a coverage report from coverage results to justify or exclude missing coverag

ANALYZED MODEL	REPORT	COM...	DECISION
crs_controller		48	79%

□ Scope coverage results to linked requirements
□ Exclude inactive variants

COVERAGE FILTERS

NAME
There are no coverage filter files applied.

Create Test Result Report

Title Page Information
Title: Test Result Report
Author: MathWorks
☑ Include MATLAB version

Include in Report
Results for: All Tests
☑ Test requirements
□ MATLAB figures
☑ Error and log messages
☑ Simulation metadata
☑ Coverage results
☑ Plots of criteria and assessments
□ Plots for simulation output and baseline

Output Options
File Format: PDF
File Name: C:\01_MW_Works\Workshop\Re

Customization
Template File: *Select template path (optional)*
Report Class: *Type custom report class name*

Create | Cancel

Control Mode Unit Tests

Test Result Information
Result Type: Test Suite Result
Parent: None
Start Time: 2023-07-31 16:36:32
End Time: 2023-07-31 16:37:59
Outcome: Total: 29, Passed: 29

Test Suite Information
Name: Control Mode Unit Tests

Aggregated Coverage Results

Analyzed Model	Sim Mode	Complexity	Decision	Condition	MCDC	Execution
cc_ControlMode/Control_Mode_StateMachine	Normal	20	100%	100%	96%	100%

Back to Report Summary

Enable cruise control

Test Result Information
Result Type: Test Case
Parent: Control I

Test Case Requirements
Description: Transition to ENABLED
Document: cc_SoftwareReqs.slreqx

Description: Transition to ENABLED condition
Document: cc_SoftwareReqs.slreqx

Description: Transition to ENABLED
Document: cc_SoftwareReqs.slreqx

Description: Transition to ENABLED
Document: cc_SoftwareReqs.slreqx

Verify Result

✓ ModeEquality/step_1:verify(op_mode== op_mode_expecte

✓ ModeEquality/step_1:verify(op_mode== op_mode_expecte

Back to Report SummaryBack to Signal Summary

Simulation

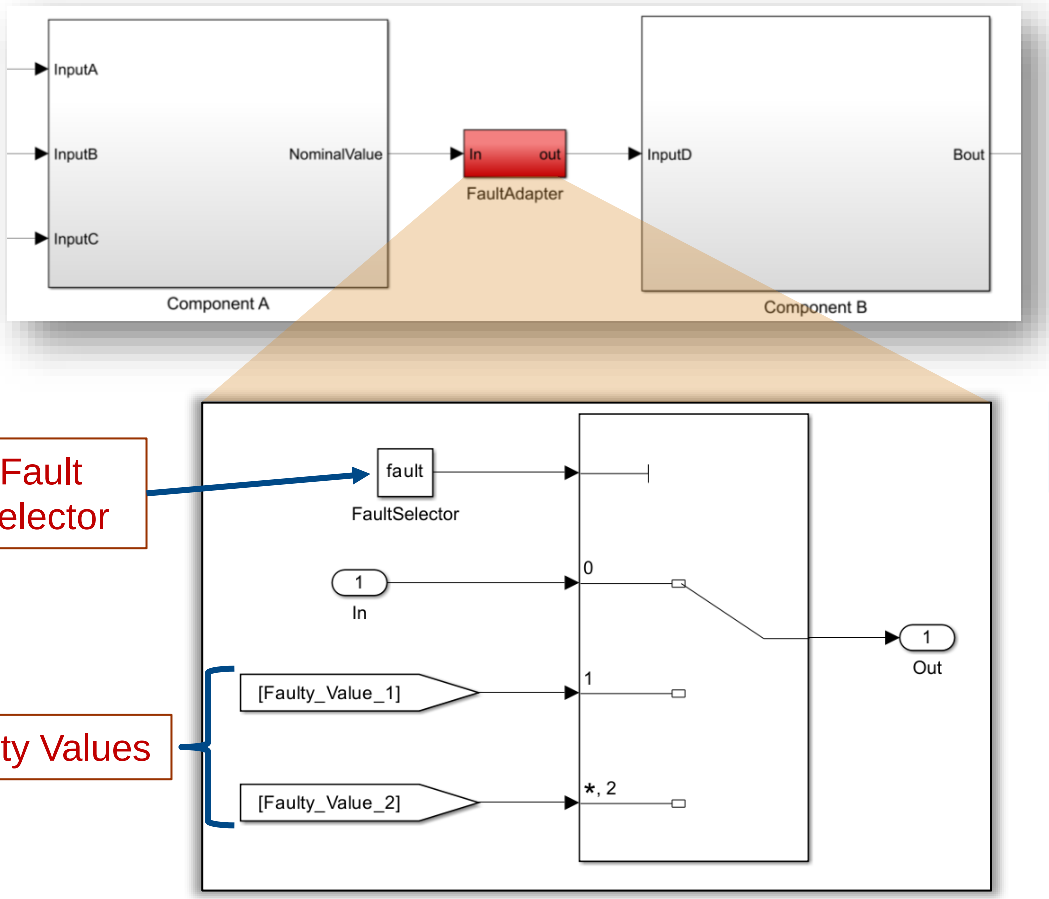
System Under Test Information
Model: cc_ControlMode
Harness: cc_ControlMode_Harness_SM
Harness Owner: cc_ControlMode/Control_Mode_StateMachine
Release: Current
Simulation Mode: normal

https://kr.mathworks.com/help/sltest/ug/generate-test-results-reports.html

Fault Injection Testing

Simulink Fault Analyzer

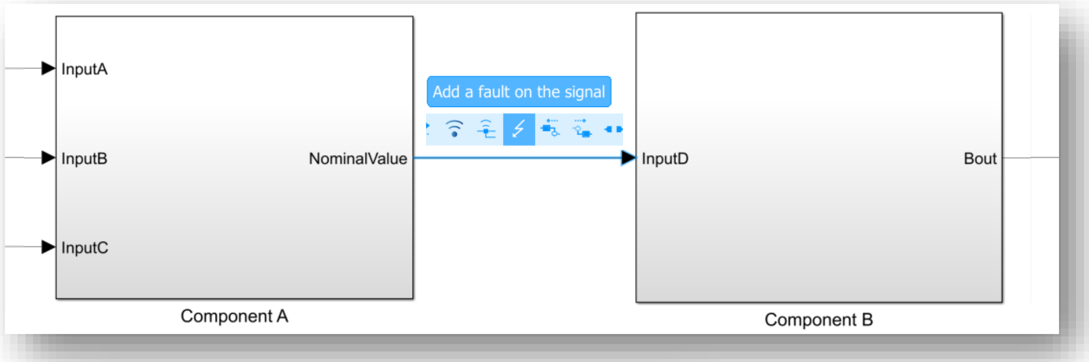
- Ad-hoc Fault Modeling in Traditional MBD

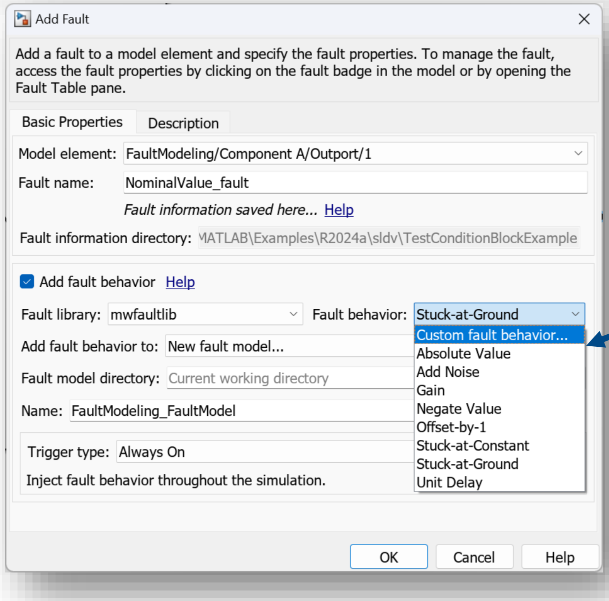


Fault Selector

Faulty Values

- Fault Modeling using Simulink Fault Analyzer




Add Fault

Add a fault to a model element and specify the fault properties. To manage the fault, access the fault properties by clicking on the fault badge in the model or by opening the Fault Table pane.

Basic Properties | Description

Model element: FaultModeling/Component A/Outport/1

Fault name: NominalValue_fault

Fault information saved here... Help

Fault information directory: MATLAB\Examples\R2024a\sldv\TestConditionBlockExample

☑ Add fault behavior Help

Fault library: mwfaultlib

Fault behavior: Stuck-at-Ground

Add fault behavior to: New fault model...

Fault model directory: Current working directory

Name: FaultModeling_FaultModel

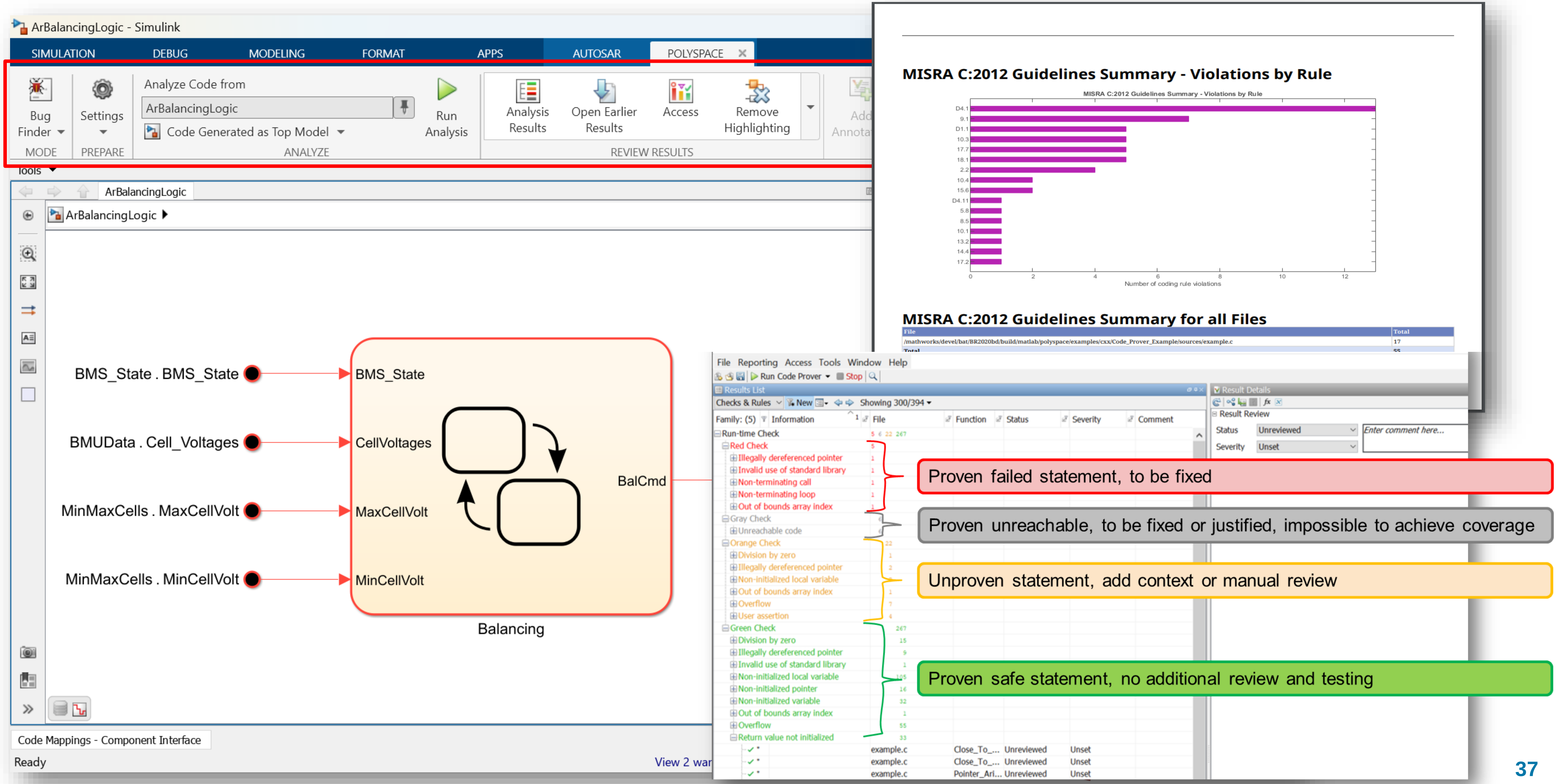
Trigger type: Always On

Inject fault behavior throughout the simulation.

Select a fault behavior or, design a custom fault

https://kr.mathworks.com/products/simulink-fault-analyzer.html

Perform Static Code Verification of Software Units



MISRA C:2012 Guidelines Summary - Violations by Rule

MISRA C:2012 Guidelines Summary - Violations by Rule
D4.1
9.1
D1.1
10.3
17.7
18.1
2.2
10.4
15.6
D4.11
5.8
8.5
10.1
13.2
14.4
17.2
0
2
4
6
8
10
12
Number of coding rule violations

MISRA C:2012 Guidelines Summary for all Files

File	Total
/mathworks/devel/bat/BR2020bd/build/matlab/polyspace/examples/cxx/Code_Prover_Example/sources/example.c	17
Total	55

Family: (5)	Information	File	Function	Status	Severity	Comment
Run-time Check	5 6 22 267					
Red Check	5					
Illegally dereferenced pointer	1					
Invalid use of standard library	1					
Non-terminating call	1					
Non-terminating loop	1					
Out of bounds array index	1					
Gray Check	6					
Unreachable code	6					
Orange Check	22					
Division by zero	1					
Illegally dereferenced pointer	2					
Non-initialized local variable	7					
Out of bounds array index	1					
Overflow	7					
User assertion	4					
Green Check	267					
Division by zero	15					
Illegally dereferenced pointer	9					
Invalid use of standard library	1					
Non-initialized local variable	105					
Non-initialized pointer	16					
Non-initialized variable	32					
Out of bounds array index	1					
Overflow	55					
Return value not initialized	33					
✓ *		example.c	Close_To_...	Unreviewed	Unset	
✓ *		example.c	Close_To_...	Unreviewed	Unset	
✓ *		example.c	Pointer_Ari...	Unreviewed	Unset	

Proven failed statement, to be fixed

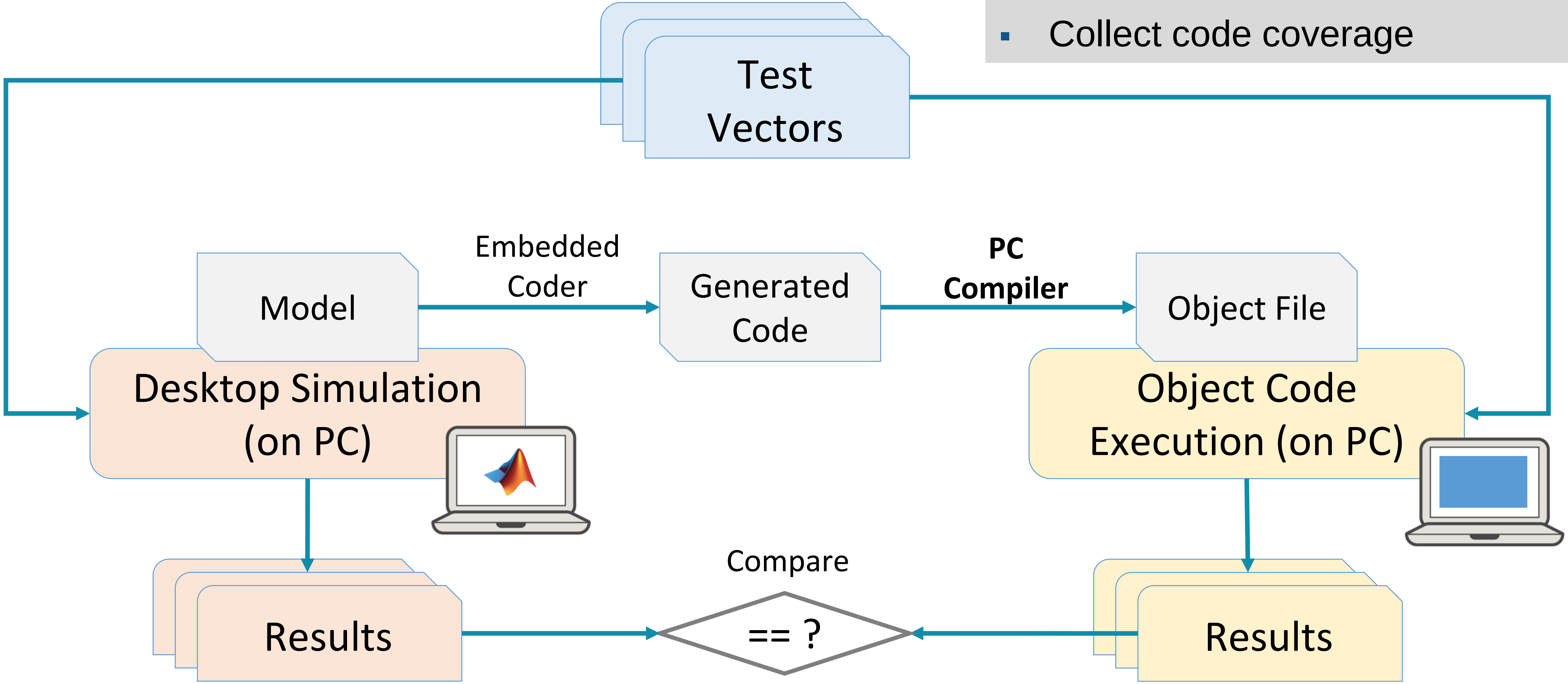
Proven unreachable, to be fixed or justified, impossible to achieve coverage

Unproven statement, add context or manual review

Proven safe statement, no additional review and testing

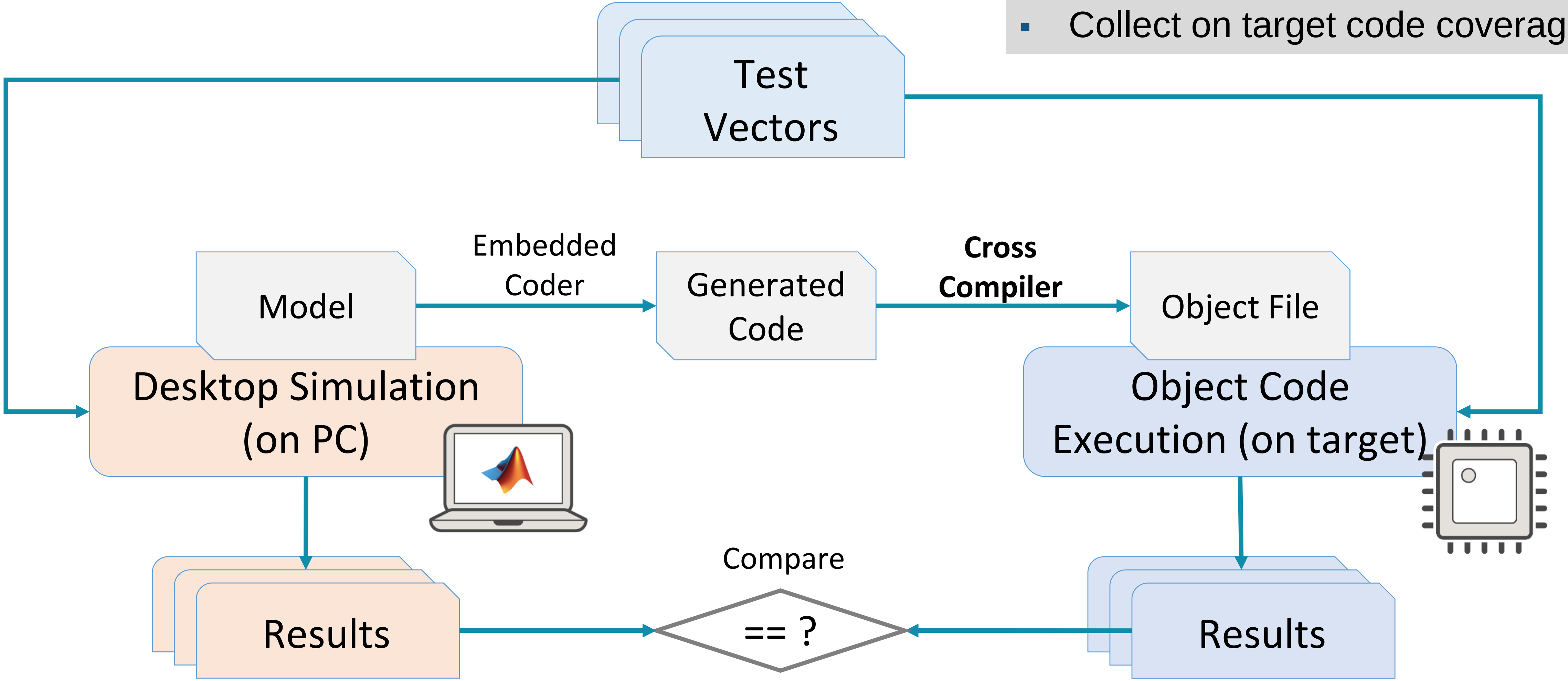
Software In the Loop (SIL) Testing

- Show equivalence, model to code
- Assess code execution time
- Collect code coverage



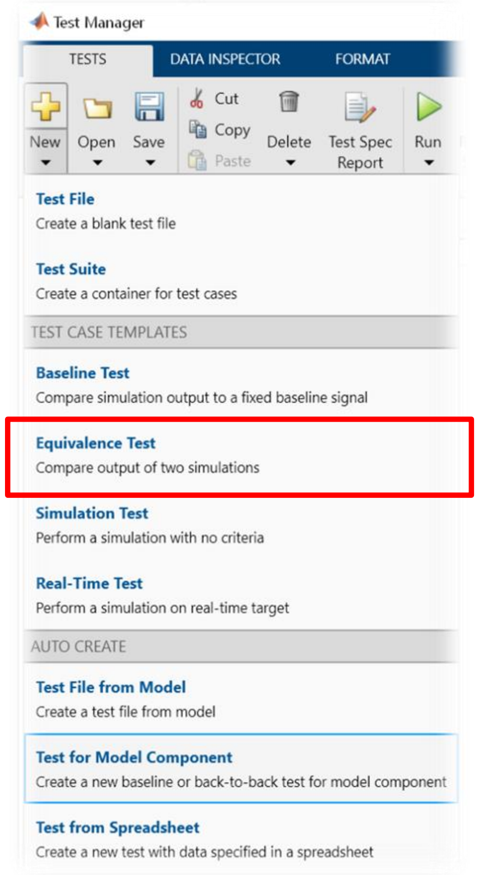
Processor In the Loop (PIL) Testing

- Verify numerical equivalence
- Assess target execution time
- Collect on target code coverage



Automate Test Creation for Equivalence Test

Simulink Test



Test File
Create a blank test file

Test Suite
Create a container for test cases

TEST CASE TEMPLATES

Baseline Test
Compare simulation output to a fixed baseline signal

Equivalence Test
Compare output of two simulations

Simulation Test
Perform a simulation with no criteria

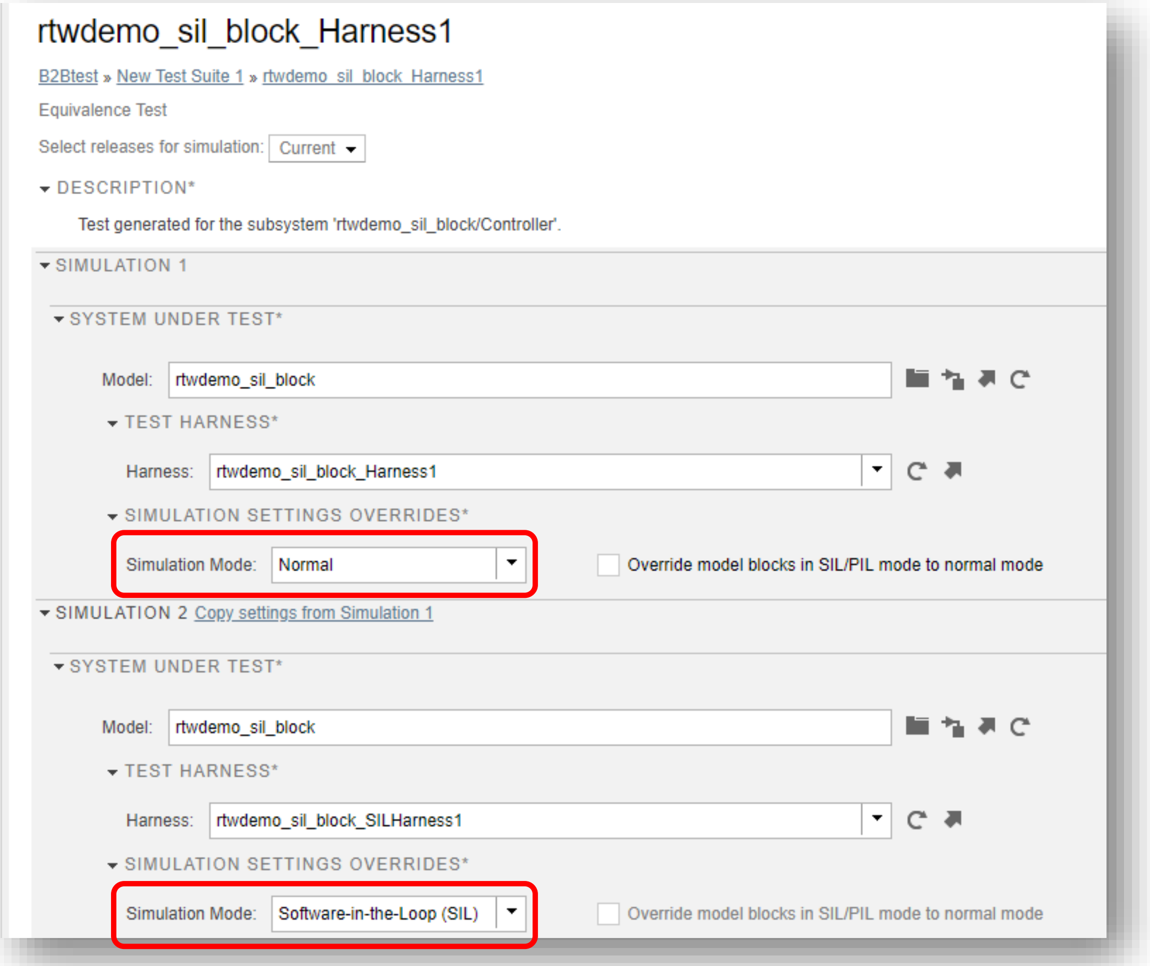
Real-Time Test
Perform a simulation on real-time target

AUTO CREATE

Test File from Model
Create a test file from model

Test for Model Component
Create a new baseline or back-to-back test for model component

Test from Spreadsheet
Create a new test with data specified in a spreadsheet



rtwdemo_sil_block_Harness1

B2Btest » New Test Suite 1 » rtwdemo_sil_block_Harness1

Equivalence Test

Select releases for simulation: Current

▾ DESCRIPTION*

Test generated for the subsystem 'rtwdemo_sil_block/Controller'.

▾ SIMULATION 1

▾ SYSTEM UNDER TEST*

Model: rtwdemo_sil_block

▾ TEST HARNESS*

Harness: rtwdemo_sil_block_Harness1

▾ SIMULATION SETTINGS OVERRIDES*

Simulation Mode: Normal

□ Override model blocks in SIL/PIL mode to normal mode

▾ SIMULATION 2 Copy settings from Simulation 1

▾ SYSTEM UNDER TEST*

Model: rtwdemo_sil_block

▾ TEST HARNESS*

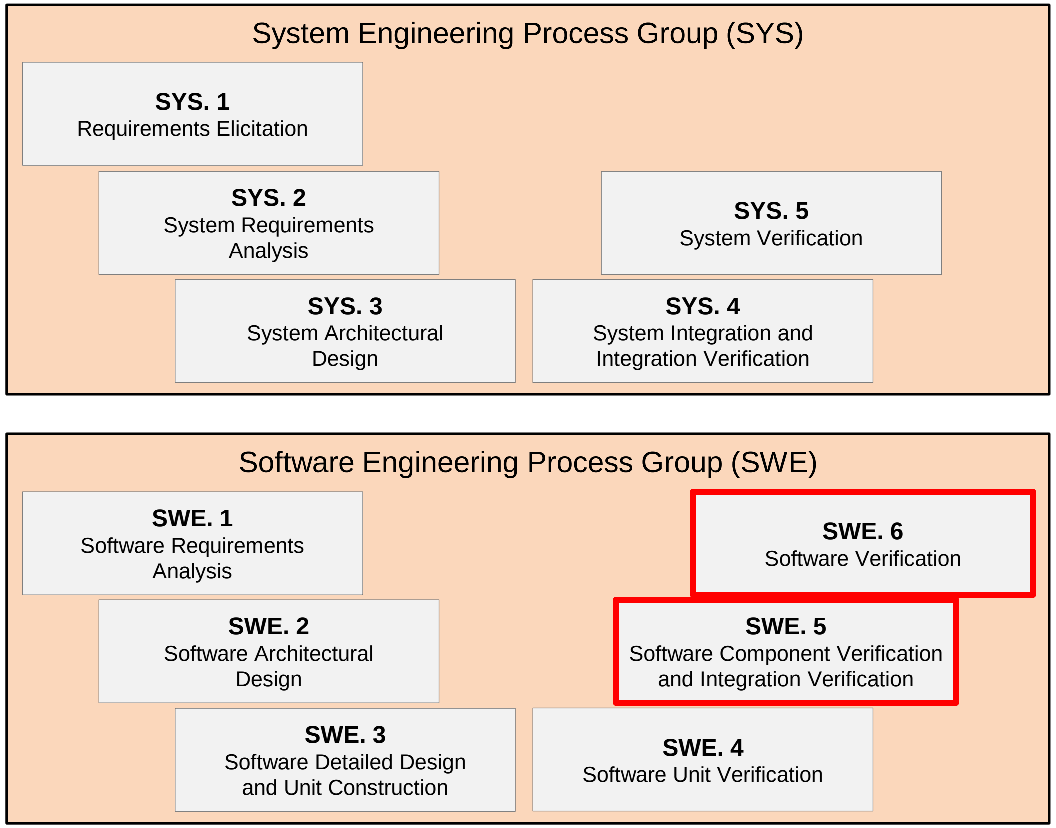
Harness: rtwdemo_sil_block_SILHarness1

▾ SIMULATION SETTINGS OVERRIDES*

Simulation Mode: Software-in-the-Loop (SIL)

□ Override model blocks in SIL/PIL mode to normal mode

Software Verification



System Engineering Process Group (SYS)

SYS. 1
Requirements Elicitation

SYS. 2
System Requirements Analysis

SYS. 3
System Architectural Design

SYS. 4
System Integration and Integration Verification

SYS. 5
System Verification

Software Engineering Process Group (SWE)

SWE. 1
Software Requirements Analysis

SWE. 2
Software Architectural Design

SWE. 3
Software Detailed Design and Unit Construction

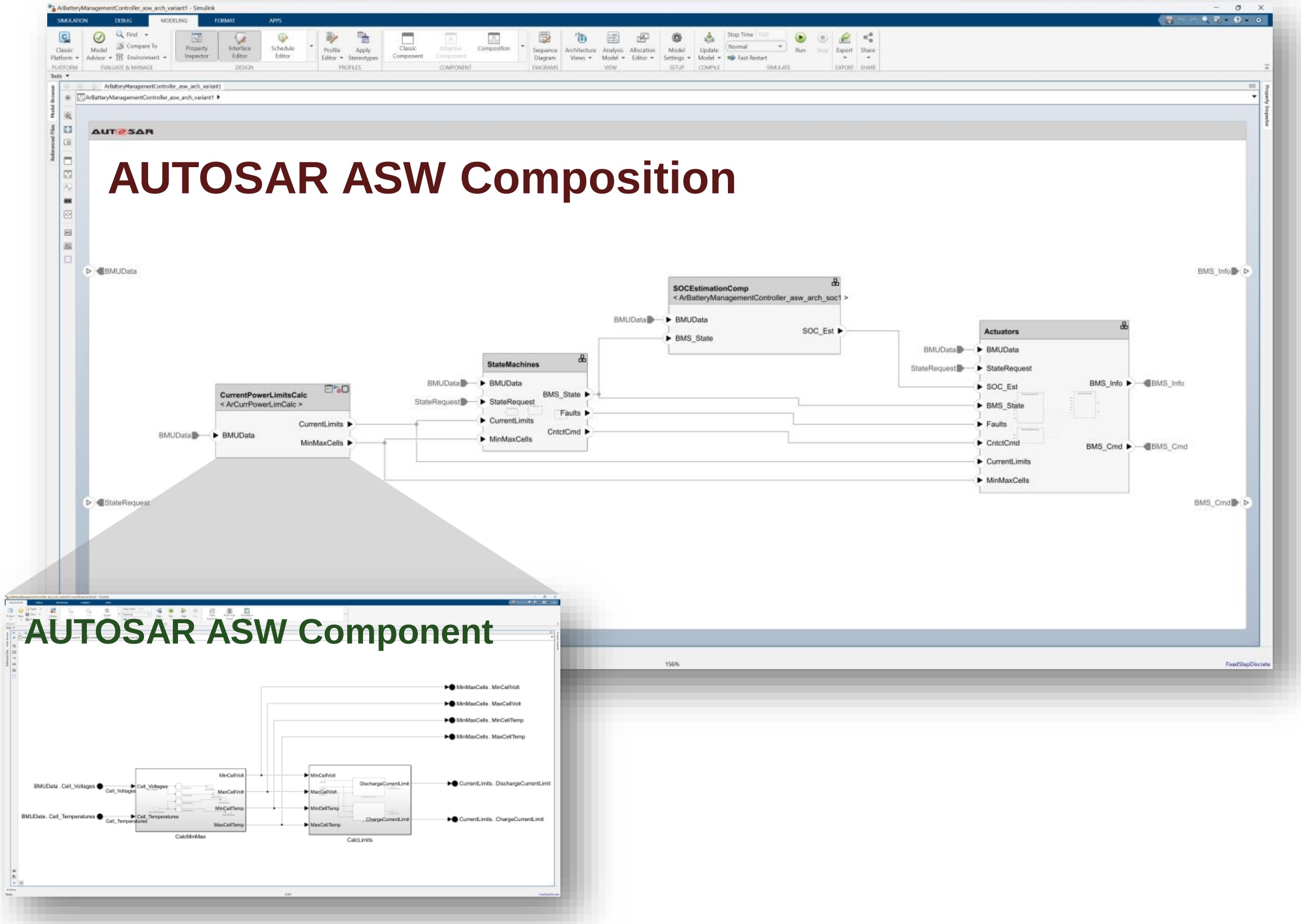
SWE. 4
Software Unit Verification

SWE. 5
Software Component Verification and Integration Verification

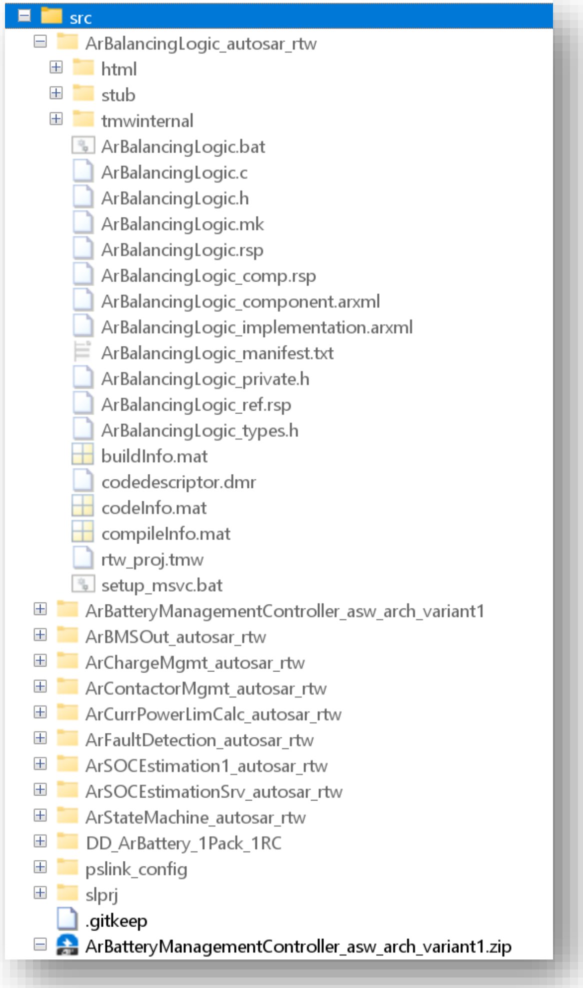
SWE. 6
Software Verification

Integrate Software Units

- AUTOSAR ASW composition and the code generation



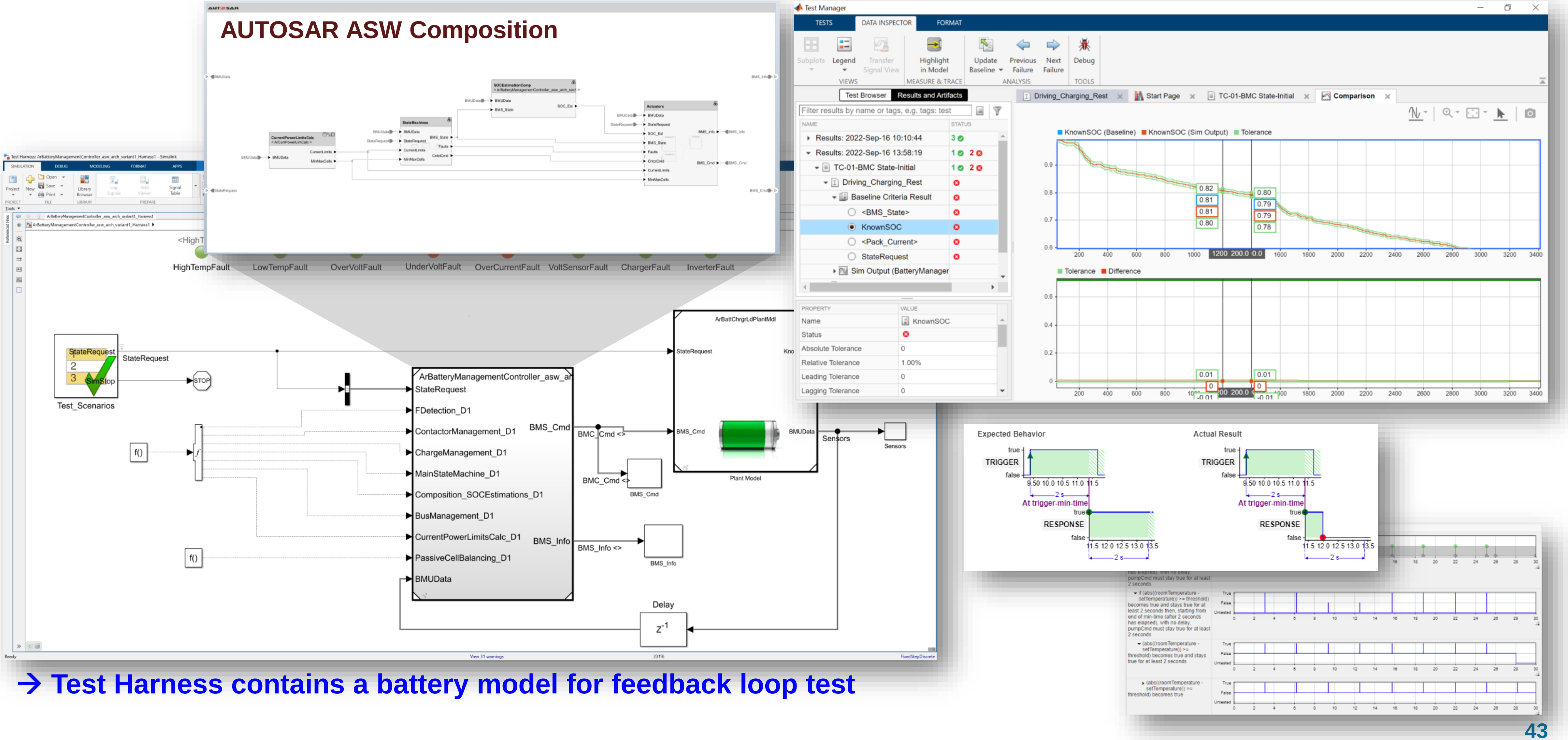
code generation



- src
  - ArBalancingLogic_autosar_rtw
    - html
    - stub
    - tmwinternal
    - ArBalancingLogic.bat
    - ArBalancingLogic.c
    - ArBalancingLogic.h
    - ArBalancingLogic.mk
    - ArBalancingLogic.rsp
    - ArBalancingLogic_comp.rsp
    - ArBalancingLogic_component.arxml
    - ArBalancingLogic_implementation.arxml
    - ArBalancingLogic_manifest.txt
    - ArBalancingLogic_private.h
    - ArBalancingLogic_ref.rsp
    - ArBalancingLogic_types.h
    - buildInfo.mat
    - codedescriptor.dmr
    - codeInfo.mat
    - compileInfo.mat
    - rtw_proj.tmw
    - setup_msvc.bat
  - ArBatteryManagementController_asw_arch_variant1
  - ArBMSOut_autosar_rtw
  - ArChargeMgmt_autosar_rtw
  - ArContactorMgmt_autosar_rtw
  - ArCurrPowerLimCalc_autosar_rtw
  - ArFaultDetection_autosar_rtw
  - ArSOCEstimation1_autosar_rtw
  - ArSOCEstimationSrv_autosar_rtw
  - ArStateMachine_autosar_rtw
  - DD_ArBattery_1Pack_1RC
  - pslink_config
  - slprj
  - .gitkeep
  - ArBatteryManagementController_asw_arch_variant1.zip

Perform Software Integration Test

Simulink Check

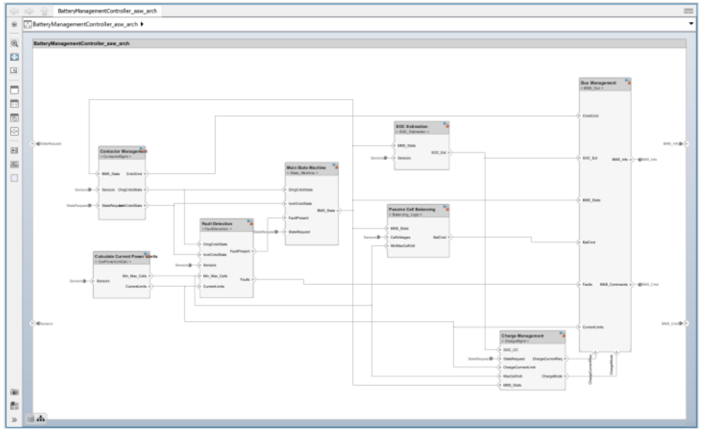


NAME	STATUS
Results: 2022-Sep-16 10:10:44	3 ✓
Results: 2022-Sep-16 13:58:19	1 ✓ 2 ✗
TC-01-BMC State-Initial	1 ✓ 2 ✗
Driving_Charging_Rest	✗
Baseline Criteria Result	✗
○ <BMS_State>	✗
● KnownSOC	✗
○ <Pack_Current>	✗
○ StateRequest	✗
Sim Output (BatteryManager	

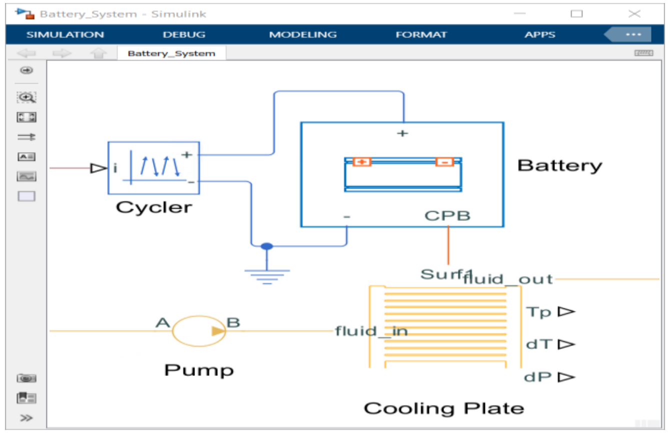
PROPERTY	VALUE
Name	KnownSOC
Status	✗
Absolute Tolerance	0
Relative Tolerance	1.00%
Leading Tolerance	0
Lagging Tolerance	0

→ Test Harness contains a battery model for feedback loop test

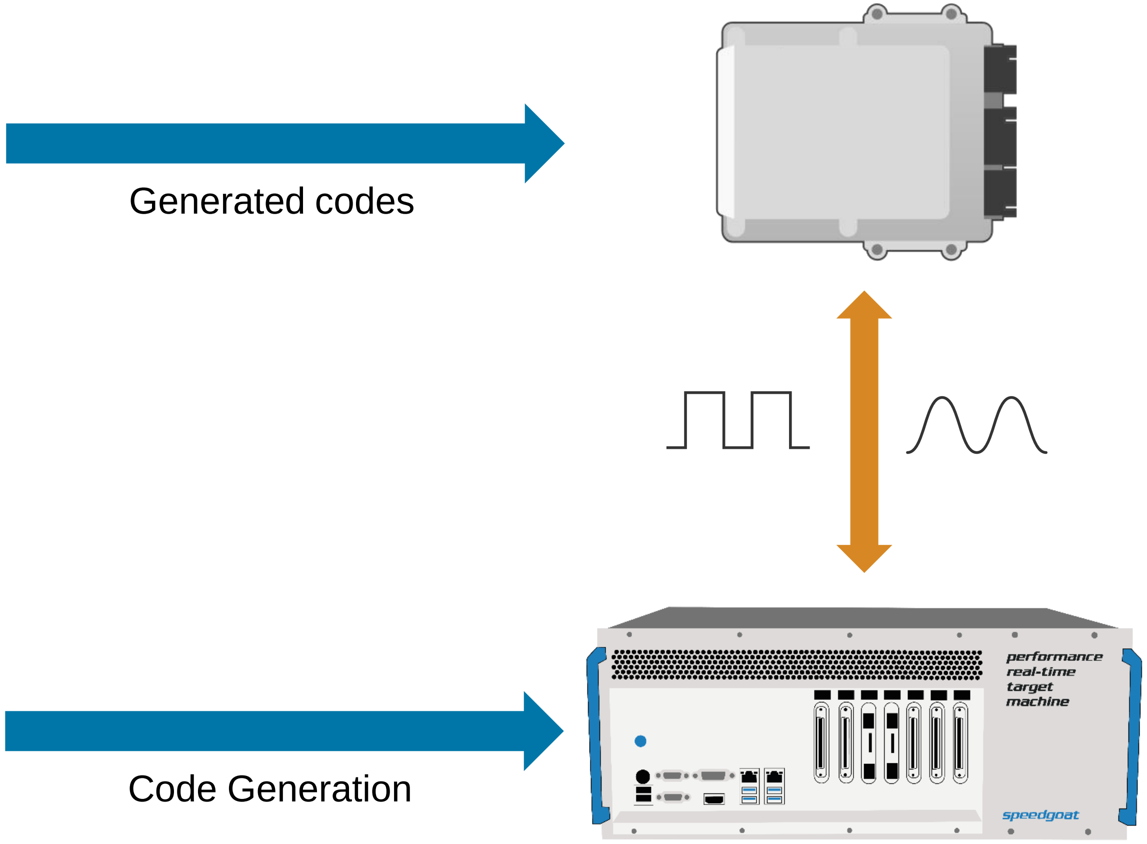
Test Integrated Software – Hardware-in-the-Loop Testing



Battery Management Controller Model



Battery Model

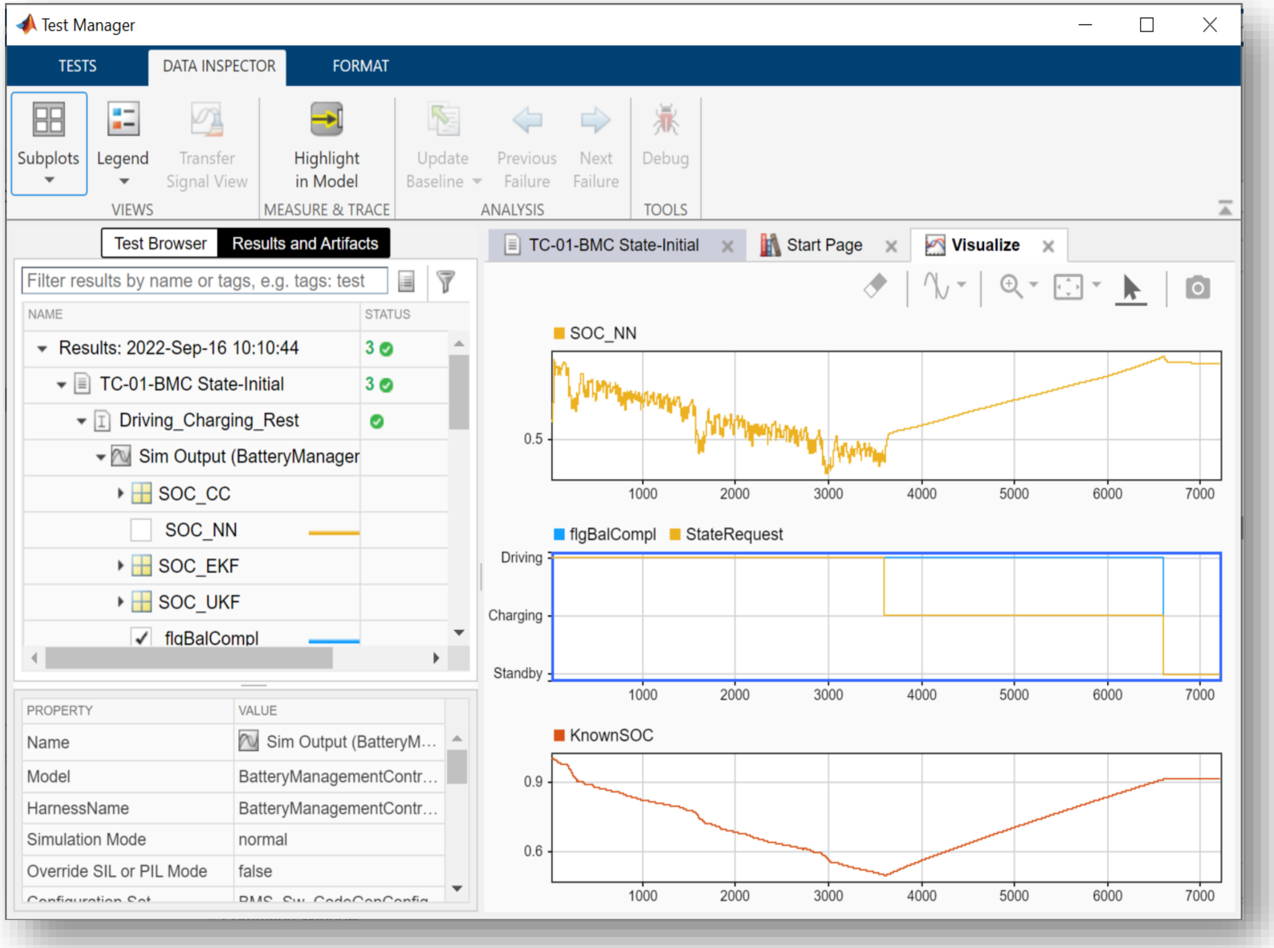


Generated codes

Code Generation

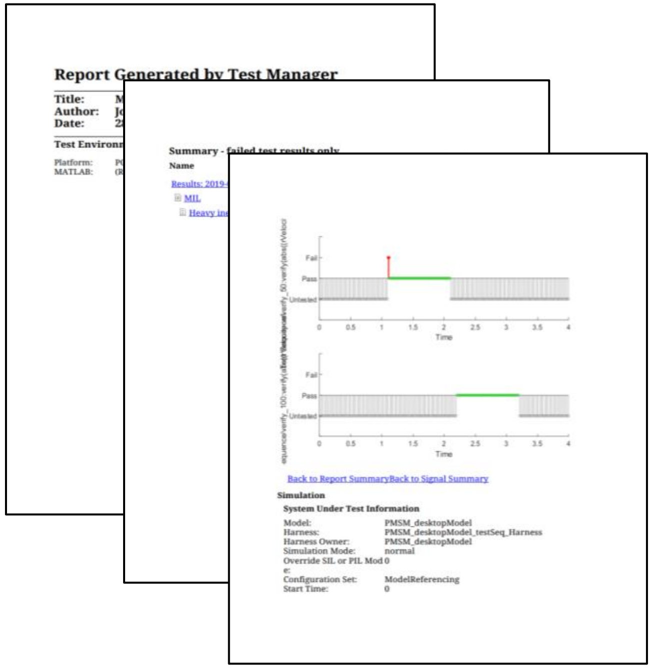
Configure and Simulate AUTOSAR Classic Models with Simulink Real-Time

Test Integrated Software – Hardware-in-the-Loop Testing



NAME	STATUS
Results: 2022-Sep-16 10:10:44	3 ✓
TC-01-BMC State-Initial	3 ✓
Driving_Charging_Rest	✓
Sim Output (BatteryManager	
SOC_CC	
□ SOC_NN	
SOC_EKF	
SOC_UKF	
☑ flgBalCompl	

PROPERTY	VALUE
Name	Sim Output (BatteryM…
Model	BatteryManagementContr…
HarnessName	BatteryManagementContr…
Simulation Mode	normal
Override SIL or PIL Mode	false
Configuration Set	BMS_Sw_CodeGenConfig…



Report Generated by Test Manager

Title: M
Author: Jo
Date: 2

Test Environm

Platform: PC
MATLAB: (R

Summary - failed test results only

Name

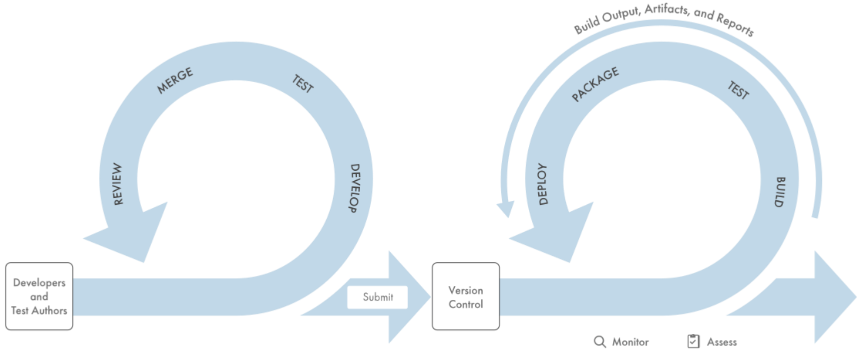
Results: 2019-
MIL
Heavy ine

Back to Report SummaryBack to Signal Summary

Simulation

System Under Test Information

Model:	PMSM_desktopModel
Harness:	PMSM_desktopModel_testSeq_Harness
Harness Owner:	PMSM_desktopModel
Simulation Mode:	normal
Override SIL or PIL Mode:	0
Configuration Set:	ModelReferencing
Start Time:	0

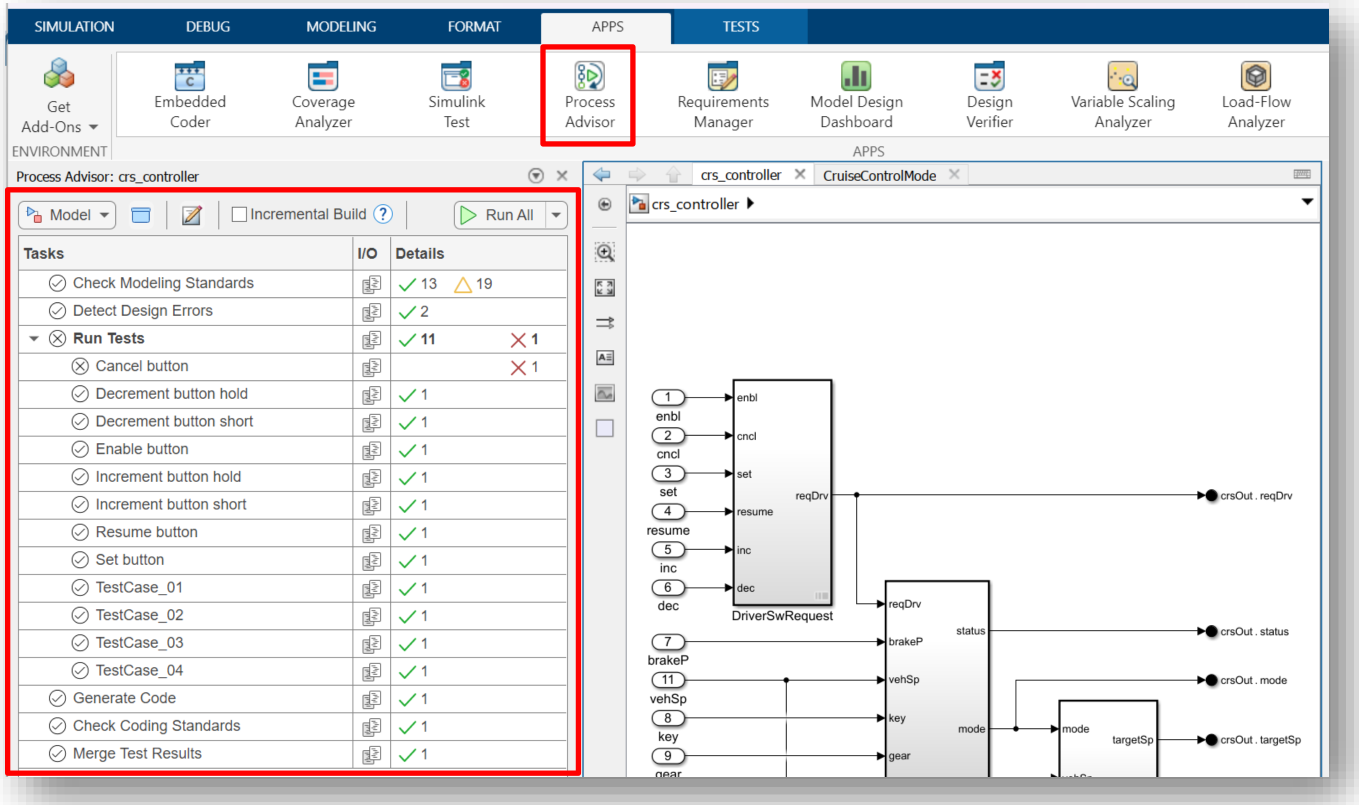


Compatible with any CI platform: Jenkins®, GitHub® Actions, GitLab® CI Pipelines…

Continuous Integration Workflow with Model-Based Design

Process Advisor

- How do I define & deploy an MBD workflow?
- Prequalify locally to reduce build failures
- Reproduce & debug build failures
- Integrate Process into CI Platforms
  → auto generate pipeline configuration file



Process Advisor: crs_controller

Model ▾ | ☐ Incremental Build ? | ▷ Run All ▾

Tasks	I/O	Details
Check Modeling Standards		✓ 13 △ 19
Detect Design Errors		✓ 2
▾ **Run Tests**		✓ 11 ✗ 1
Cancel button		✗ 1
Decrement button hold		✓ 1
Decrement button short		✓ 1
Enable button		✓ 1
Increment button hold		✓ 1
Increment button short		✓ 1
Resume button		✓ 1
Set button		✓ 1
TestCase_01		✓ 1
TestCase_02		✓ 1
TestCase_03		✓ 1
TestCase_04		✓ 1
Generate Code		✓ 1
Check Coding Standards		✓ 1
Merge Test Results		✓ 1

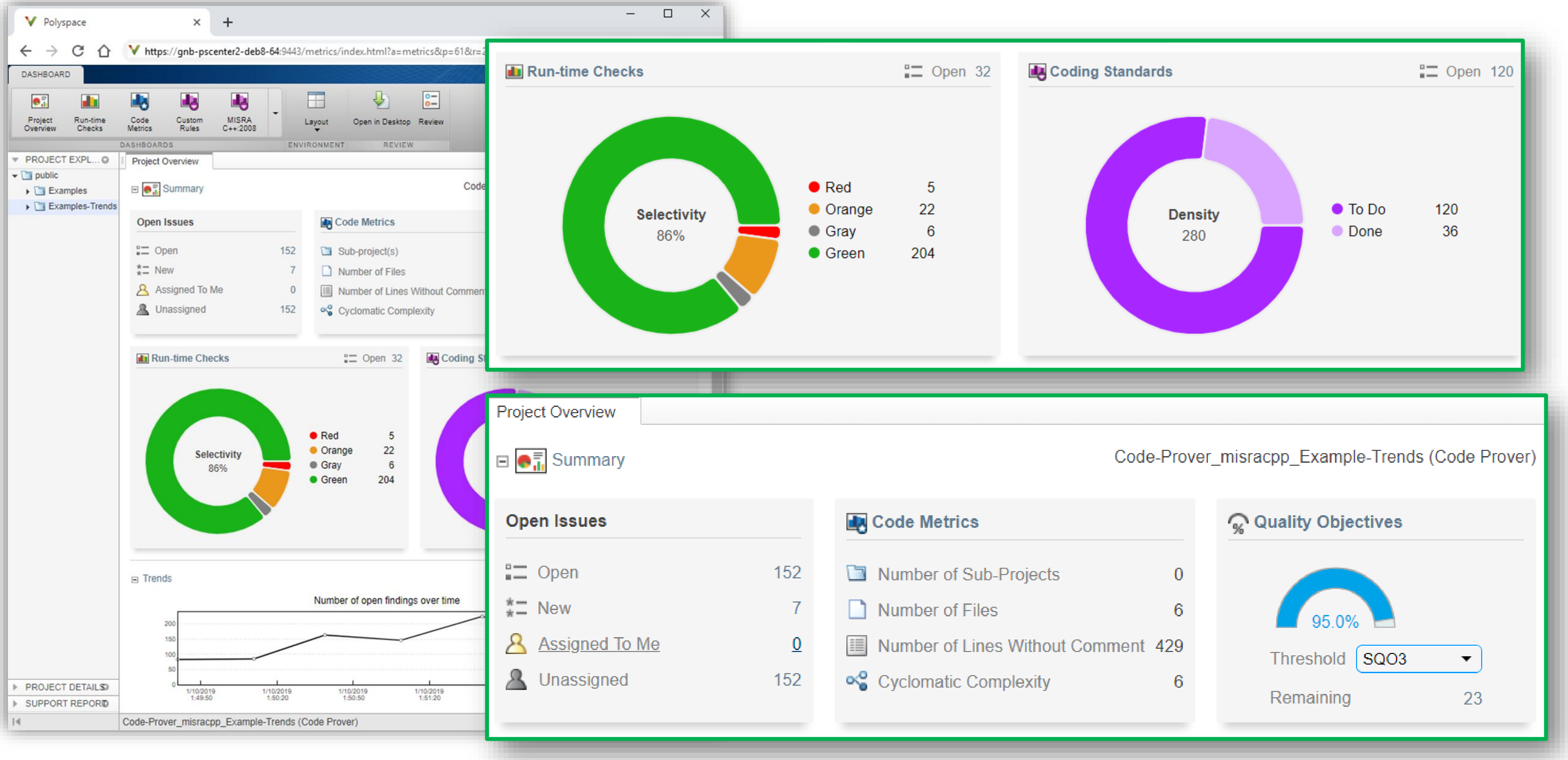
✓ Graphical Front-End to Model-Based Design Build System

✓ Interactive Workflows

✓ Rapid Iteration

Process Advisor: CI/CD Automation for Simulink Check

Perform Software Integration Test with Polyspace



Run-time Checks Open 32

Selectivity 86%

● Red	5
● Orange	22
● Gray	6
● Green	204

Coding Standards Open 120

Density 280

● To Do	120
● Done	36

Project Overview

Summary — Code-Prover_misracpp_Example-Trends (Code Prover)

Open Issues

Open	152
New	7
Assigned To Me	0
Unassigned	152

Code Metrics

Number of Sub-Projects	0
Number of Files	6
Number of Lines Without Comment	429
Cyclomatic Complexity	6

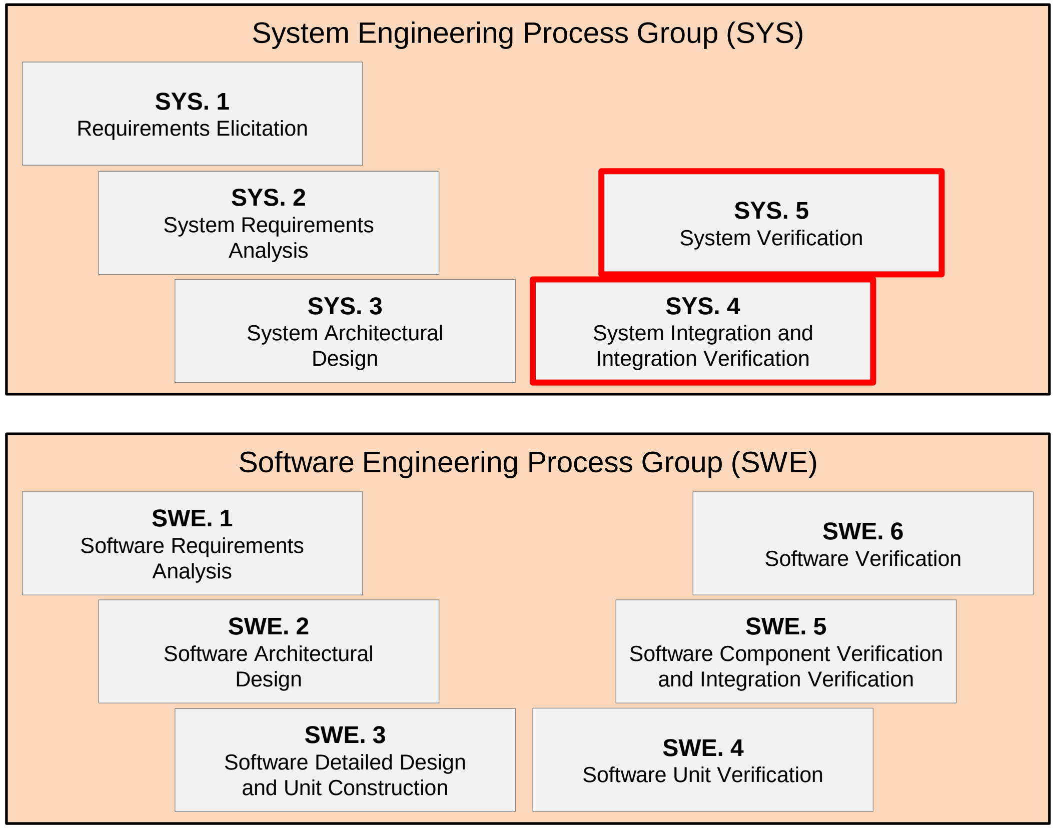
Quality Objectives

95.0%

Threshold: SQO3

Remaining: 23

System Verification



System Engineering Process Group (SYS)

SYS. 1
Requirements Elicitation

SYS. 2
System Requirements Analysis

SYS. 3
System Architectural Design

SYS. 4
System Integration and Integration Verification

SYS. 5
System Verification

Software Engineering Process Group (SWE)

SWE. 1
Software Requirements Analysis

SWE. 2
Software Architectural Design

SWE. 3
Software Detailed Design and Unit Construction

SWE. 4
Software Unit Verification

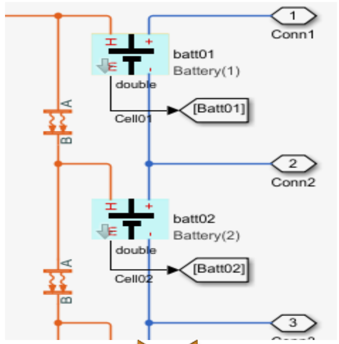
SWE. 5
Software Component Verification and Integration Verification

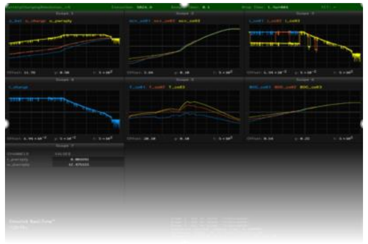
SWE. 6
Software Verification

Hardware-In-Loop Testing of Battery Management System

Testing BMS with Emulated Battery Cells

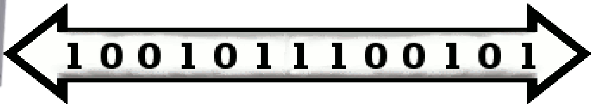
- Reduce testing time
- Test fault conditions safely
- Automate testing





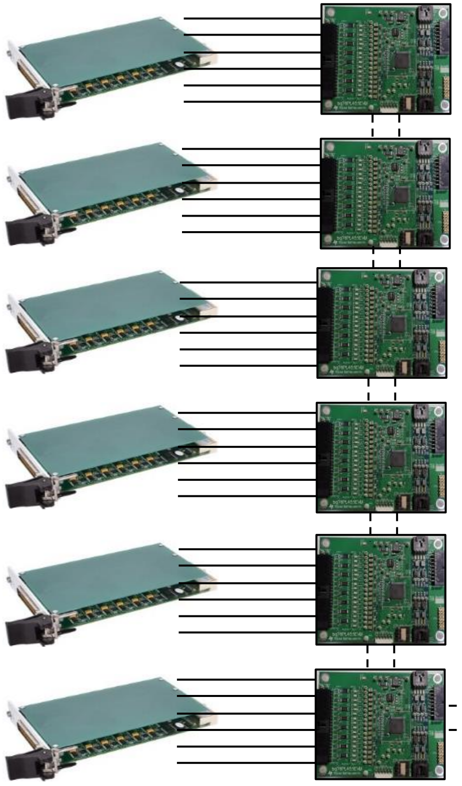
Automatic Code Generation

Battery Emulation



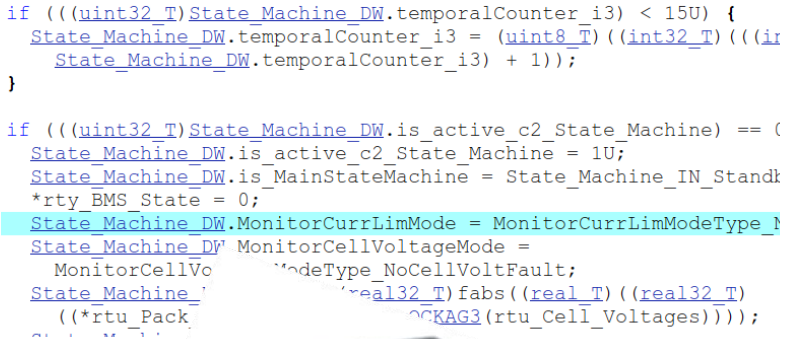
Wiring and Signal Conditioning

Cell Monitoring Software

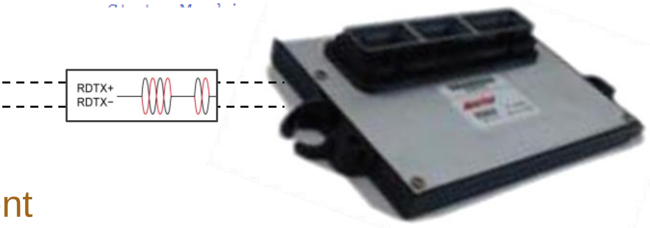


Measurement
Cell Diagnostic,
Cell Balancing

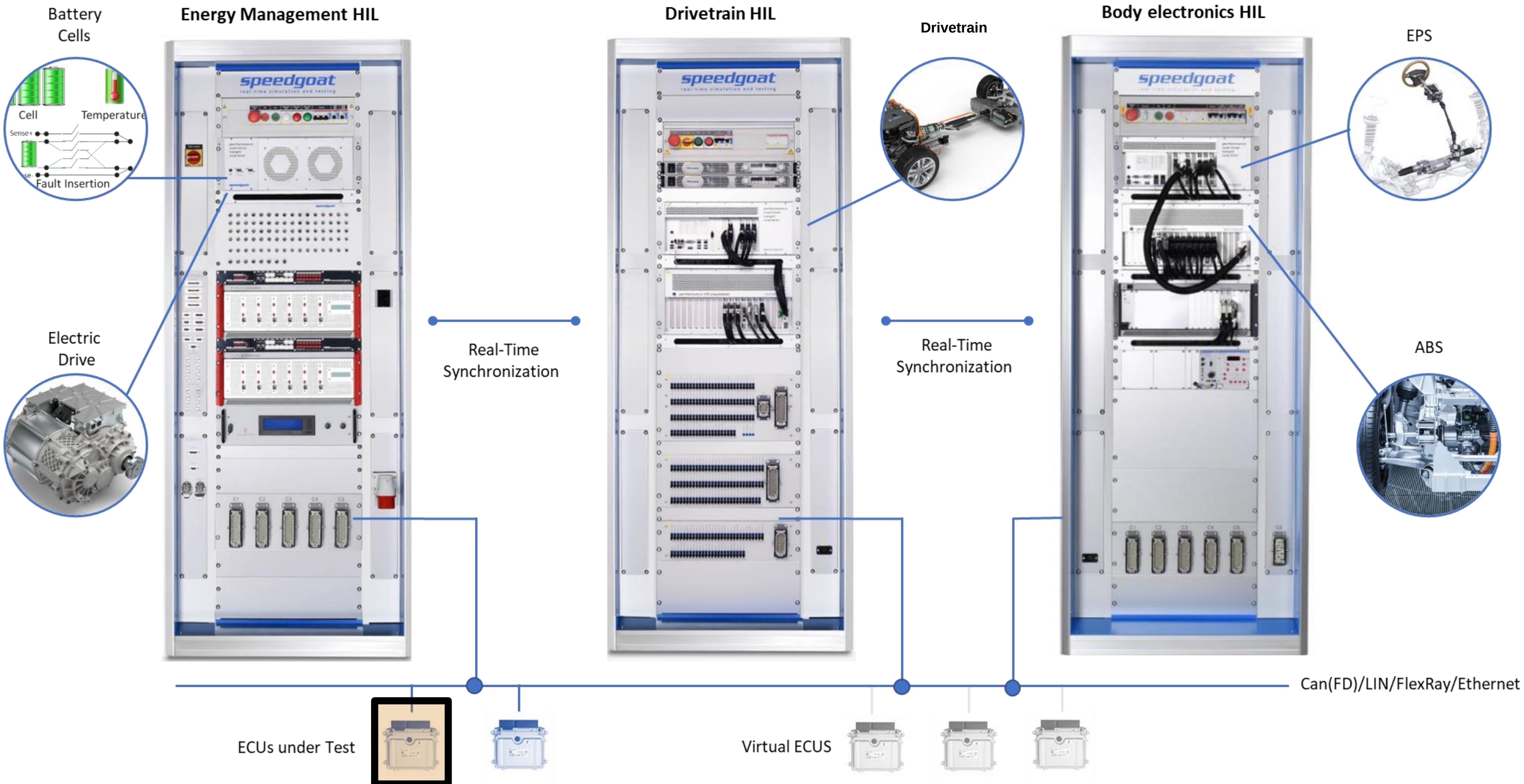
Supervisory tasks
SOC estimation
Contactor management
Isolation monitoring
Fault detection and recovery
Thermal management
Current & power limits



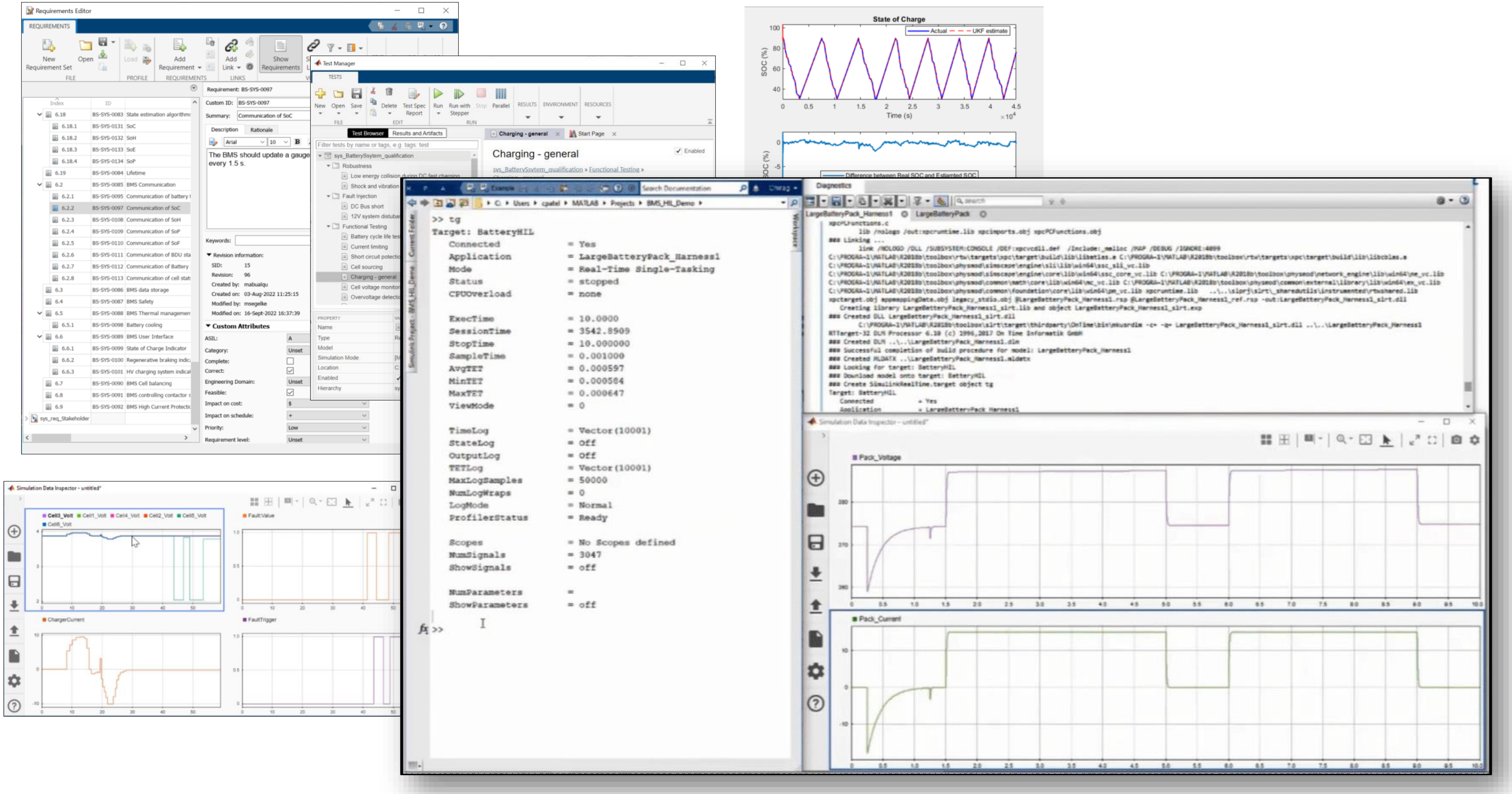
```
if (((uint32_T)State_Machine_DW.temporalCounter_i3) < 15U) {
  State_Machine_DW.temporalCounter_i3 = (uint8_T)((int32_T)(((ir
    State_Machine_DW.temporalCounter_i3) + 1));
}

if (((uint32_T)State_Machine_DW.is_active_c2_State_Machine) == (
  State_Machine_DW.is_active_c2_State_Machine = 1U;
  State_Machine_DW.is_MainStateMachine = State_Machine_IN_Standk
  *rty_BMS_State = 0;
  State_Machine_DW.MonitorCurrLimMode = MonitorCurrLimModeType_N
  State_Machine_DW MonitorCellVoltageMode =
    MonitorCellVo      ModeType_NoCellVoltFault;
  State_Machine_      real32_T)fabs((real_T)((real32_T)
    ((*rtu_Pack                     CKAG3(rtu_Cell_Voltages))));
```



System Qualification Test in HIL



System Qualification Test in HIL



Index	ID	
6.18	BS-SYS-0083	State estimation algorithms
6.18.1	BS-SYS-0131	SoC
6.18.2	BS-SYS-0132	SoH
6.18.3	BS-SYS-0133	SoE
6.18.4	BS-SYS-0134	SoP
6.19	BS-SYS-0084	Lifetime
6.2	BS-SYS-0085	BMS Communication
6.2.1	BS-SYS-0095	Communication of battery t
6.2.2	BS-SYS-0097	Communication of SoC
6.2.3	BS-SYS-0108	Communication of SoH
6.2.4	BS-SYS-0109	Communication of SoP
6.2.5	BS-SYS-0110	Communication of SoF
6.2.6	BS-SYS-0111	Communication of BDU sta
6.2.7	BS-SYS-0112	Communication of Battery
6.2.8	BS-SYS-0113	Communication of cell stat
6.3	BS-SYS-0086	BMS data storage
6.4	BS-SYS-0087	BMS Safety
6.5	BS-SYS-0088	BMS Thermal managemen
6.5.1	BS-SYS-0098	Battery cooling
6.6	BS-SYS-0089	BMS User Interface
6.6.1	BS-SYS-0099	State of Charge Indicator
6.6.2	BS-SYS-0100	Regenerative braking indic
6.6.3	BS-SYS-0101	HV charging system indicat
6.7	BS-SYS-0090	BMS Cell balancing
6.8	BS-SYS-0091	BMS controlling contactor s
6.9	BS-SYS-0092	BMS High Current Protectio
sys_req_Stakeholder		

Requirement: BS-SYS-0097

Custom ID: BS-SYS-0097

Summary: Communication of SoC

Description | Rationale

The BMS should update a gauge
every 1.5 s.

Keywords:

Revision information:

SID: 15
Revision: 96
Created by: mabualqu
Created on: 03-Aug-2022 11:25:15
Modified by: msegelke
Modified on: 16-Sept-2022 16:37:39

Custom Attributes

ASIL: A
Category: Unset
Complete: ☐
Correct: ☑
Engineering Domain: Unset
Feasible: ☑
Impact on cost: $
Impact on schedule: +
Priority: Low
Requirement level: Unset

- sys_BatterySsytem_qualification
  - Robustness
    - Low energy collision during DC fast charging
    - Shock and vibration
  - Fault Injection
    - DC Bus short
    - 12V system distubar
  - Functional Testing
    - Battery cycle life tes
    - Current limiting
    - Short circuit potectio
    - Cell sourcing
    - Charging - general
    - Cell voltage monitor
    - Overvoltage detectio

PROPERTY: Name, Type, Model, Simulation Mode, Location, Enabled, Hierarchy

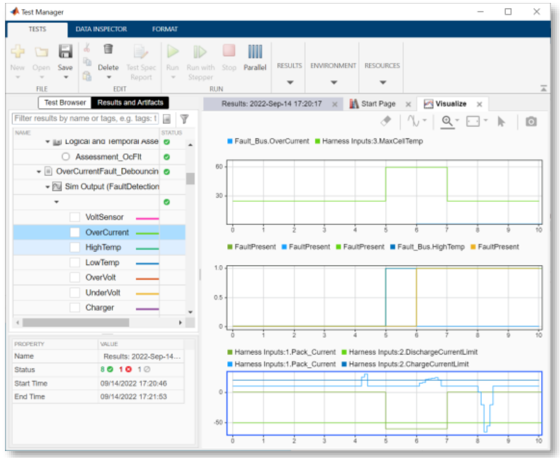
Charging - general ☑ Enabled

sys_BatterySsytem_qualification » Functional Testing »

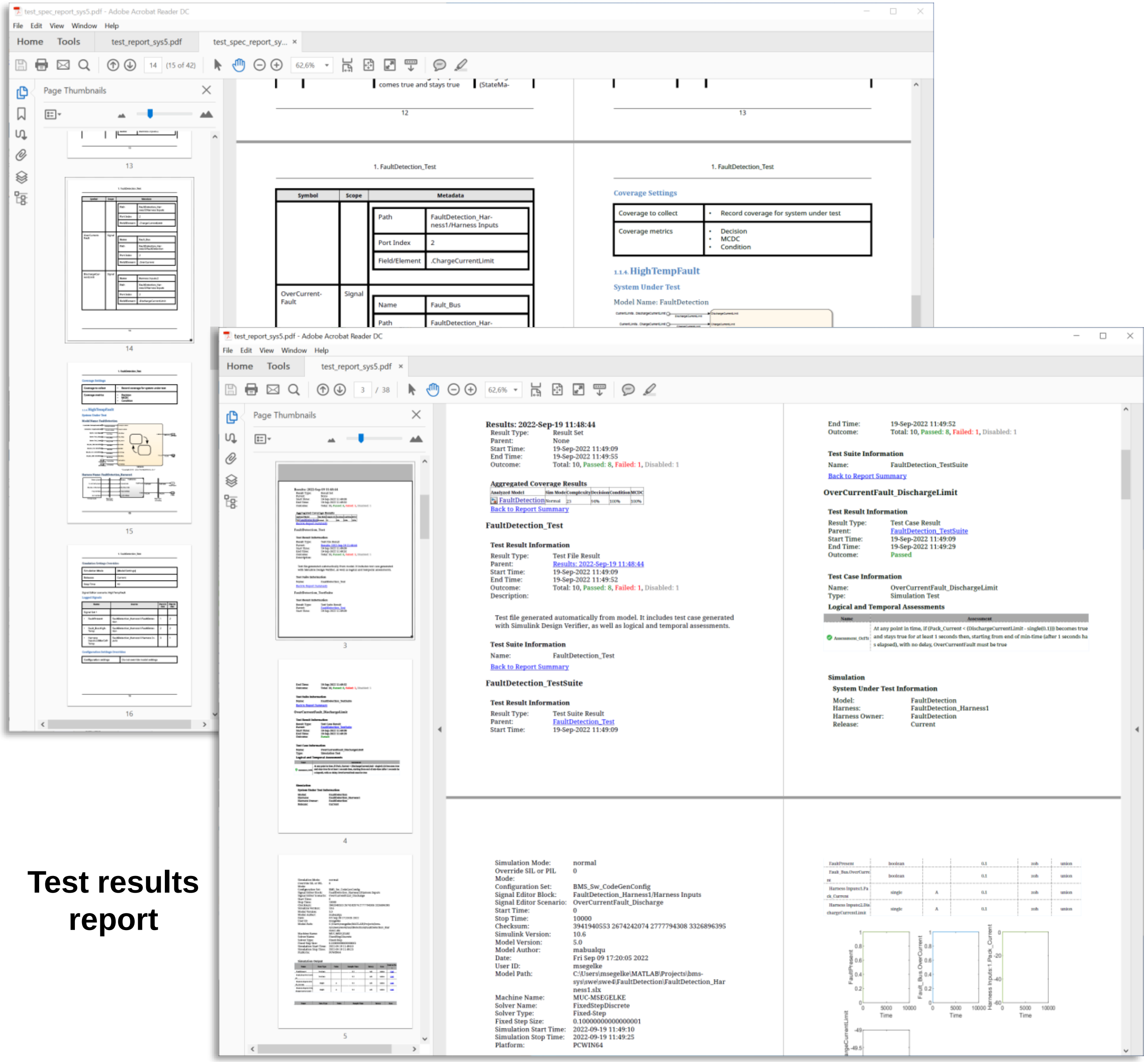
```
>> tg
Target: BatteryHIL
   Connected            = Yes
   Application          = LargeBatteryPack_Harness1
   Mode                 = Real-Time Single-Tasking
   Status               = stopped
   CPUOverload          = none

   ExecTime             = 10.0000
   SessionTime          = 3542.8909
   StopTime             = 10.000000
   SampleTime           = 0.001000
   AvgTET               = 0.000597
   MinTET               = 0.000584
   MaxTET               = 0.000647
   ViewMode             = 0

   TimeLog              = Vector(10001)
   StateLog             = Off
   OutputLog            = Off
   TETLog               = Vector(10001)
   MaxLogSamples        = 50000
   NumLogWraps          = 0
   LogMode              = Normal
   ProfilerStatus       = Ready

   Scopes               = No Scopes defined
   NumSignals           = 3047
   ShowSignals          = off

   NumParameters        =
   ShowParameters       = off

fx >>
```

```
xpcPCFunctions.c
        lib /nologo /out:xpcruntime.lib xpcimports.obj xpcPCFunctions.obj
### Linking ...
        link /NOLOGO /DLL /SUBSYSTEM:CONSOLE /DEF:xpcvcdll.def  /Include:_malloc /MAP /DEBUG /IGNORE:4099
C:\PROGRA~1\MATLAB\R2018b\toolbox\rtw\targets\xpc\target\build\lib\libatlas.a C:\PROGRA~1\MATLAB\R2018b\toolbox\rtw\targets\xpc\target\build\lib\libcblas.a
C:\PROGRA~1\MATLAB\R2018b\toolbox\physmod\simscape\engine\sli\lib\win64\ssc_sli_vc.lib
C:\PROGRA~1\MATLAB\R2018b\toolbox\physmod\simscape\engine\core\lib\win64\ssc_core_vc.lib C:\PROGRA~1\MATLAB\R2018b\toolbox\physmod\network_engine\lib\win64\ne_vc.lib
C:\PROGRA~1\MATLAB\R2018b\toolbox\physmod\common\math\core\lib\win64\mc_vc.lib C:\PROGRA~1\MATLAB\R2018b\toolbox\physmod\common\external\library\lib\win64\ex_vc.lib
C:\PROGRA~1\MATLAB\R2018b\toolbox\physmod\common\foundation\core\lib\win64\pm_vc.lib xpcruntime.lib   ..\..\slprj\slrt\_sharedutils\instrumented\rtwshared.lib
xpctarget.obj appmappingData.obj legacy_stdio.obj @LargeBatteryPack_Harness1.rsp @LargeBatteryPack_Harness1_ref.rsp -out:LargeBatteryPack_Harness1_slrt.dll
   Creating library LargeBatteryPack_Harness1_slrt.lib and object LargeBatteryPack_Harness1_slrt.exp
### Created DLL LargeBatteryPack_Harness1_slrt.dll
        C:\PROGRA~1\MATLAB\R2018b\toolbox\slrt\target\thirdparty\OnTime\bin\mkusrdlm -c+ -q+ LargeBatteryPack_Harness1_slrt.dll ..\..\LargeBatteryPack_Harness1
RTTarget-32 DLM Processor 6.10 (c) 1996,2017 On Time Informatik GmbH
### Created DLM ..\..\LargeBatteryPack_Harness1.dlm
### Successful completion of build procedure for model: LargeBatteryPack_Harness1
### Created MLDATX ..\LargeBatteryPack_Harness1.mldatx
### Looking for target: BatteryHIL
### Download model onto target: BatteryHIL
### Create SimulinkRealTime.target object tg
Target: BatteryHIL
   Connected            = Yes
   Application          = LargeBatteryPack_Harness1
```

Report Generation



Test specification report



comes true and stays true

(StateMa-

12

13

1. FaultDetection_Test

Symbol	Scope	Metadata
		Path: FaultDetection_Harness1/Harness Inputs Port Index: 2 Field/Element: .ChargeCurrentLimit
OverCurrent-Fault	Signal	Name: Fault_Bus Path: FaultDetection_Har-

1. FaultDetection_Test

Coverage Settings

Coverage to collect	• Record coverage for system under test
Coverage metrics	• Decision • MCDC • Condition

1.1.4. HighTempFault

System Under Test

Model Name: FaultDetection

Results: 2022-Sep-19 11:48:44

Result Type: Result Set
Parent: None
Start Time: 19-Sep-2022 11:49:09
End Time: 19-Sep-2022 11:49:55
Outcome: Total: 10, Passed: 8, Failed: 1, Disabled: 1

Aggregated Coverage Results

Analyzed Model	Sim Mode	Complexity	Decision	Condition	MCDC
FaultDetection	Normal	23	94%	100%	100%

Back to Report Summary

FaultDetection_Test

Test Result Information

Result Type: Test File Result
Parent: Results: 2022-Sep-19 11:48:44
Start Time: 19-Sep-2022 11:49:09
End Time: 19-Sep-2022 11:49:52
Outcome: Total: 10, Passed: 8, Failed: 1, Disabled: 1
Description:

Test file generated automatically from model. It includes test case generated with Simulink Design Verifier, as well as logical and temporal assessments.

Test Suite Information

Name: FaultDetection_Test

Back to Report Summary

FaultDetection_TestSuite

Test Result Information

Result Type: Test Suite Result
Parent: FaultDetection_Test
Start Time: 19-Sep-2022 11:49:09
End Time: 19-Sep-2022 11:49:52
Outcome: Total: 10, Passed: 8, Failed: 1, Disabled: 1

Test Suite Information

Name: FaultDetection_TestSuite

Back to Report Summary

OverCurrentFault_DischargeLimit

Test Result Information

Result Type: Test Case Result
Parent: FaultDetection_TestSuite
Start Time: 19-Sep-2022 11:49:09
End Time: 19-Sep-2022 11:49:29
Outcome: Passed

Test Case Information

Name: OverCurrentFault_DischargeLimit
Type: Simulation Test

Logical and Temporal Assessments

Name	Assessment
✓ Assessment_OcFlt	At any point in time, if (Pack_Current < (DischargeCurrentLimit - single(0.1))) becomes true and stays true for at least 1 seconds then, starting from end of min-time (after 1 seconds ha s elapsed), with no delay, OverCurrentFault must be true

Simulation

System Under Test Information

Model: FaultDetection
Harness: FaultDetection_Harness1
Harness Owner: FaultDetection
Release: Current

Simulation Mode: normal
Override SIL or PIL Mode: 0
Configuration Set: BMS_Sw_CodeGenConfig
Signal Editor Block: FaultDetection_Harness1/Harness Inputs
Signal Editor Scenario: OverCurrentFault_Discharge
Start Time: 0
Stop Time: 10000
Checksum: 3941940553 2674242074 2777794308 3326896395
Simulink Version: 10.6
Model Version: 5.0
Model Author: mabualqu
Date: Fri Sep 09 17:20:05 2022
User ID: msegelke
Model Path: C:\Users\msegelke\MATLAB\Projects\bms-sys\swe\swe4\FaultDetection\FaultDetection_Harness1.slx
Machine Name: MUC-MSEGELKE
Solver Name: FixedStepDiscrete
Solver Type: Fixed-Step
Fixed Step Size: 0.10000000000000001
Simulation Start Time: 2022-09-19 11:49:10
Simulation Stop Time: 2022-09-19 11:49:25
Platform: PCWIN64

Test results report

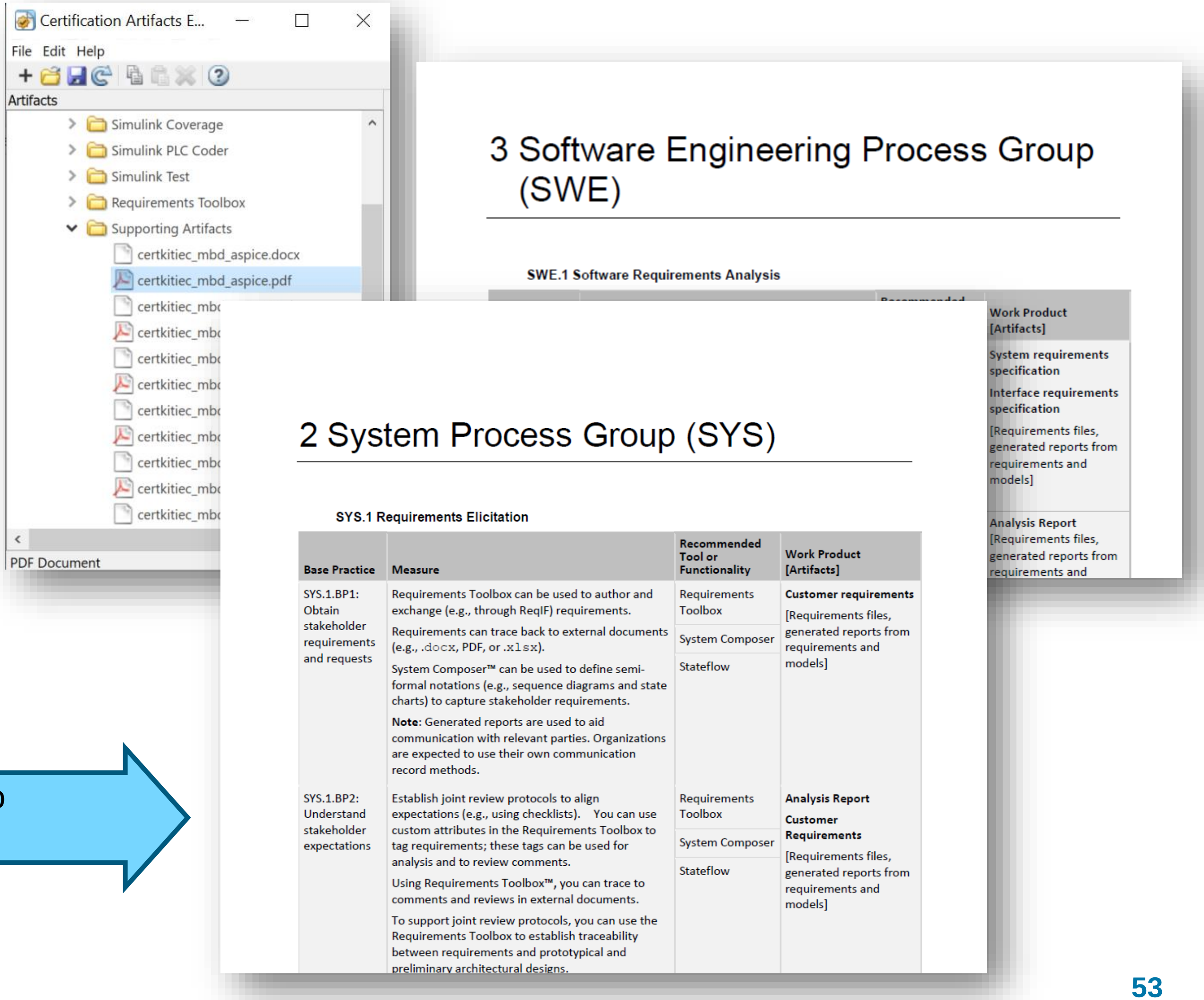
Referencde MBD Process for A-SPICE®

IEC Certification Kit

4.3.2. SYS.2 System Requirements Analysis

SYS.2 System Requirements Analysis	Outcome 1	Outcome 2	Outcome 3	Outcome 4	Outcome 5	Outcome 6
Output Information Items						
17-00 Requirement	X	X				
17-54 Requirement Attribute		X	X			
15-51 Analysis Results			X	X		
13-51 Consistency Evidence					X	
13-52 Communication Evidence						X
Base Practices						
BP1: Specify system requirements	X					
BP2: Structure system requirements		X				
BP3: Analyze system requirements			X			
BP4: Analyze the impact on the system context				X		
BP5: Ensure consistency and establish bidirectional traceability					X	
BP6: Communicate agreed system requirements and impact on the system context						X

Mapping A-SPICE process to MathWorks products



Simulink Coverage
Simulink PLC Coder
Simulink Test
Requirements Toolbox
Supporting Artifacts
certkitiec_mbd_aspice.docx
certkitiec_mbd_aspice.pdf
certkitiec_mbc
certkitiec_mbc
certkitiec_mbc
certkitiec_mbc
certkitiec_mbc
certkitiec_mbc
certkitiec_mbc
certkitiec_mbc
certkitiec_mbc

3 Software Engineering Process Group (SWE)

SWE.1 Software Requirements Analysis

Work Product [Artifacts]
System requirements specification **Interface requirements specification** [Requirements files, generated reports from requirements and models]
Analysis Report [Requirements files, generated reports from requirements and

2 System Process Group (SYS)

SYS.1 Requirements Elicitation

Base Practice	Measure	Recommended Tool or Functionality	Work Product [Artifacts]
SYS.1.BP1: Obtain stakeholder requirements and requests	Requirements Toolbox can be used to author and exchange (e.g., through ReqIF) requirements. Requirements can trace back to external documents (e.g., `.docx`, PDF, or `.xlsx`). System Composer™ can be used to define semi-formal notations (e.g., sequence diagrams and state charts) to capture stakeholder requirements. **Note**: Generated reports are used to aid communication with relevant parties. Organizations are expected to use their own communication record methods.	Requirements Toolbox System Composer Stateflow	**Customer requirements** [Requirements files, generated reports from requirements and models]
SYS.1.BP2: Understand stakeholder expectations	Establish joint review protocols to align expectations (e.g., using checklists). You can use custom attributes in the Requirements Toolbox to tag requirements; these tags can be used for analysis and to review comments. Using Requirements Toolbox™, you can trace to comments and reviews in external documents. To support joint review protocols, you can use the Requirements Toolbox to establish traceability between requirements and prototypical and preliminary architectural designs.	Requirements Toolbox System Composer Stateflow	**Analysis Report** **Customer Requirements** [Requirements files, generated reports from requirements and models]

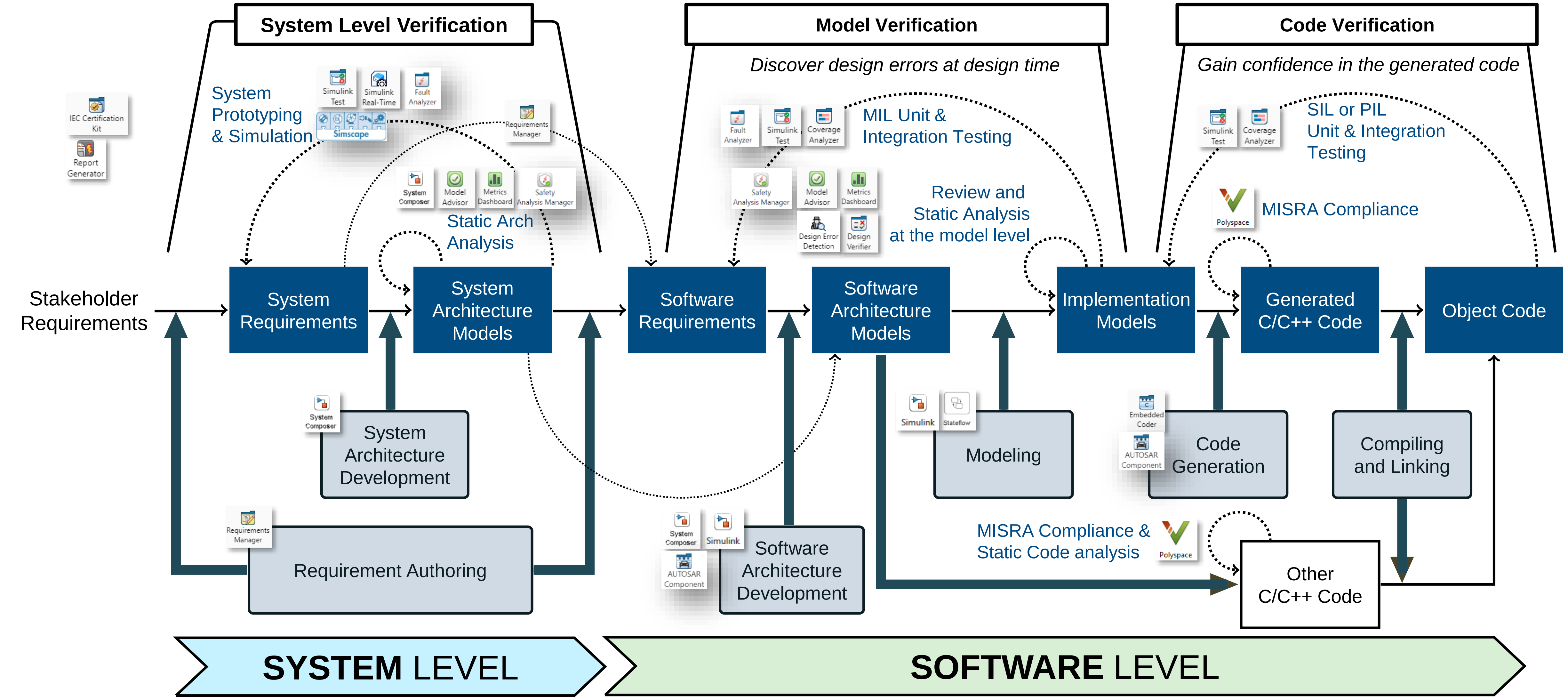
Key Takeaways

Model-Based Design and Model-Based Systems Engineering enable:

1. **Fast development and realization** of **system and software** architecture and design
2. **Early testing** to detect errors in designs and their realization
3. **Fast and efficient iterations**

Develop **high quality products** following an **efficient Automotive SPICE® compliant process**

Reference Workflow for A-SPICE® and ISO 26262



IEC Certification Kit

Report Generator

System Level Verification

System Prototyping & Simulation

Simulink Test, Simulink Real-Time, Fault Analyzer, Simscape

Requirements Manager

System Composer, Model Advisor, Metrics Dashboard, Safety Analysis Manager

Static Arch Analysis

Model Verification

Discover design errors at design time

Fault Analyzer, Simulink Test, Coverage Analyzer

MIL Unit & Integration Testing

Safety Analysis Manager, Model Advisor, Metrics Dashboard, Design Error Detection, Design Verifier

Review and Static Analysis at the model level

Code Verification

Gain confidence in the generated code

Simulink Test, Coverage Analyzer

SIL or PIL Unit & Integration Testing

Polyspace

MISRA Compliance

Stakeholder Requirements

System Requirements

System Architecture Models

Software Requirements

Software Architecture Models

Implementation Models

Generated C/C++ Code

Object Code

System Composer

System Architecture Development

Requirements Manager

Requirement Authoring

System Composer, Simulink, AUTOSAR Component

Software Architecture Development

Simulink, Stateflow

Modeling

Embedded Coder, AUTOSAR Component

Code Generation

Compiling and Linking

MISRA Compliance & Static Code analysis

Polyspace

Other C/C++ Code

SYSTEM LEVEL

SOFTWARE LEVEL

MathWorks
AUTOMOTIVE CONFERENCE 2024
Korea

Thank you

MathWorks