

MathWorks®

Streamlining Software development for Speed and Compliance

The Model-based Software Factory



Dr. Tjorben Groß



Stefan David

Today's topics



- Challenging demands for Automotive SW Engineering
- Improve SW factories through augmentation with model-based simulation capabilities
- Key building blocks for your SW delivery process to achieve speed and compliance

Automotive Industry – The Current Status, Goals and Challenges

Brand-distinctive features and main customer value will come from Software

Customer Expectations

- Frequent new features
- Up-to-date functions
- High quality

Regulation Demands

- Timely security/safety updates
- Patches for many years
- Rigorous documentation



Monetization

Cost savings

Software Factories

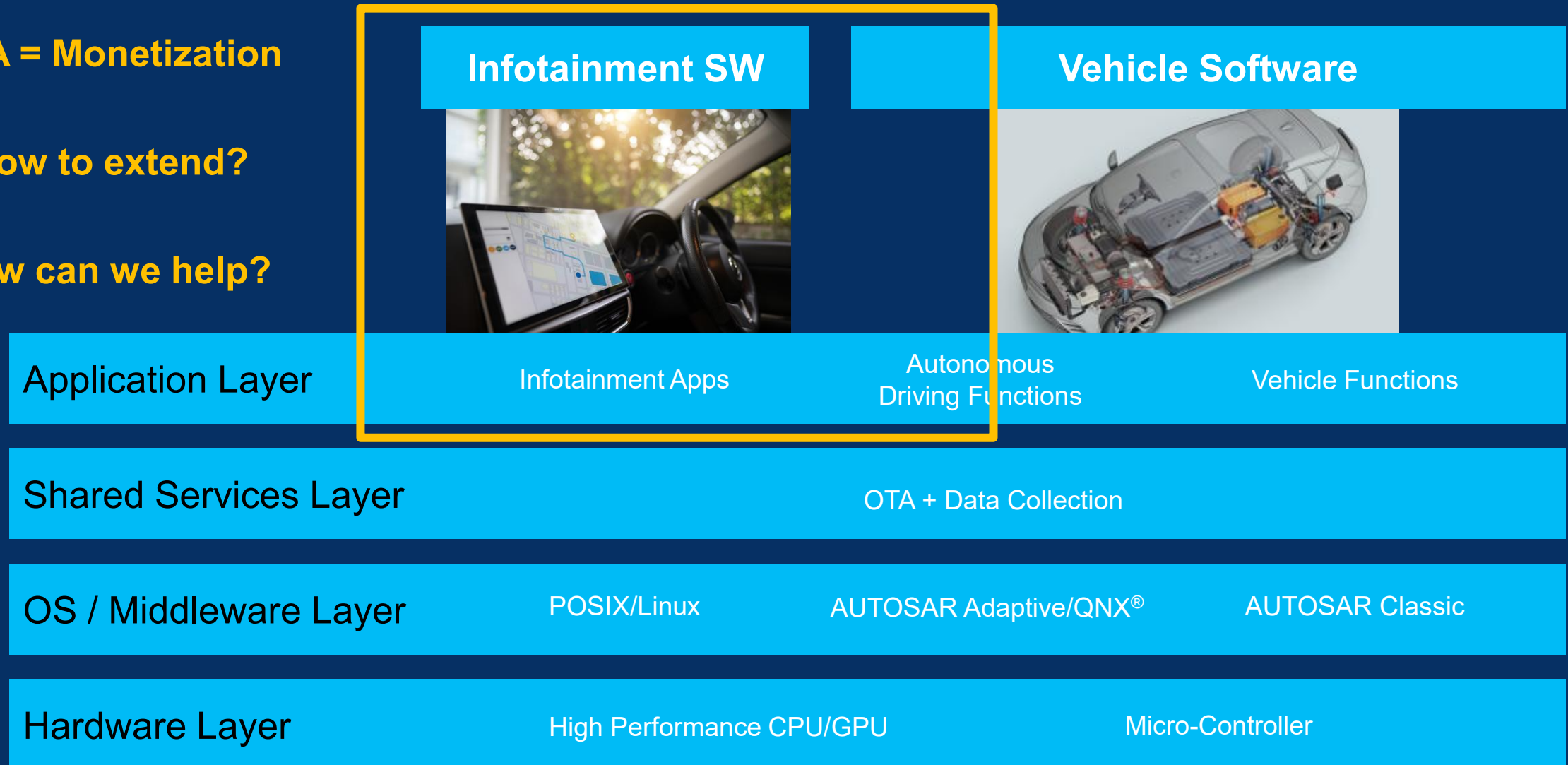
- Streamlined engineering processes
- Development platforms
- Automated SW integration, test and deployments

Automotive Industry – The Current Status, Goals and Challenges

OTA = Monetization

How to extend?

How can we help?



Virtual validation for monetizable vehicle function update



Cloud Instance #1



Full System



Components



Cloud Instance #2



Production Middleware



Virtual Processor

Cloud Instance #3



Scenarios



Virtual validation for monetizable vehicle function update

Automotive Software Development in the Cloud

powered by 

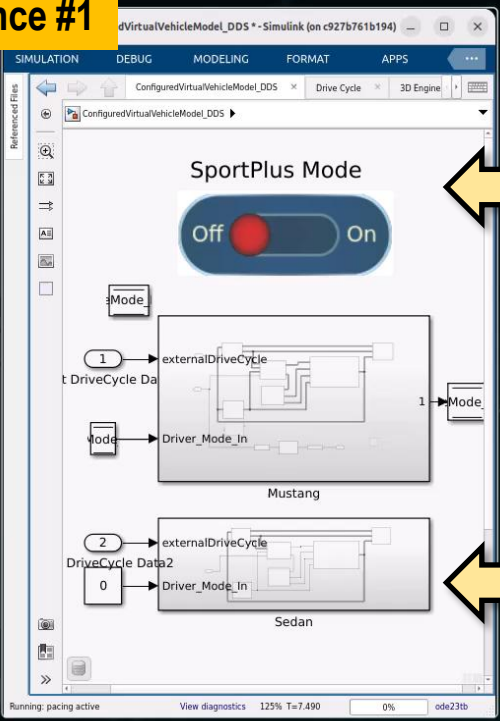
Cloud Instance #1



Full System

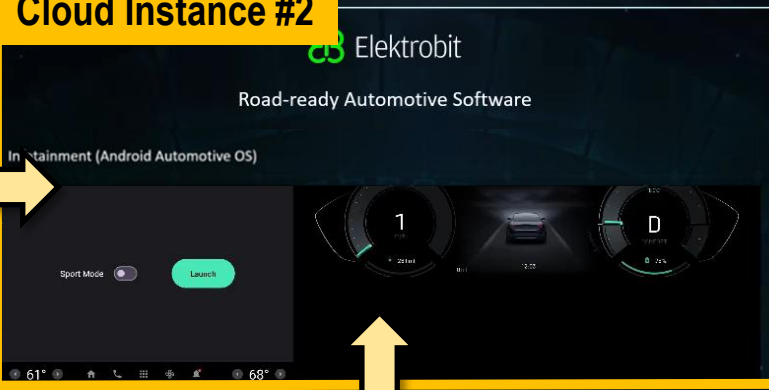


Components

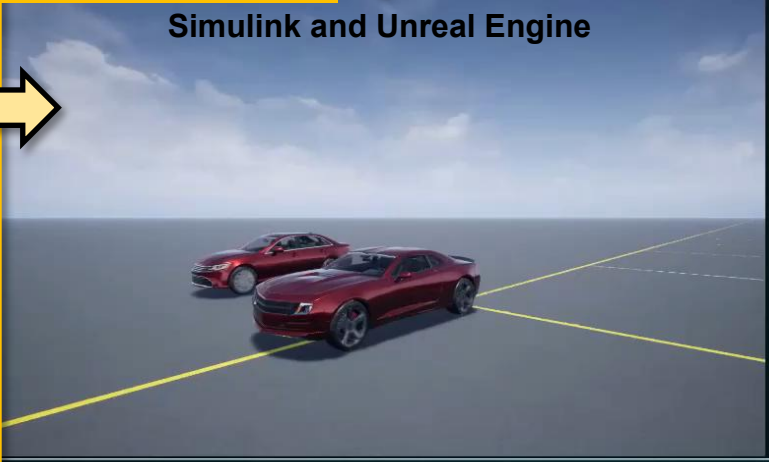


Application Code from Model

Cloud Instance #2



Cloud Instance #3



Scenarios



Production Middleware



Virtual Processor

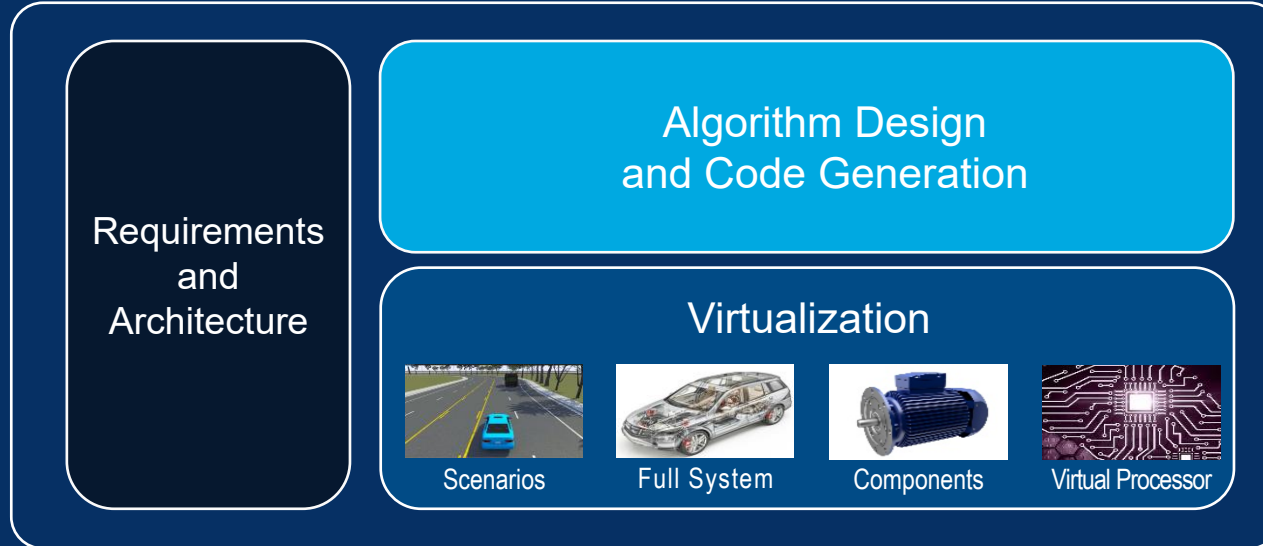


ECU Simulation

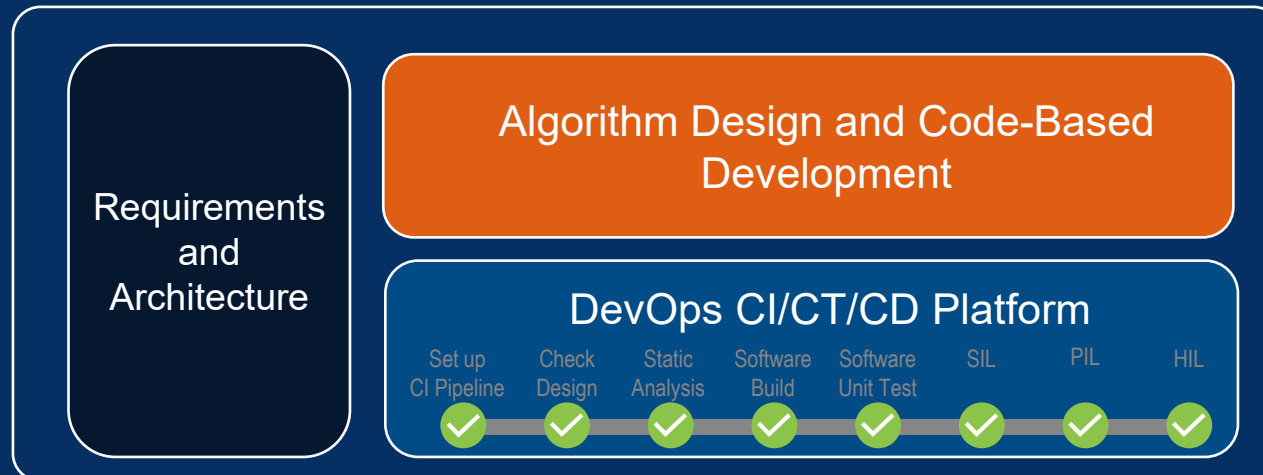
Align and Automate Systems- and Software Engineering

Enable rapid development, verification & validation cycles

VEHICLE ASW
ENGINEERING



INFOTAINMENT + BSW
ENGINEERING



ASW ...Application Software
BSW ...Basic Software



Align and Automate Systems- and Software Engineering

Enable rapid development, verification & validation cycles

VEHICLE ASW
ENGINEERING

Requirements
and
Architecture

Algorithm Design
and Code Generation

Virtualization



Scenarios



Full System



Components

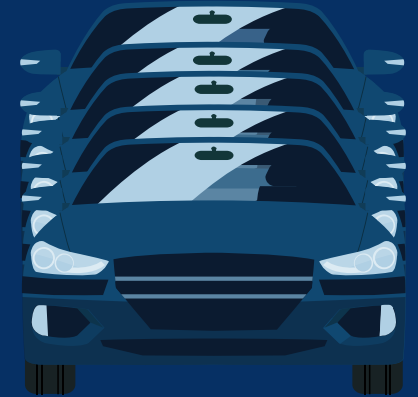
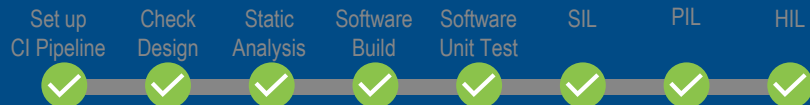


Virtual Processor

Requirements
and
Architecture

Algorithm Design and Code-Based
Development

DevOps CI/CT/CD Platform



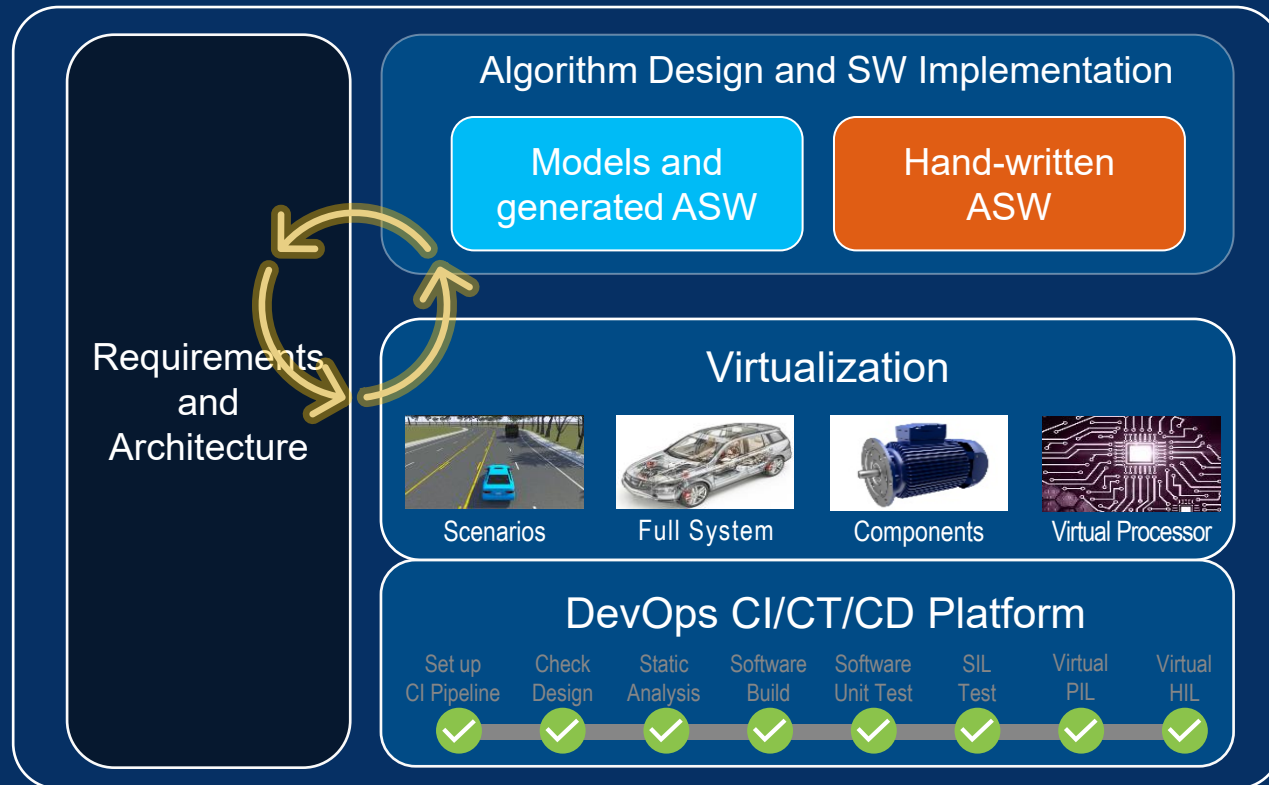
INFOTAINMENT + BSW
ENGINEERING

ASW ...Application Software
BSW ...Basic Software

Align and Automate Systems- and Software Engineering

Enable rapid development, verification & validation cycles

VEHICLE ASW
ENGINEERING



ASW ...Application Software
BSW ...Basic Software

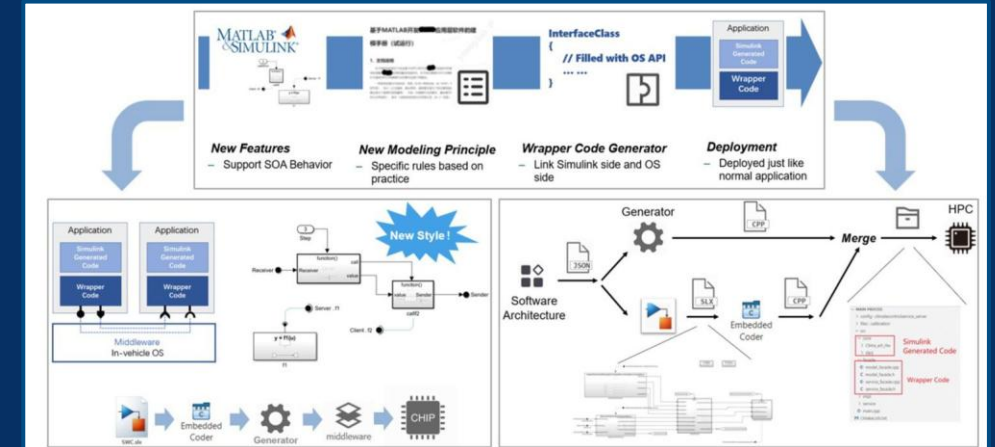


Zeekr Innovates Software-Defined Vehicle Engineering

Zeekr streamlines SDV development with Model-Based Design, integrating agile methods to enhance speed and reduce costs, focusing on simulation and CI/CD for superior software quality.

Key Outcomes

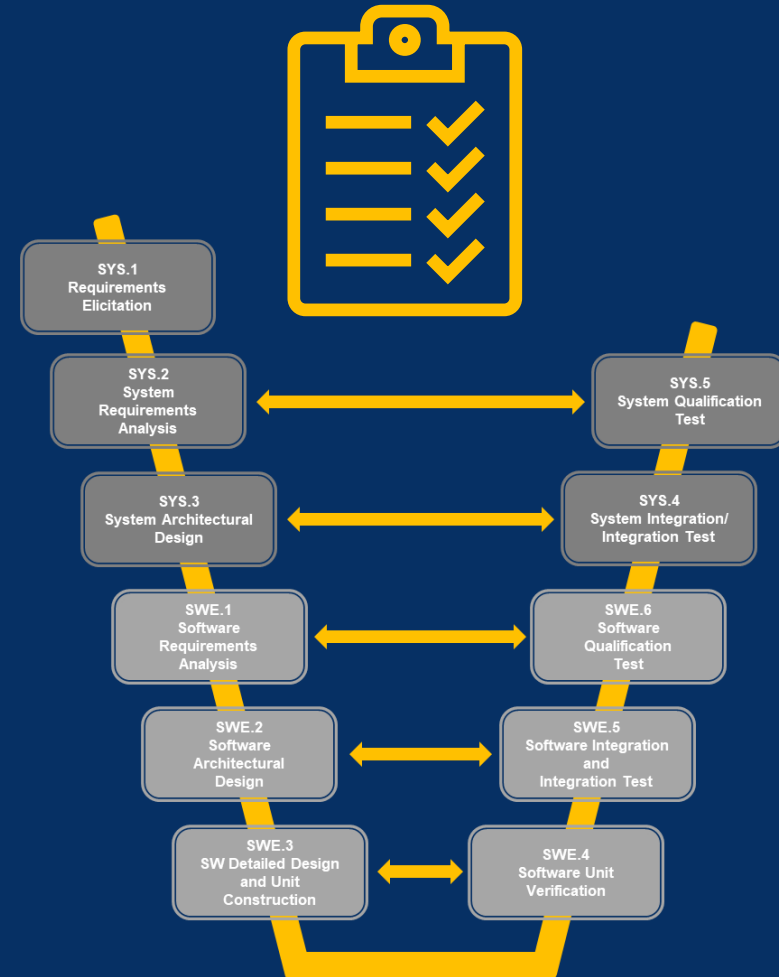
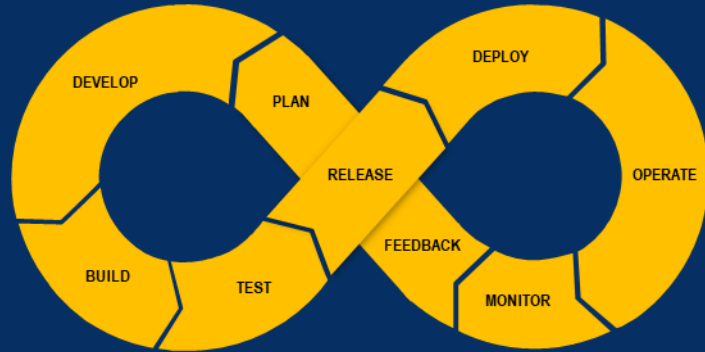
- Model-Based Design helped accelerate software delivery, reducing development time and costs
- Model-Based Design enhanced vehicle software quality and reliability by helping integrate agile and V-model development approaches
- Hybrid approach improved efficiency in software iteration and testing with virtual simulation and CI/CD practices



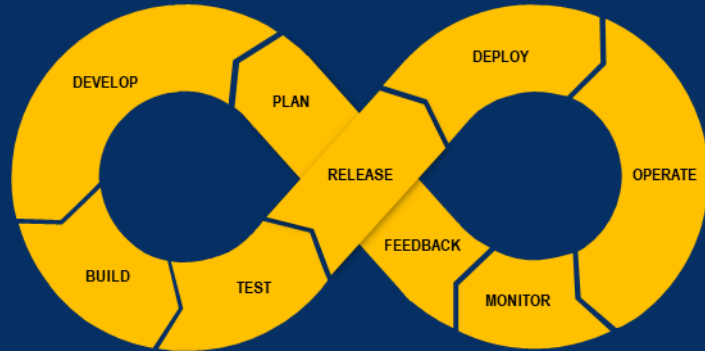
Zeekr is developing and deploying new features for SDVs.

Model-Based Design extends agile principles to system development efforts that include physical components and software. From requirements definition, system architecture, and component design, to implementation, validation, testing, and deployment, Model-Based Design spans the entire development cycle and enables rapid adaptation to changing requirements.

Achieve fast modern SW development **and** Compliance



Achieve fast modern SW development **and** Compliance

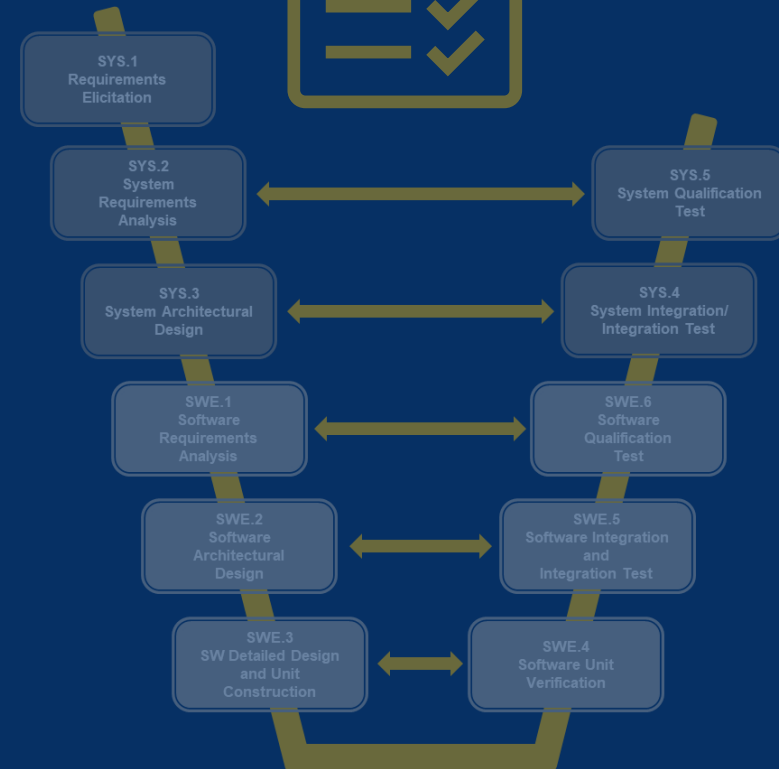


Virtualization / Shift-left

Process Automation

Reuse and integration

Scaling in the cloud

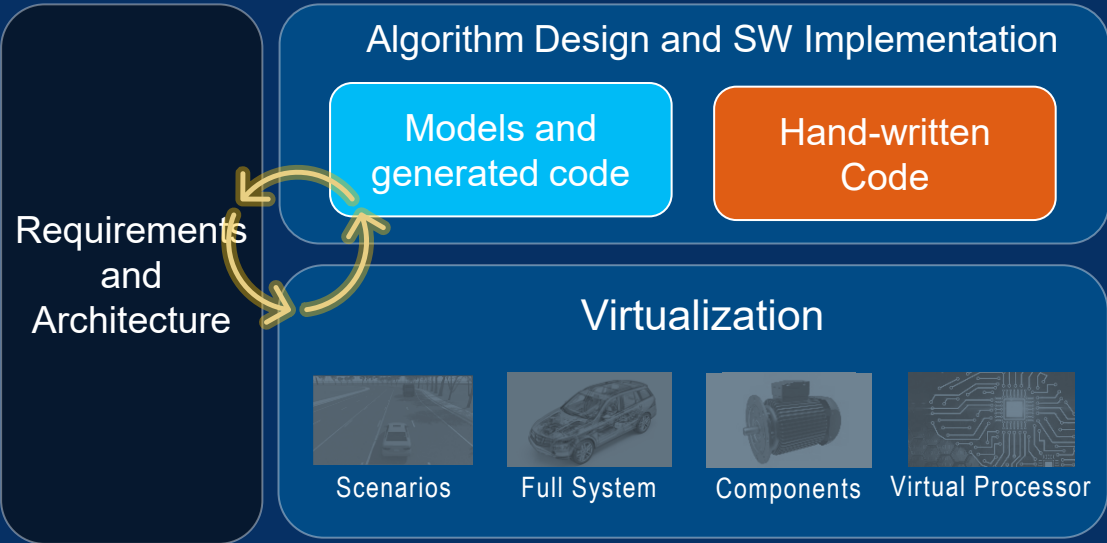


...

Virtualization for early validation

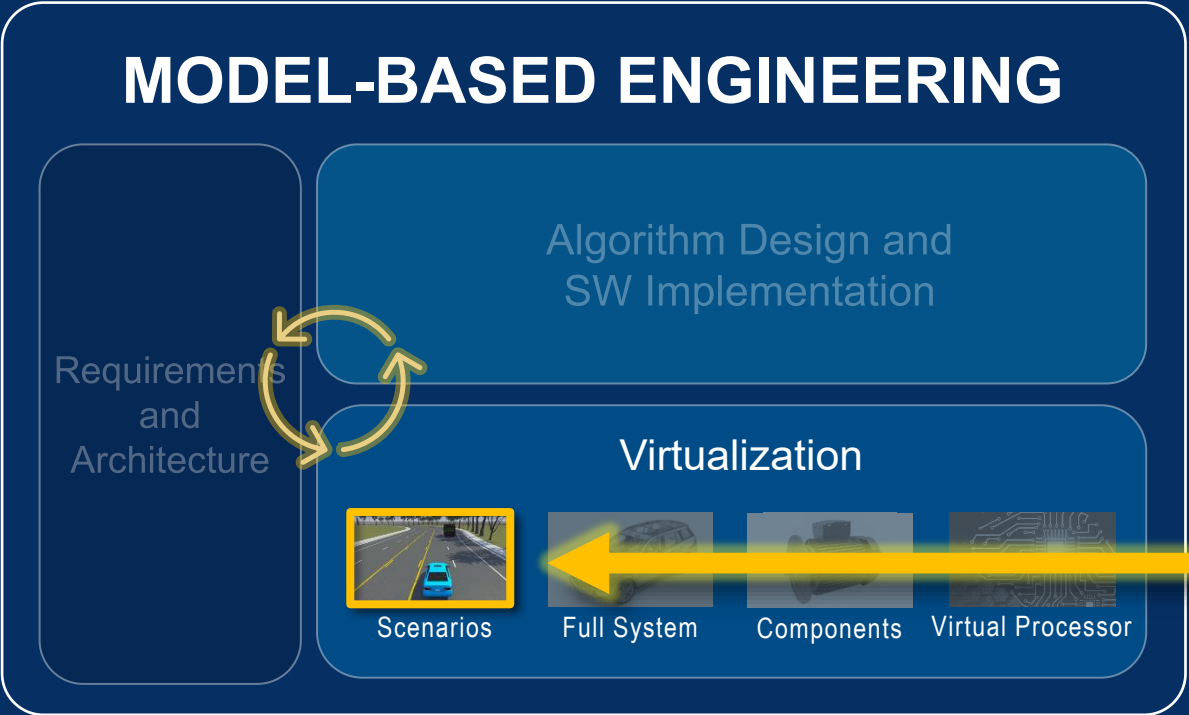


MODEL-BASED ENGINEERING

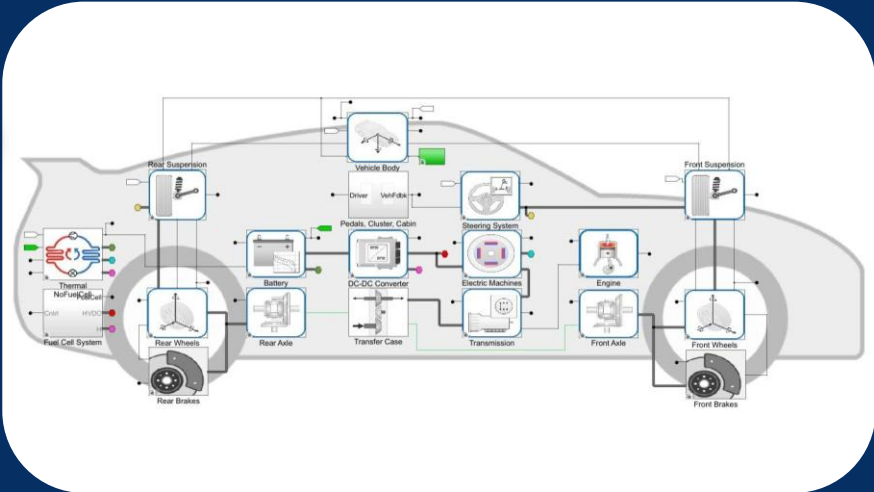
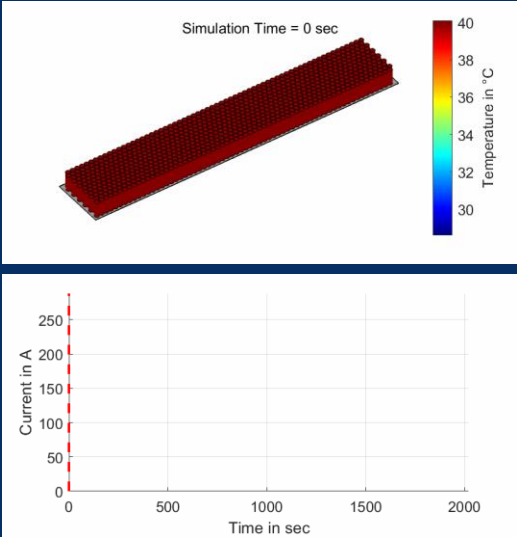
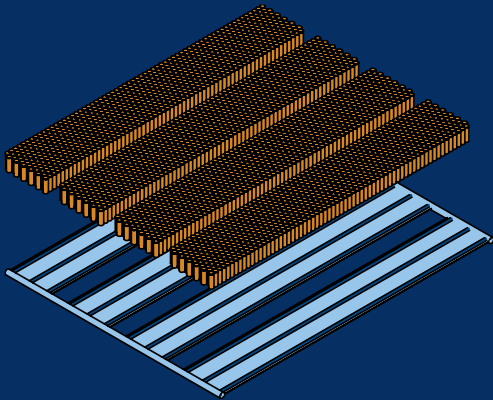
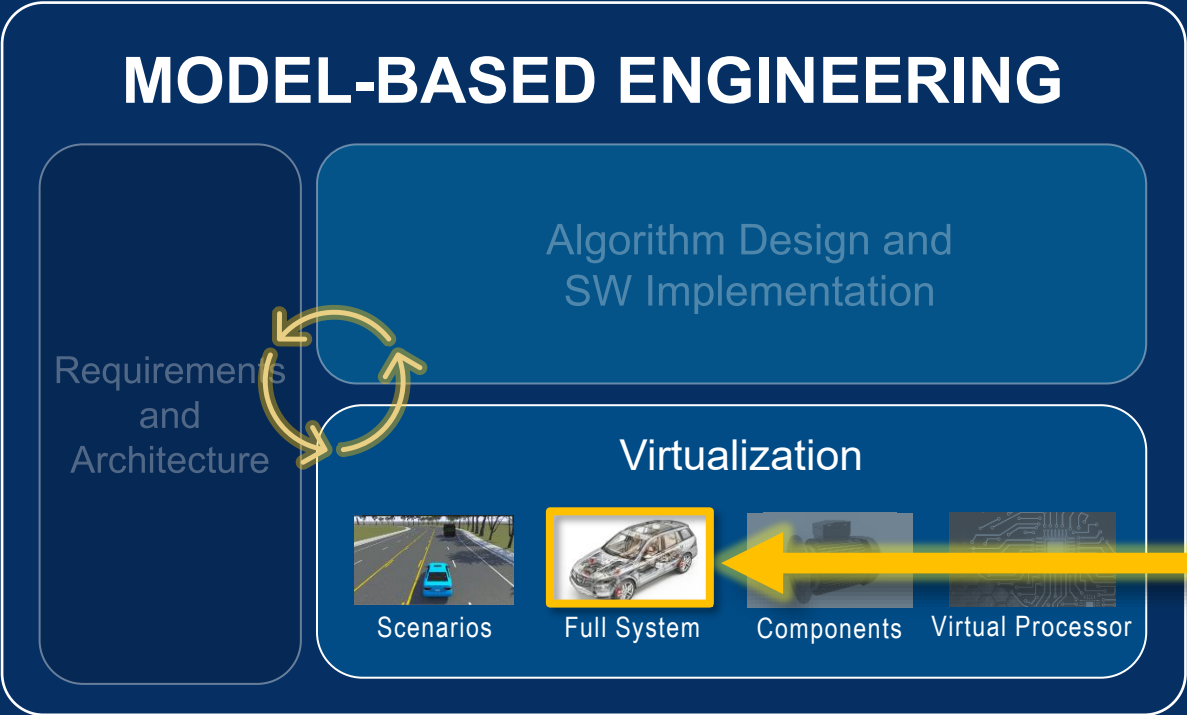


- Not only the ECU:
All aspects of the SDV
- Shift-Left and virtually integrate
- Automate validation with
Simulation

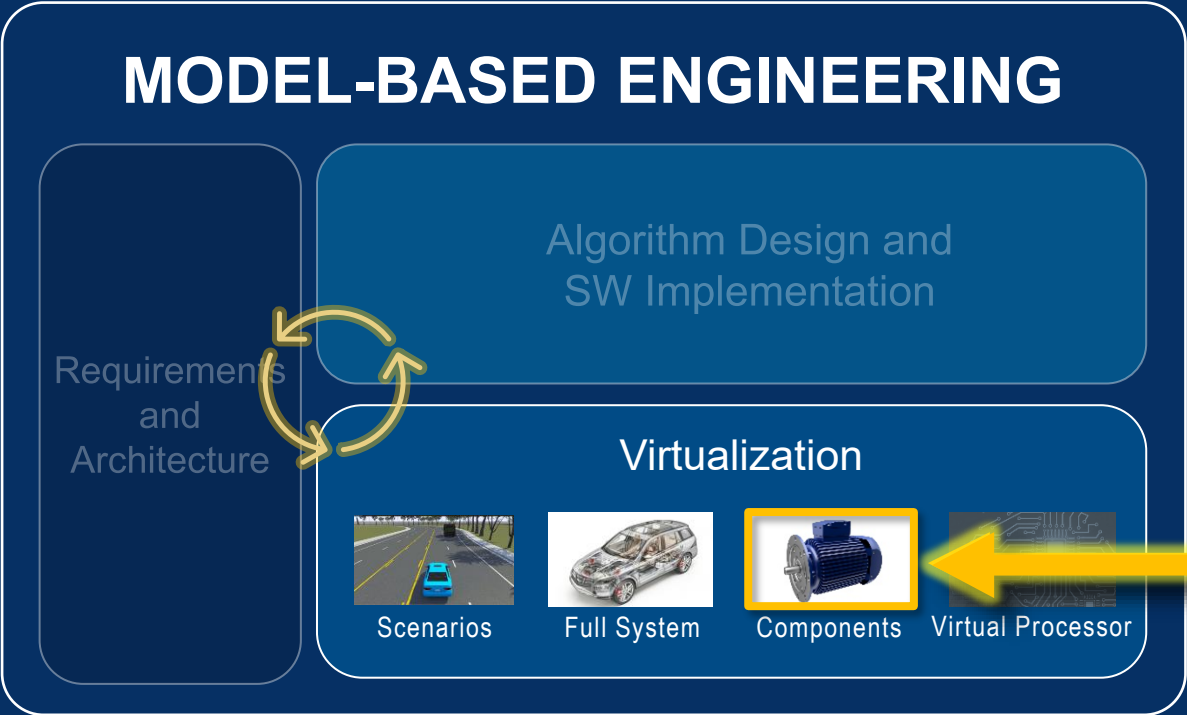
Virtualization: Scenarios



Virtualization: Vehicle



Virtualization: Components



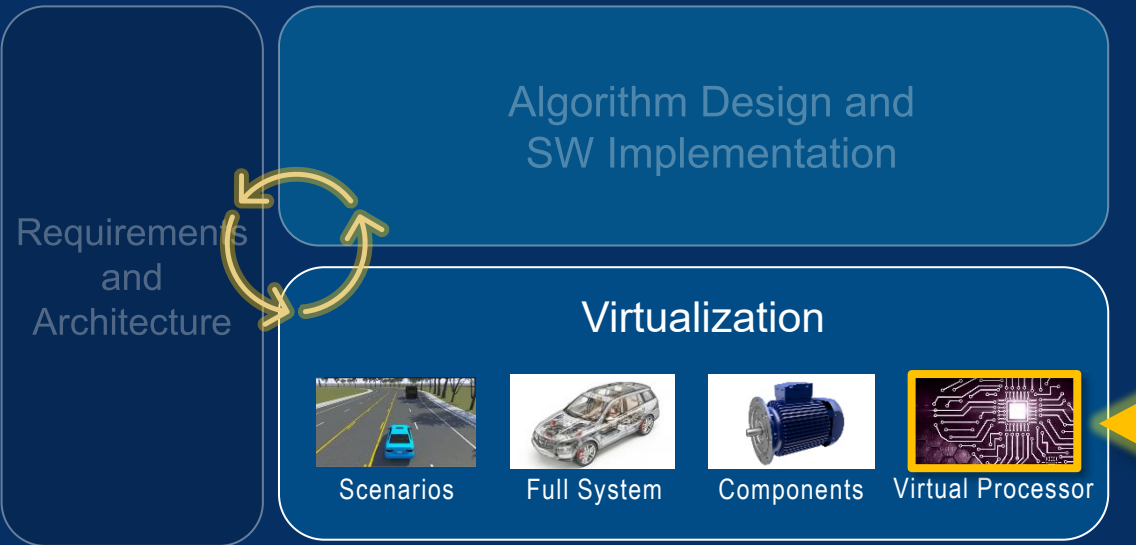
Component integration with 100+ third-party tools and languages

Systems engineering			
Propulsion and Battery			
Mechanical engineering			
Electrical engineering			
AI and code			

Virtualization: Processor

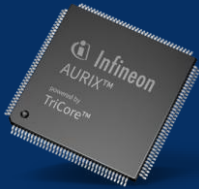
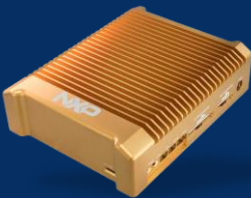
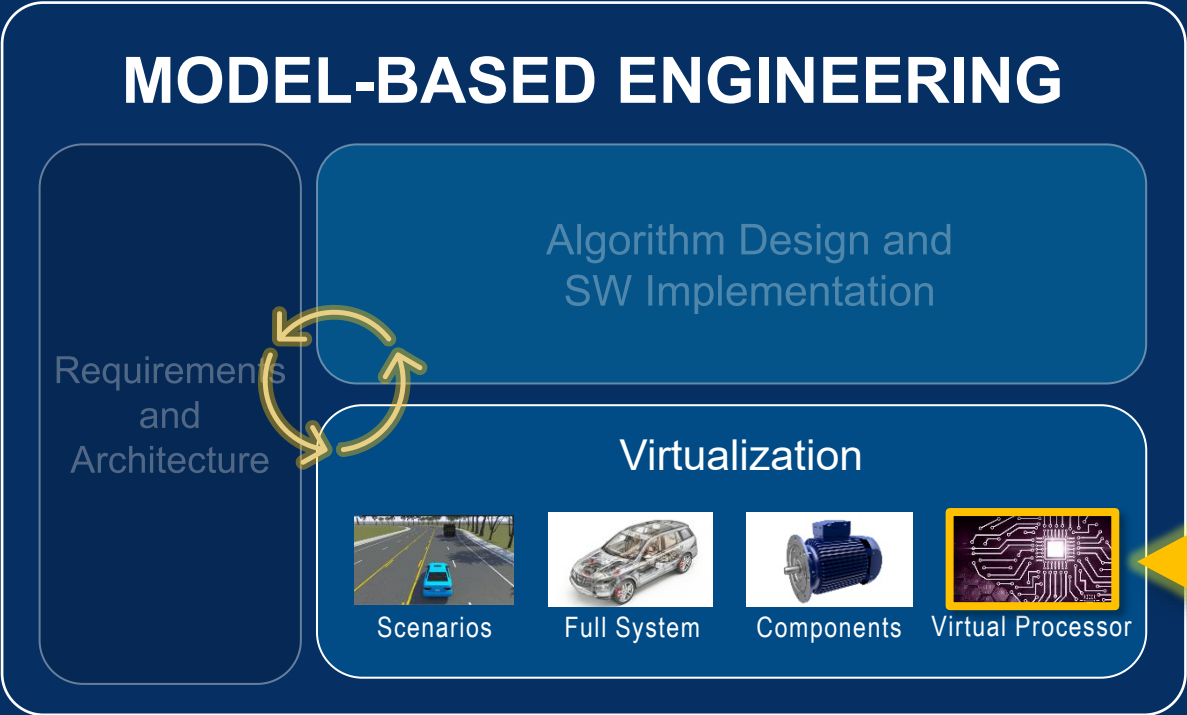


MODEL-BASED ENGINEERING



L0 L1 L2 L3 L4
prostep ivip Virtual ECU levels

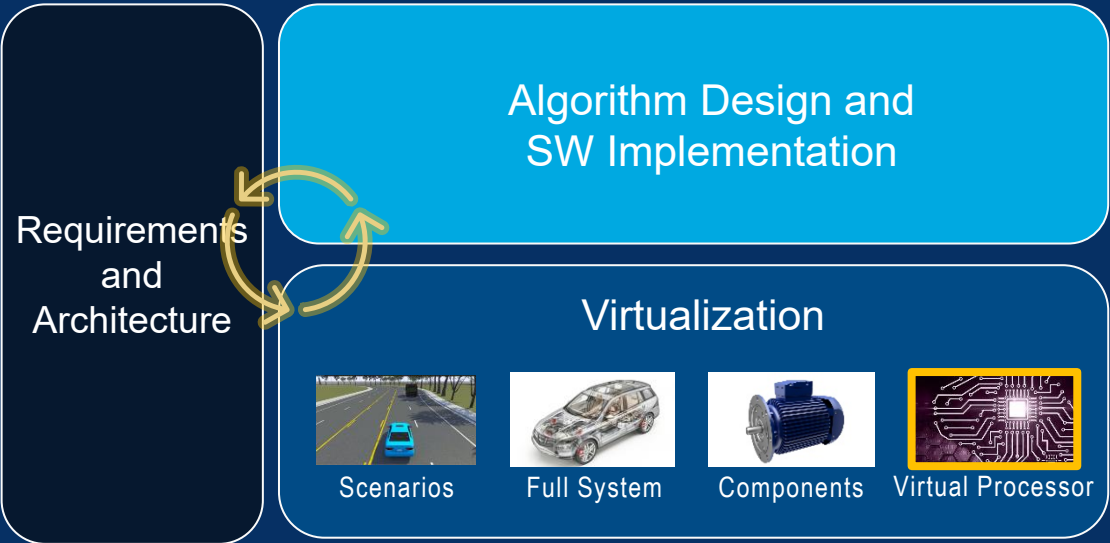
Virtualization: Processor



Virtualization: Processor



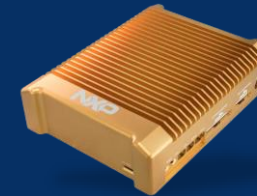
MODEL-BASED ENGINEERING



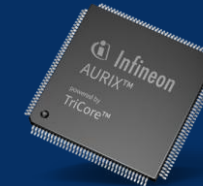
OPTIMIZE RIGHT



SHIFT LEFT



Qualcomm



CES - Final checkouts and test Private Link2

37:44

Take control Pop out Chat People Raise React View Notes Rooms Apps More Camera Mic Share Leave

Qualcomm Device Cloud

test link 1

Snapdragon® Ride Flex

HMI Simulation (QT)

938 RPM

7 51 RPM

4

1000

8.40

AEB

Hector Sanchez Flores (Unverified)

Project: New Save Log Signals Log Events Add Views Normal Fast Restart Step Back Pause Step Forward Data Inspector Sequence Viewer Logic Analyzer

DrivingTestBench 14DOF

Driving Simulation Test Bench

Algorithm Design and Code Generation

Virtual Processor (Qualcomm SoC)

Full System

View diagnostics 41% T=1.260 5% auto(ode45)

SA_AEB_CCRm_40.rrscenario | RR24a | MathWorks RoadRunner R2024a

SCENARIO EDITING

Scenarios

Simulation

Simulation Controls

Pause Step Forward Stop

Time: 2.220 s

Enable Pacing to Slow Down Simulation

Slower 0.05x 1x 20x Faster

Simulation Properties

Step Size: 0.03000 s Max Time: 24.910 s

Camera

Camera View Follow

Actor VUT

Distance 5.000

Height 3.000

2D Editor | Logic Playback

Library Browser

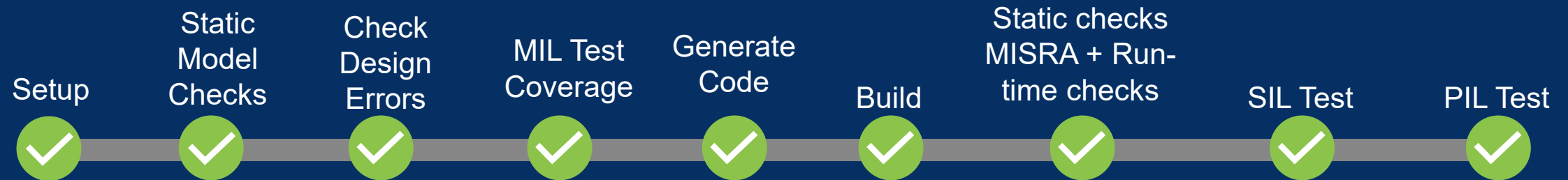
Assets

ADT Vehicles NCAP Assets Vehicle Textures

Ambulance Cement Truck CompactCar

Qualcomm MathWorks

Ease software integration and speed up delivery with Model-Based CI Pipelines



Process Advisor

PROCESS ADVISOR

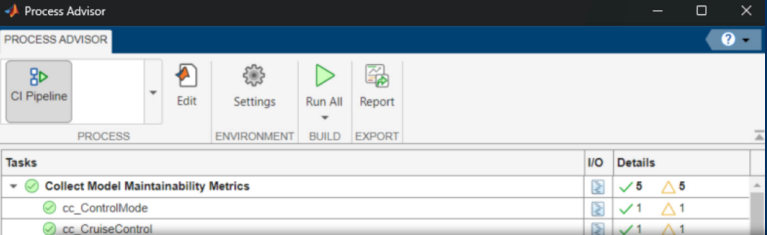
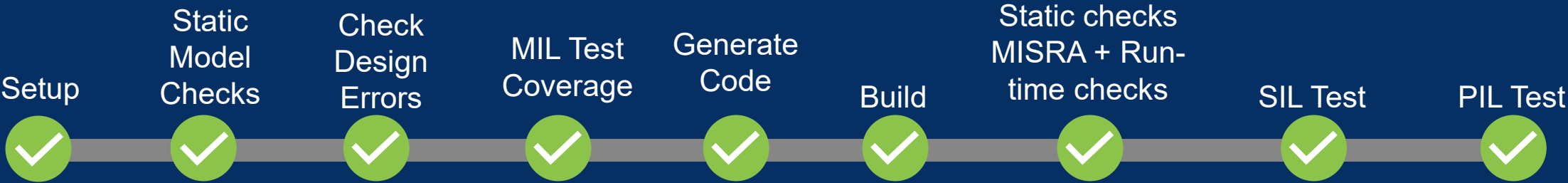
CI Pipeline

Edit Settings Run All Report

PROCESS ENVIRONMENT BUILD EXPORT

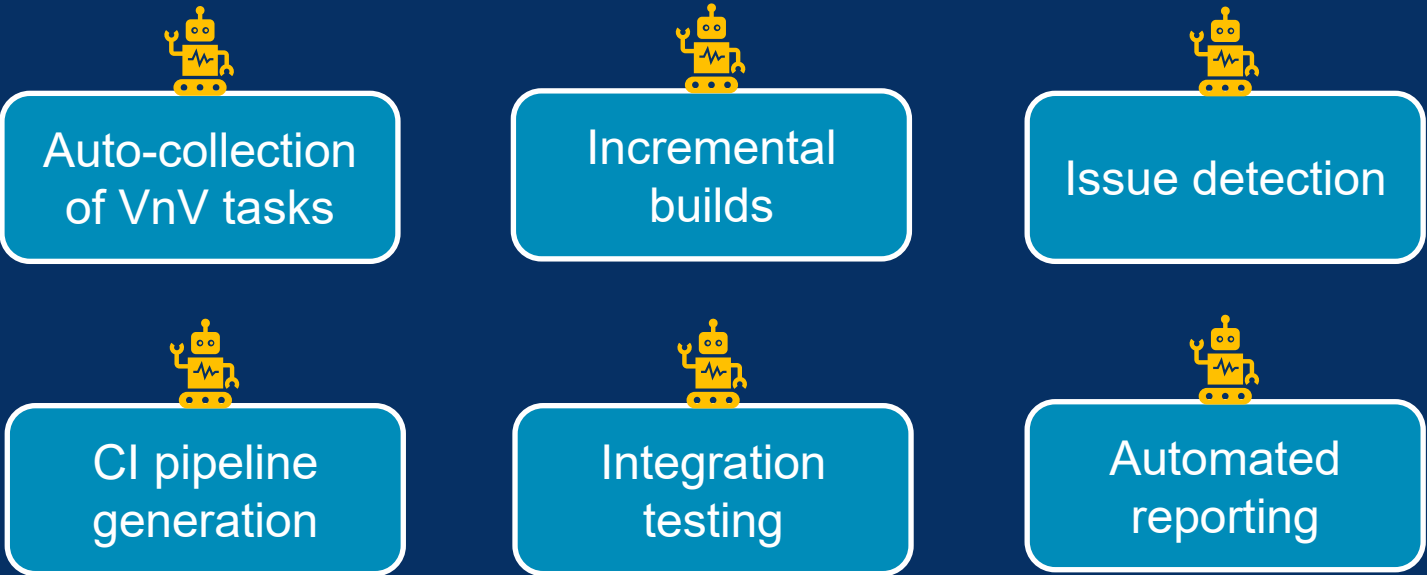
Tasks	I/O	Details
Collect Model Maintainability Metrics	5	5
cc_ControlMode	1	1
cc_CruiseControl	1	1
cc_DriverSwRequest	1	1
cc_LightControl	1	1
cc_ThrottleController	1	1
Check Modeling Standards	99+	10
cc_ControlMode	29	2
cc_CruiseControl	29	2
cc_DriverSwRequest	29	2
cc_LightControl	29	2
cc_ThrottleController	29	2
Generate SDD Report	5	
cc_ControlMode	1	
cc_CruiseControl	1	
cc_DriverSwRequest	1	
cc_LightControl	1	
cc_ThrottleController	1	
Run Tests	42	10
cc_ControlMode	27	10

Ease software integration and speed up delivery with Model-Based CI Pipelines

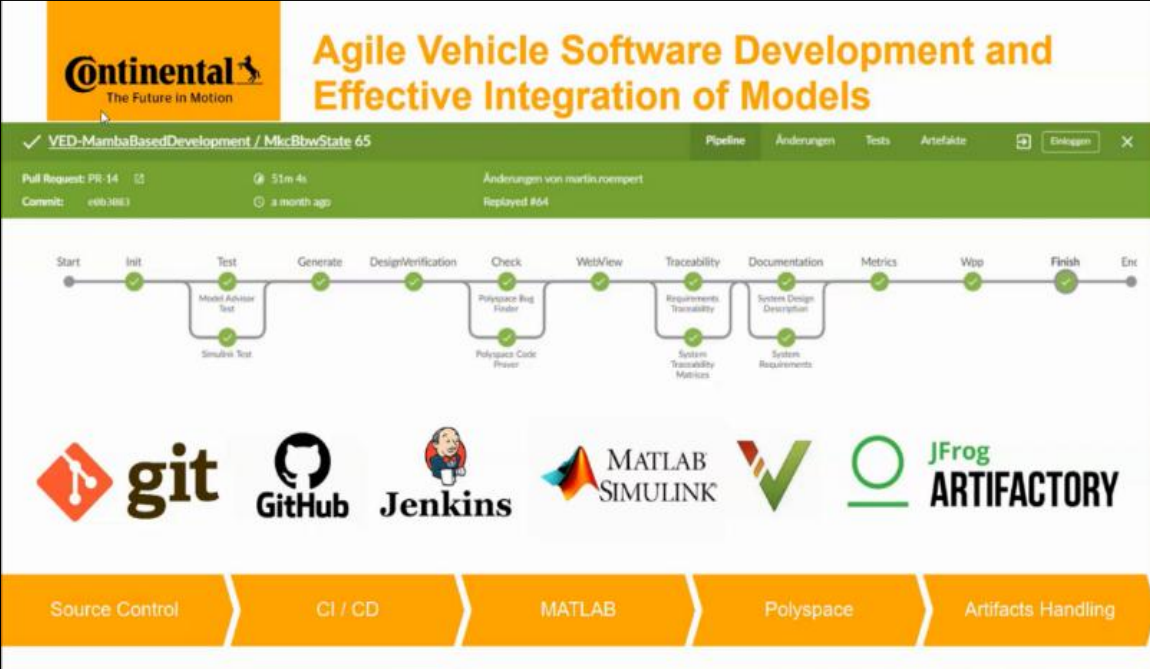


1. Summary Of Status

Task Status Totals					
Task Title	Overall Status	Pass	Fail	Error	Total
Check Modeling Standards	Pass	5	0	0	5
Collect Model Testing Metrics	Pass	4	0	0	4
Generate Code	Pass	5	0	0	5
Generate Model Comparison	Pass	5	0	0	5
Generate SDD Report	Pass	5	0	0	5
Merge Test Results	Pass	3	0	0	3
Run Tests	Pass	48	0	0	48



Implement Automotive-grade Software factory



Continental creates reusable CI Pipelines



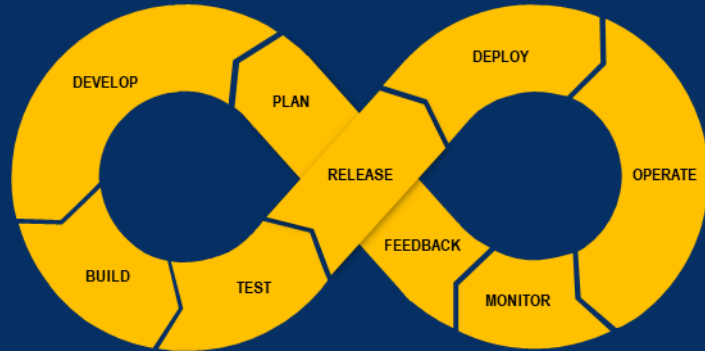
A MODEL-BASED SOFTWARE FACTORY FOR THE DEVELOPMENT OF SAFETY-CRITICAL SOFTWARE WITH AN ISO26262 COMPLIANT PROCESS



Dr Luc MALRAIT
Dipl.-Ing. Ghislain PONCET

March 21st, 2024

Achieve fast modern SW development **and** Compliance



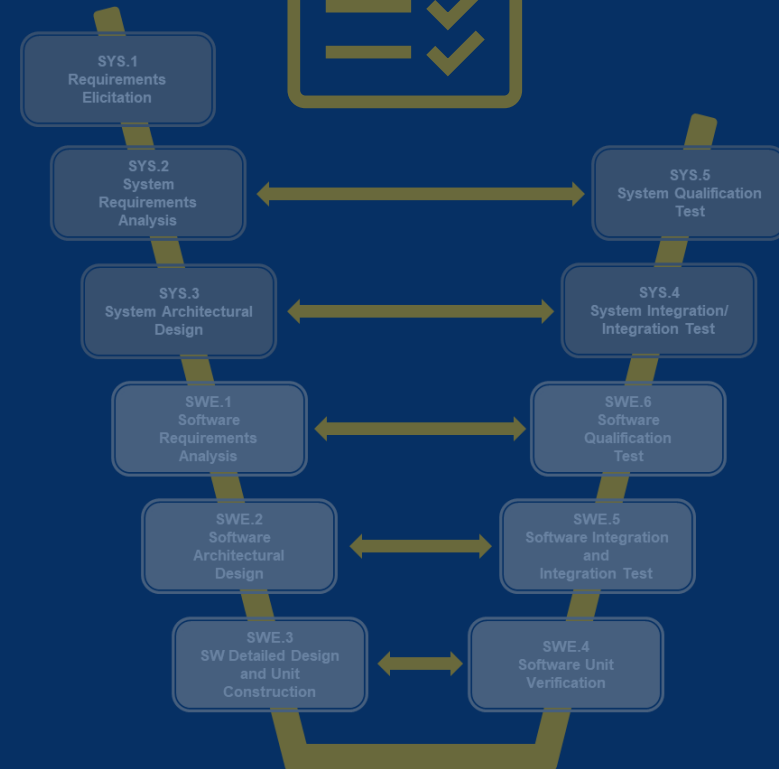
Virtualization / Shift-left

Process Automation

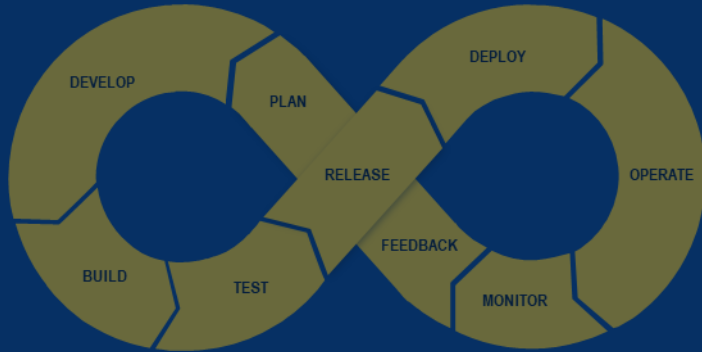
Reuse and integration

Scaling in the cloud

...



Achieve fast modern SW development **and** Compliance



Virtualization / Shift-left

Process Automation

Reuse and integration

Scaling in the cloud

...

Regulations

ISO 26262

SOTIF

Cybersecurity

...

SDV's compliance with regulations



- e.g. UN-Regulation (No. 155&156) and ISO standards (24089) for safe and secure automotive software updates

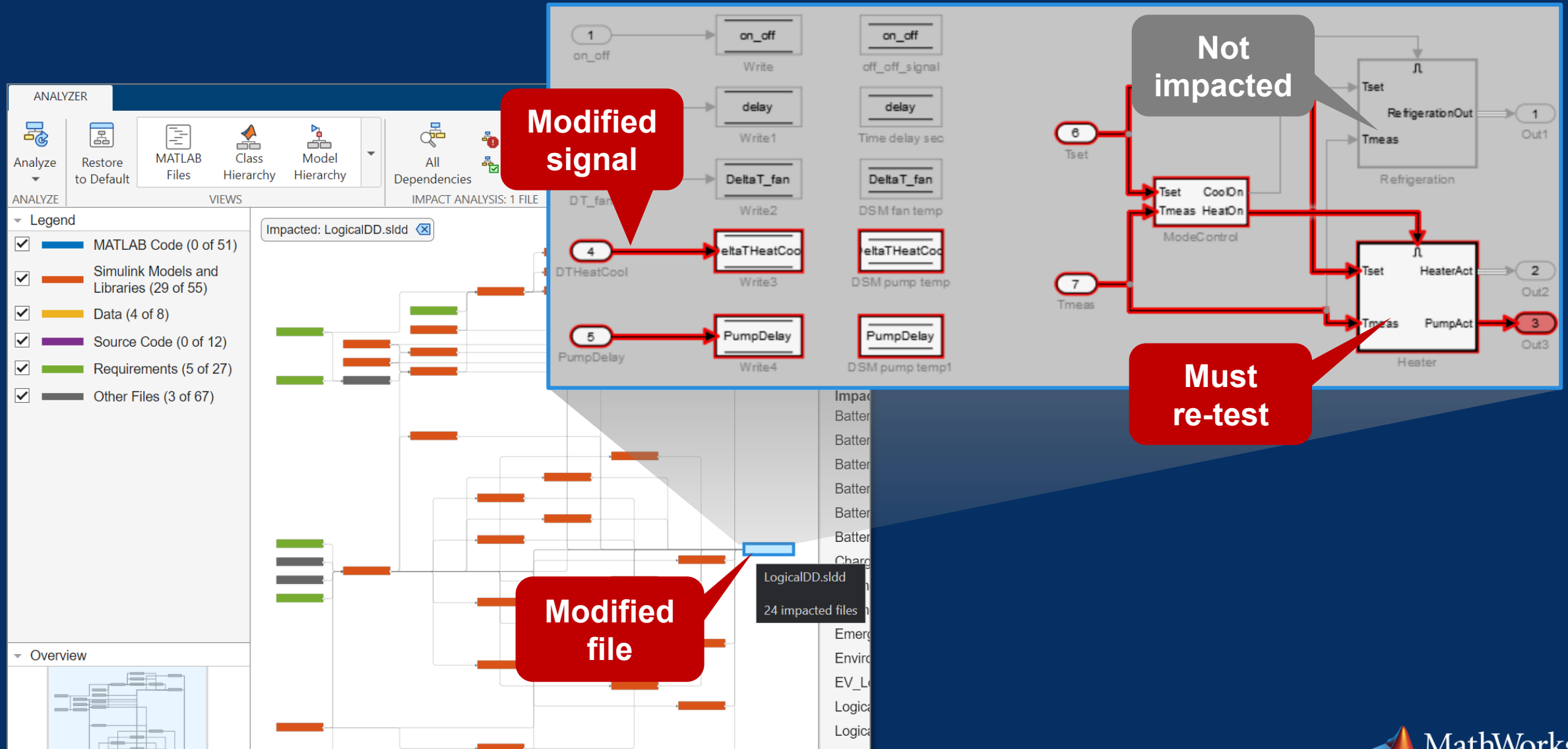
7.1.2.5.	Documentation for all software updates for that vehicle type describing:
(a)	The purpose of the update;
(b)	What systems or functions of the vehicle the update may affect;
(c)	Which of these are type approved (if any);
(d)	If applicable, whether the software update affects the fulfilment of any of the relevant requirements of those type approved system;
(e)	Whether the software update affects any system type approval parameter;
(f)	Whether an approval for the update was sought from an approval body;
(g)	How the update may be executed and under what conditions;
(h)	Confirmation that the software update will be conducted safely and securely;
(i)	Confirmation that the software update has undergone and successfully passed verification and validation procedures.



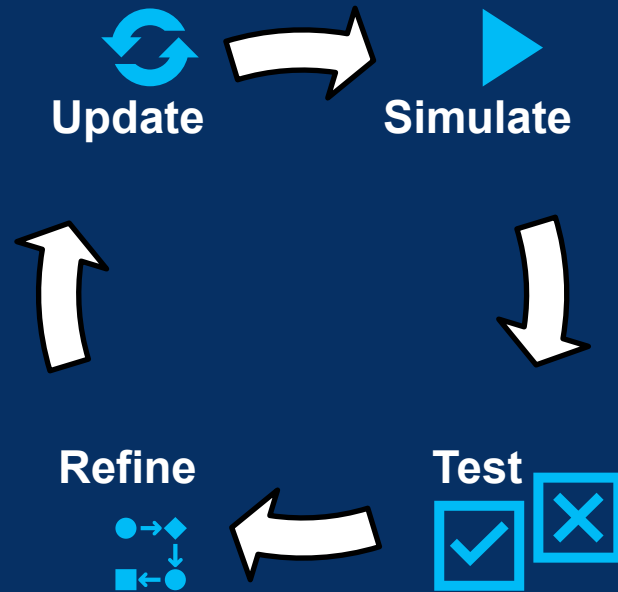
UN R156 Source: R156e (<https://unece.org/sites/default/files/2024-03/R156e%20%282%29.pdf>)

(b) What systems or functions of the vehicle the update may affect;

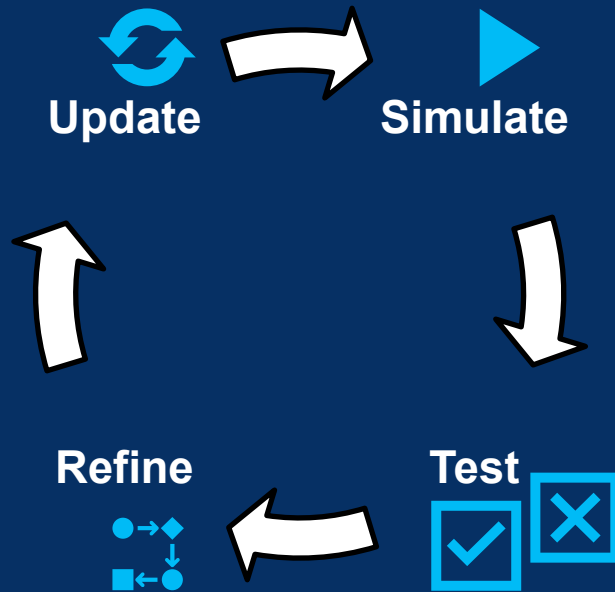
Tracking changes and minimizing re-certification (R.156)



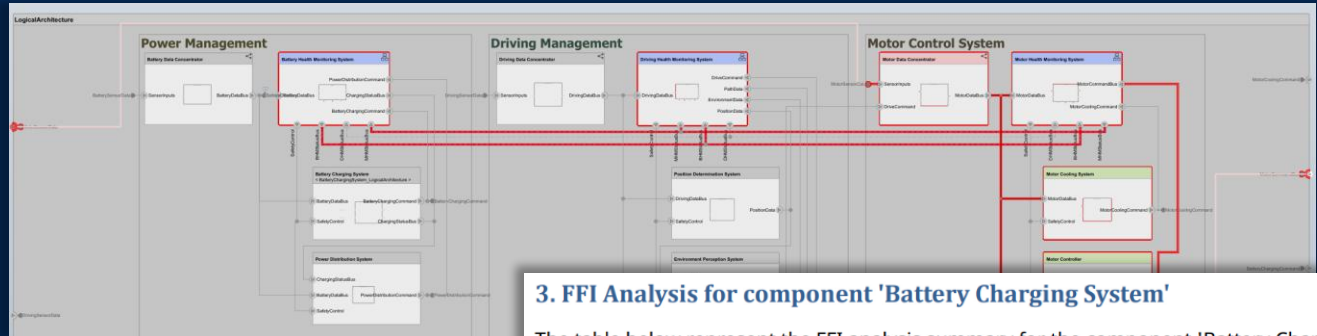
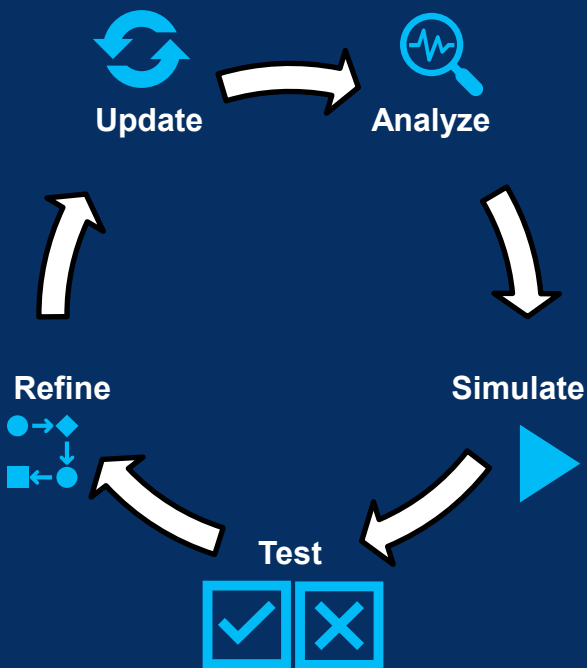
Architectural Models enable shorter integration cycles



Architectural Models enable shorter integration cycles



Architectural Models enable shorter integration cycles



3. FFI Analysis for component 'Battery Charging System'

The table below represent the FFI analysis summary for the component 'Battery Charging System'

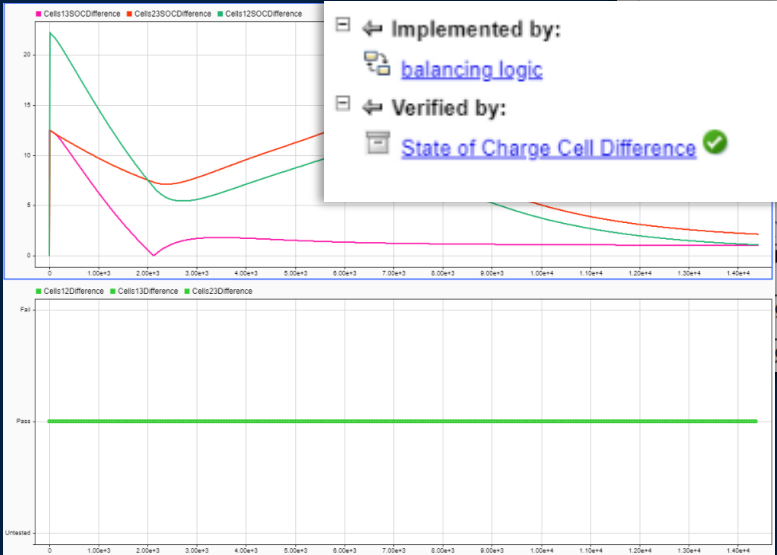
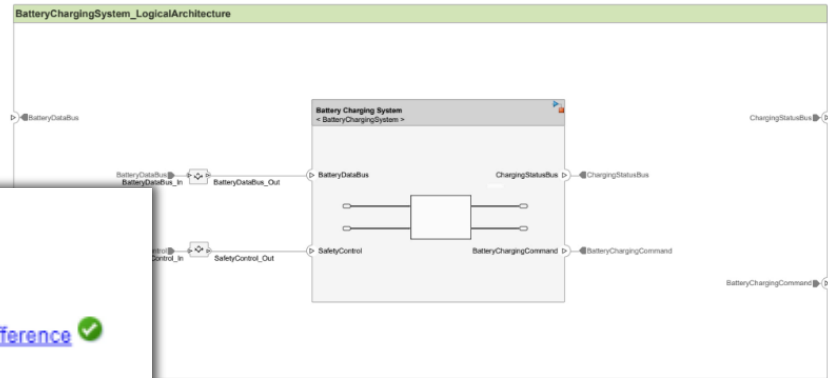
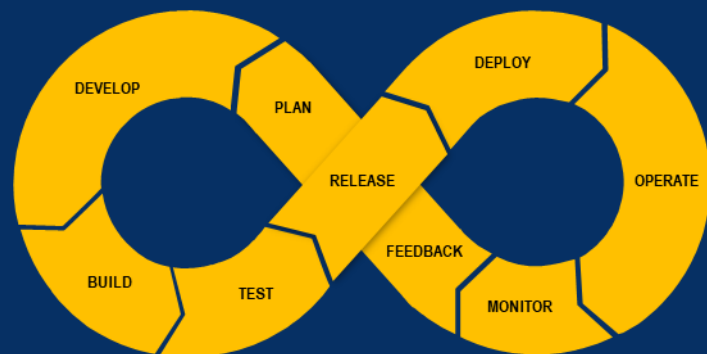


Table Result

Name	Src ASIL	Dest Component Name	Dest ASIL	ASIL Violation
g System"	"ASIL-B"	"Battery Health Monitoring System"	"ASIL-C"	"YES"
g System"	"ASIL-B"	"Power Distribution System"	"ASIL-QM"	"YES"

Achieve fast modern SW development and Compliance



Future-Proofing Automotive Software: Modularity, Reuse, and Safety with Model-Based Design

Dave Hoadley, PhD – Senior Principal Consultant at MathWorks
Brandon Trombley – Global Technical Account Manager at MathWorks

MathWorks
AUTOMOTIVE
CONFERENCE 2024
Europe

Accelerating Development of VCU Software in Iveco eDaily with Model-Based Design

Dr. Alessio Canepa, Iveco Group

MathWorks

Model-Based Design in the Context of Functional Safety

Dr. Marc Segetken
Principal Application Engineer
MathWorks

Dr. Tjorben Groß
Senior Application Engineer
MathWorks

© 2024 The MathWorks, Inc.

MathWorks

Accelerate A-SPICE® Compliance with Model-Based Design – Part 1
System Engineering Processes

Mohammad Abu Alquman
ITC Certification Kit
MathWorks
mabur@mathworks.com

Giovanni Vagnoni
Consulting Services
MathWorks
gvagnoni@mathworks.com

Marc Segetken
Application Engineering
MathWorks
msegetke@mathworks.com

RIVIAN

Modular Vehicle Configurator for Simulations with MATLAB and Git

Addhya Vignesh Jayaraman, Rivian
Oli Fairfax, Rivian

Seamless Tool Architecture From Requirement to Operations

Expectations for modern toolchains:

- State of the art processes and methods
- Highly automated
- CI/CD ready
- High availability
- Cloud-native
- Efficient and flexible license models
- Continuous and seamless rollouts
- Fast optimization cycles
- Modularization and interoperability

Integration time

Today: -44%
Future: -44%

Testing
Documentation
Build Time

Topic	
Cybersecurity: Identifying Asset & Threats	Watch video (38:24)
Cybersecurity: Risk Calculation & Link with Safety Analysis	Watch video (36:43)
Cybersecurity: Building & Verifying Countermeasures	Watch video (51:51)
Cybersecurity: Change Impact Analysis, Reporting & Framework Customization	Watch video (52:25)

MathWorks

Volvo Cars Software Factory Increases Pace and Quality of Development with Polyspace

Challenge
Develop reliable, standards-compliant software for the next generation of cars

Solution
Run static code analysis with Polyspace throughout the software development lifecycle

Results

- Critical run-time errors detected before field testing
- Improved productivity with better code reuse
- ASPICE, ISO 26262, and ISO/SAE 21434 certification requirements met

Volvo Cars uses Polyspace for static code checking throughout the development cycle.

With Polyspace, we can ensure software security and quality by identifying and fixing critical run-time errors before every code merge *
- Johannes Fofuss, Volvo Cars

Key takeaways



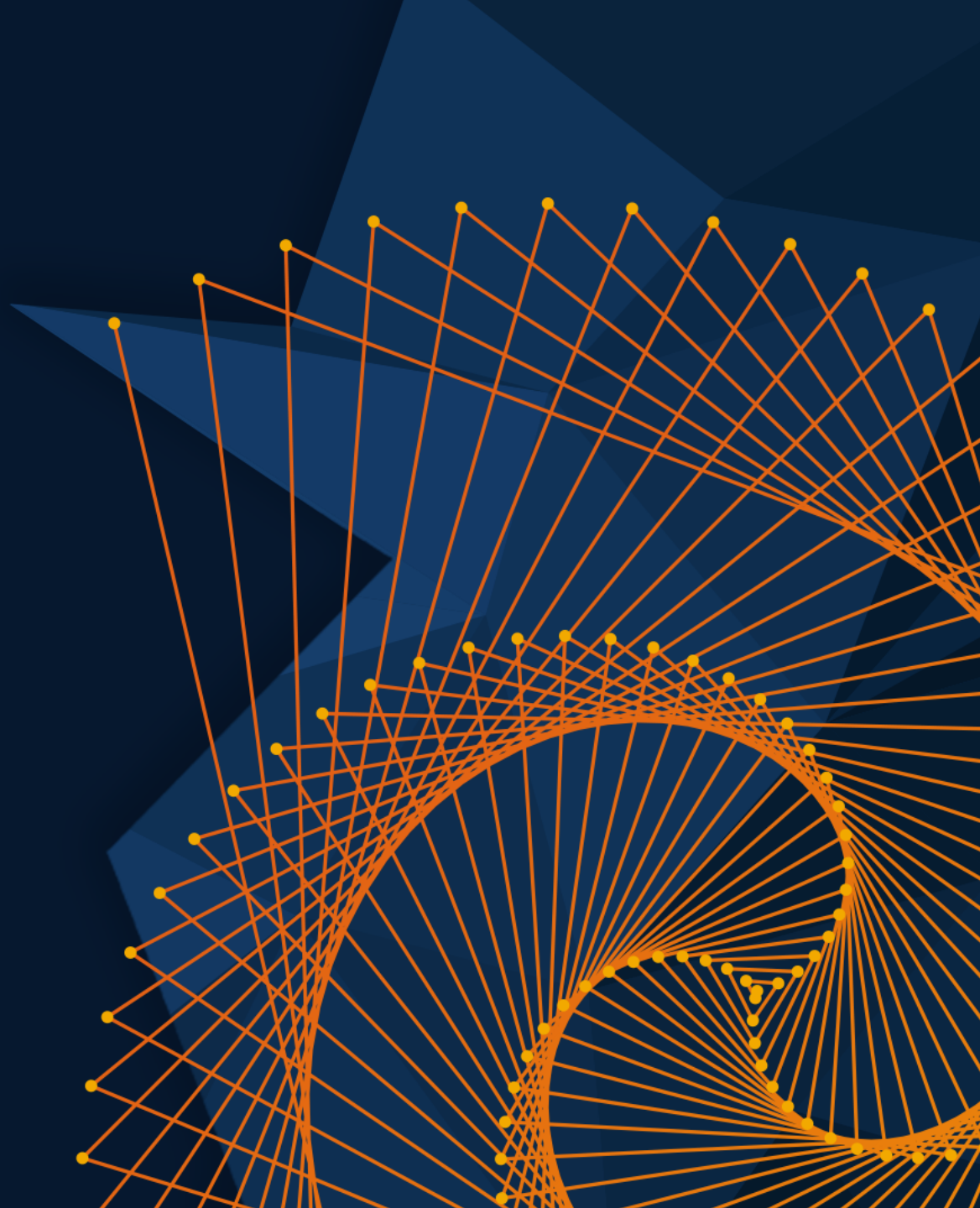
Augment SW factories with model-based simulation

- Shift left validation and verification with virtual environments
- Increase SW re-use and decouple from HW
- Reduce compliance costs and accelerate SW delivery

Thank you!



© 2025 The MathWorks, Inc. MATLAB and Simulink are registered trademarks of The MathWorks, Inc. See mathworks.com/trademarks for a list of additional trademarks. Other product or brand names may be trademarks or registered trademarks of their respective holders.





© 2025 The MathWorks, Inc. MATLAB and Simulink are registered trademarks of The MathWorks, Inc. See mathworks.com/trademarks for a list of additional trademarks. Other product or brand names may be trademarks or registered trademarks of their respective holders.

