MathWorks
AUTOMOTIVE CONFERENCE 2023
Korea

Model-Based Design for Software-Defined Vehicle

Young Joon Lee, MathWorks

MathWorks

The Rush for ~~Gold~~ **Software**

From the news…

*"**Software strategy** is one of the **key building blocks** of Stellantis' overall strategy to build the most sustainable mobility for our customers."*
Carlos Tavares – Stellantis CEO

*"The vehicle is no longer the central point of the automotive value chain, as **software, electronics and on-board intelligence increasingly determine both the value and use** of the vehicle for new mobility needs and services."*
Luca de Meo – Renault Group CEO

*"The **purpose** of the Group's own software and technology company CARIAD is to create the technical basis for **data-based business models**, **new mobility services** and **automated driving** (Level 4), and to leverage cross-brand synergies. Our aim is to **increase the proportion of software** in the vehicle that is developed in house."*
Excerpt from Volkswagen Software Strategy

https://www.stellantis.com/en/investors/events/sw-day-2021

https://www.renaultgroup.com/en/news-on-air/news/the-software-republique-a-new-ecosystem-for-innovation-in-intelligent-and-sustainable-mobility/

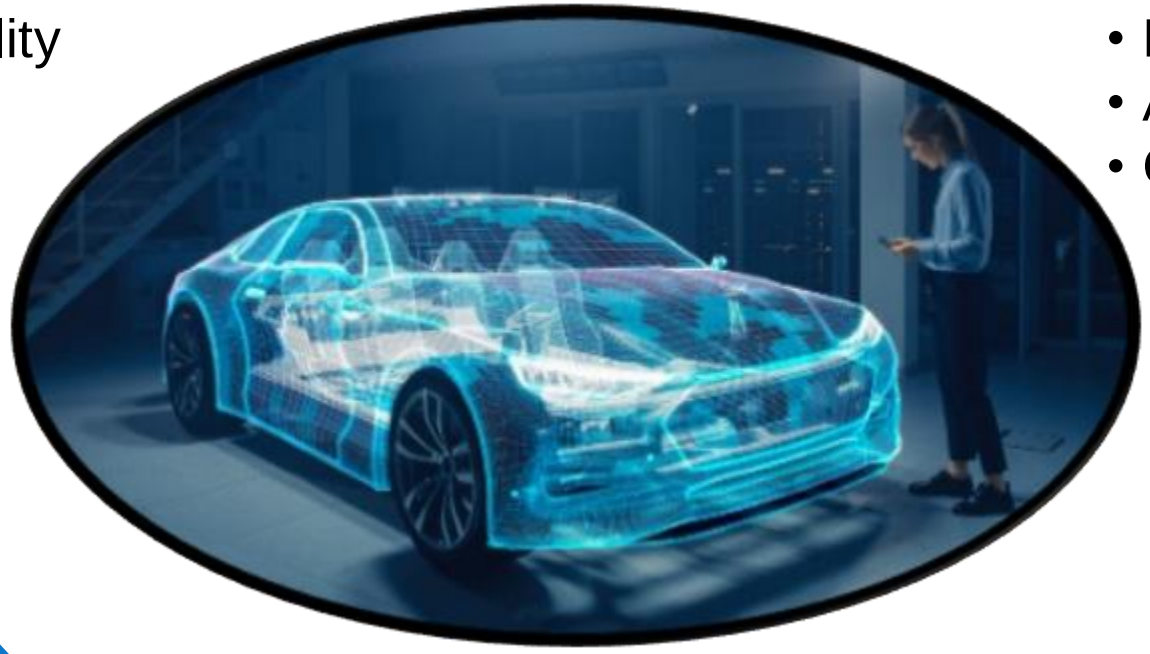
https://www.volkswagenag.com/en/strategy/software.html

Software-Defined Vehicle: Brand-distinctive features and main value for the customer will come from Software

Software-Defined Vehicle

demand

Customer Expectations
- Clean and Safe mobility
- Digital Life continuity



Technology & Innovation
- Electrification
- Autonomy
- Connectivity

invest

Business Opportunity
- App stores, SW features on demand
- SW services subscription plans

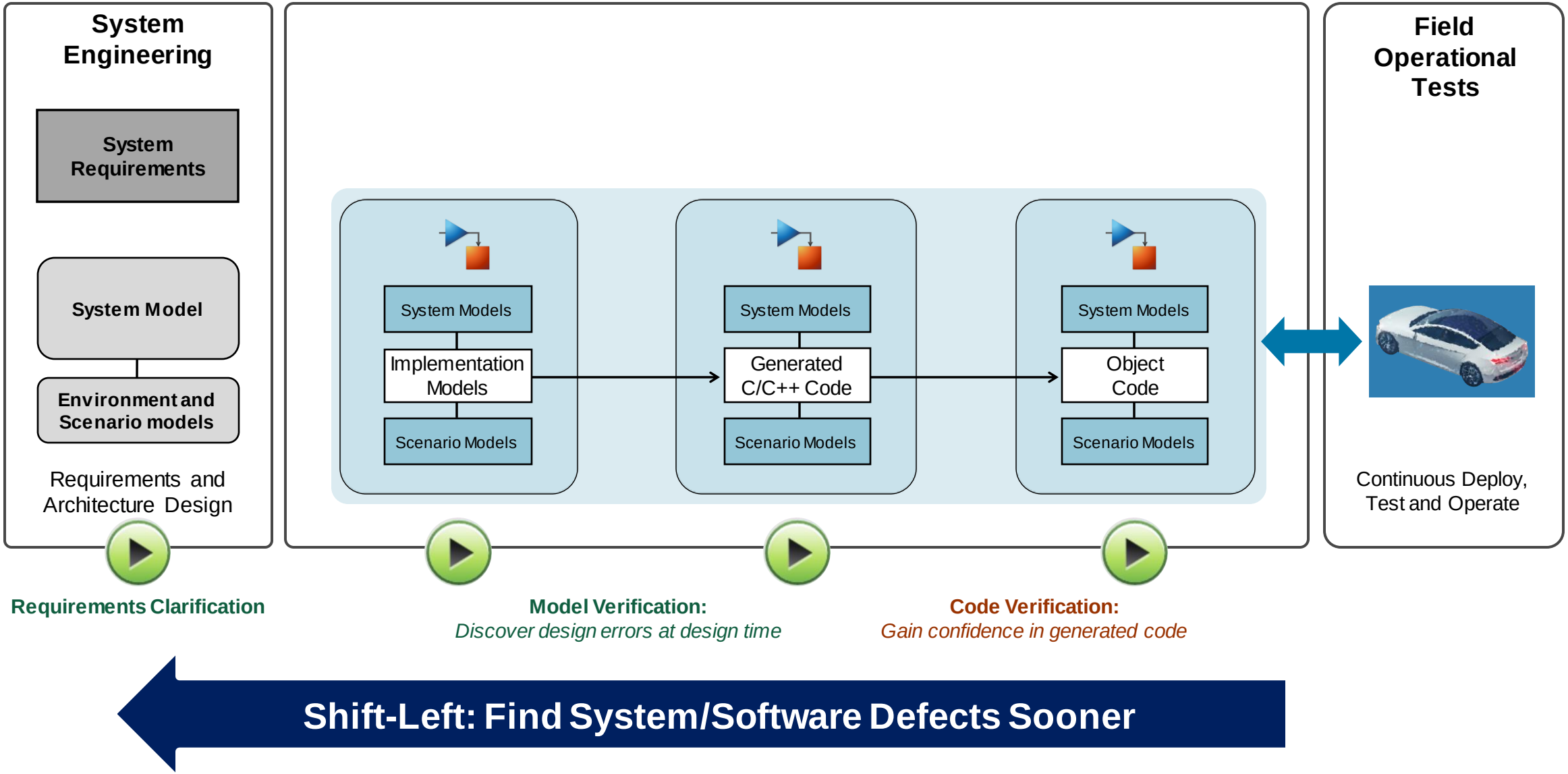
monetize

Model-Based Design

E-Powertrain & Energy | Chassis Control | ADAS & AD | Body Controls | IVI/Cockpit | Connectivity

Software: that interacts with the vehicle's physical systems
that needs real-time (deterministic) execution
that requires high assurance
that meets maturity-model principles (Automotive SPICE)
that supports standard architectures (AUTOSAR Classic and Adaptive)

Model-Based Design Today

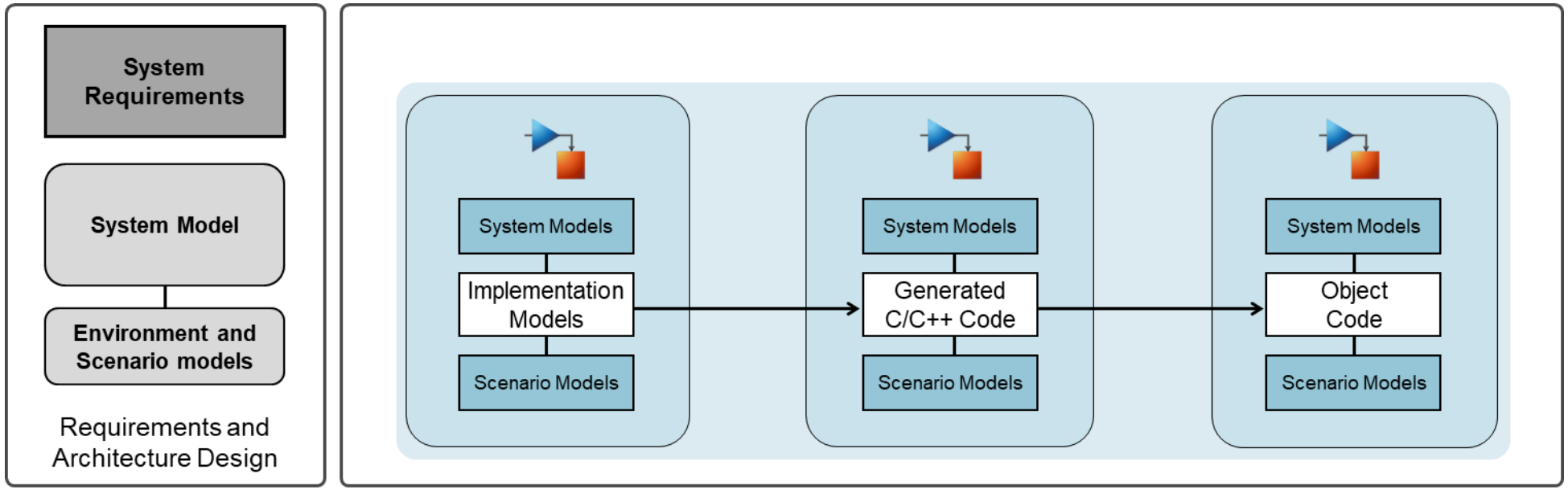


Shift-Left: Find System/Software Defects Sooner

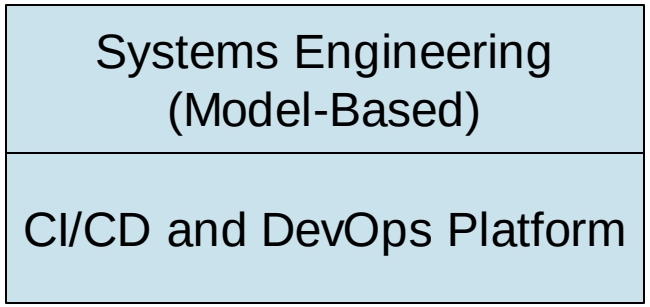
Approaches, Processes, and Tools

Model-Based Design for the Software-Defined Vehicle

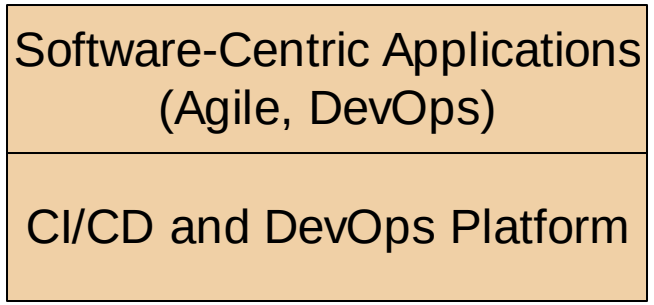
Established MBD Approaches		Updated and Additional MBD Approaches
Hard real-time, resource-limited ECUs	→	Could target ECUs, central/zonal computers, or cloud
Signal flow, invoked by RTOS/scheduler	→	Service-oriented, invoked by API calls
Tools packaged for desktop	→	Cloud execution, including as containers
Interactive, human-driven	→	Automate when possible, CI or Kubernetes driven
Focus on Design, Implementation, V&V	→	Focus on Production, Release, Throughput, Flow
Tool criteria primarily around Design	→	Tool criteria also include Automation, Cloud use
Managed by PLM systems	→	Trunk and Branch: Uses source & production repos
Installed from Windows	→	Stand up tooling using IaC (Infrastructure as Code)


System Requirements
System Model
Environment and Scenario models
Requirements and Architecture Design
System Models
Implementation Models
Scenario Models
System Models
Generated C/C++ Code
Scenario Models
System Models
Object Code
Scenario Models

Shifts in Mindset

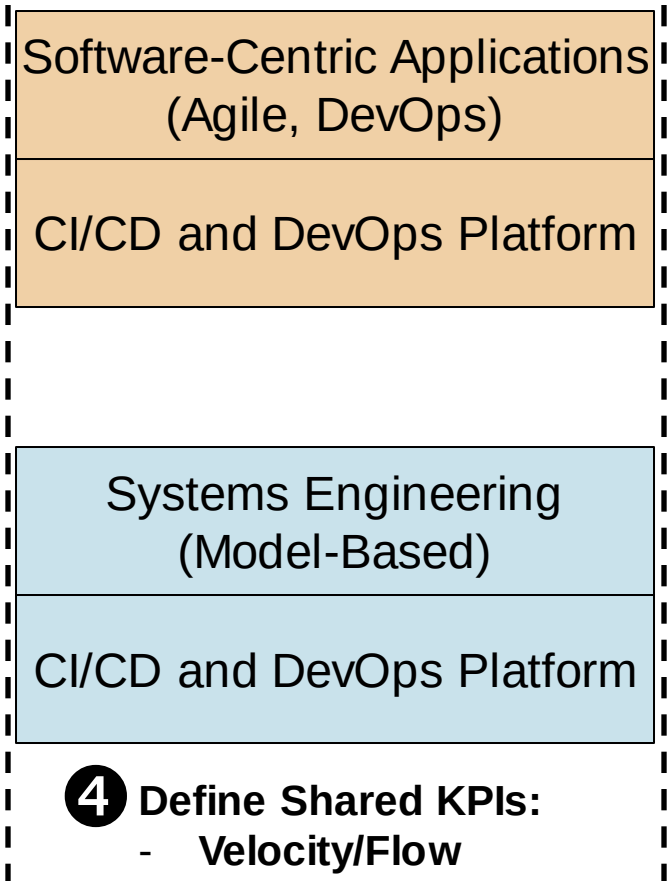


❶ **Shift Systems Team Mindset toward Automation and DevOps**



❷ **Shift Software Team Mindset to Leverage Systems via Simulation**

Shifts in Mindset

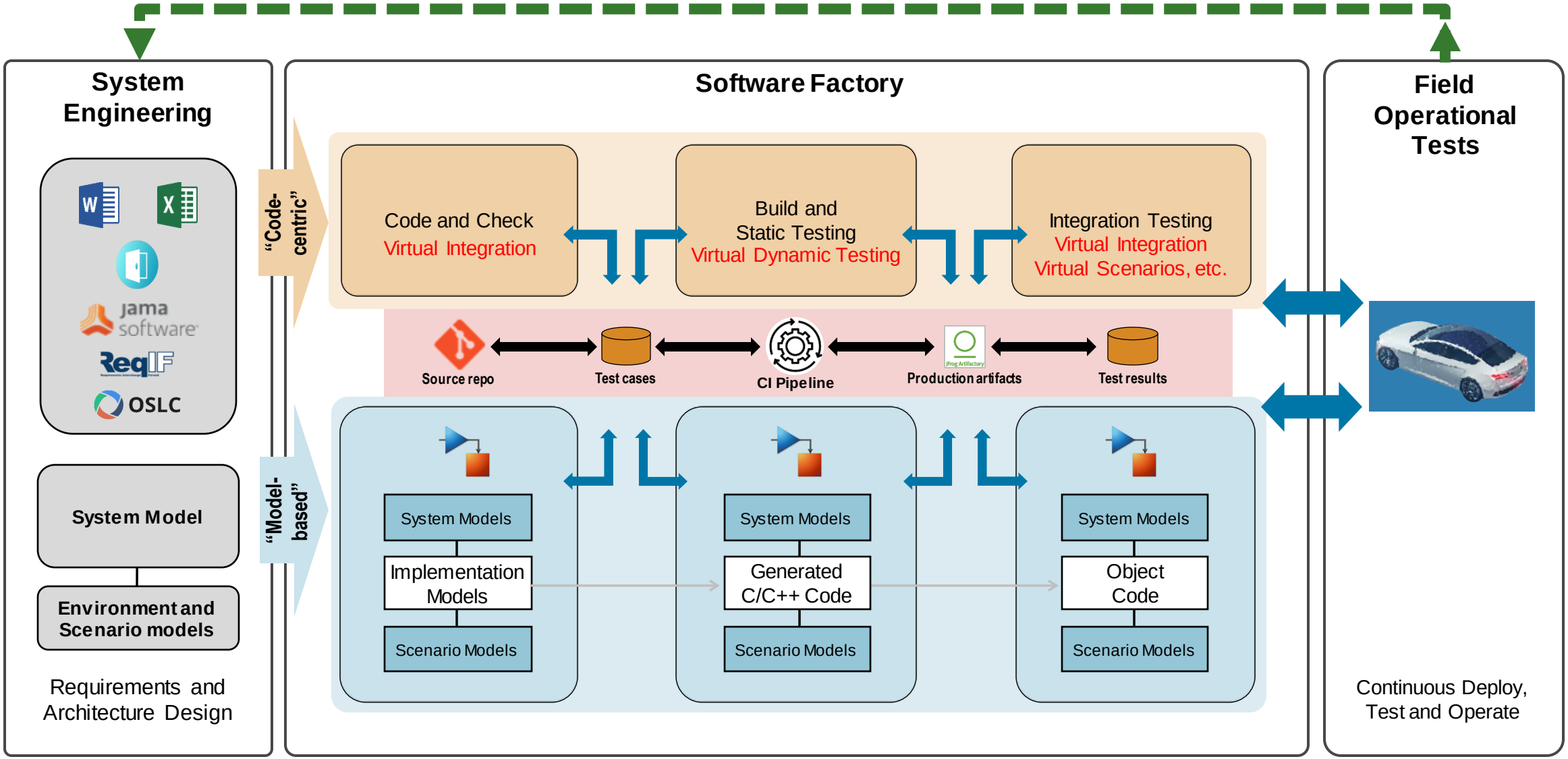


2 Shift Software Team Mindset to Leverage Systems via Simulation

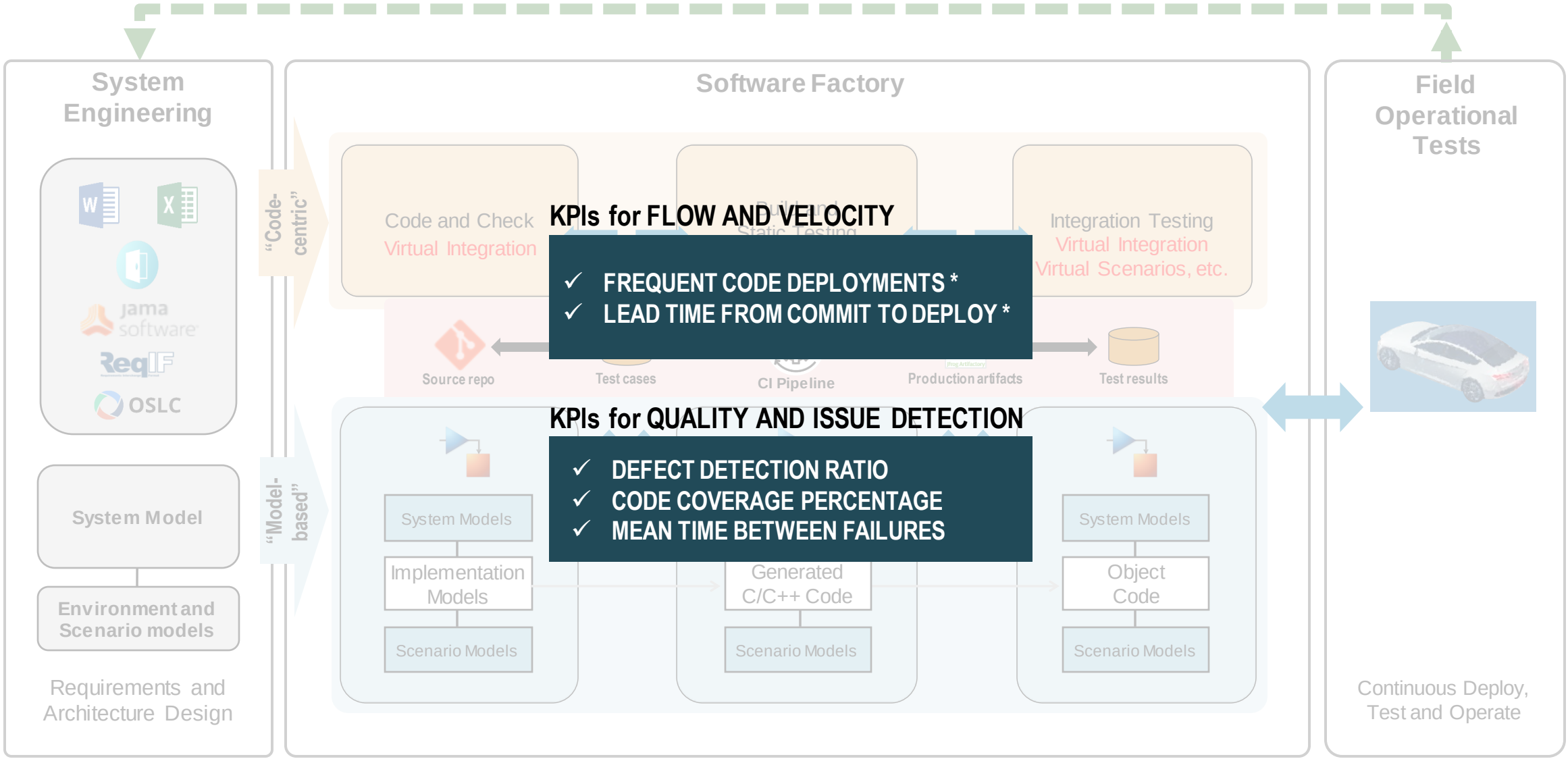
3 Shift Mindsets to Commonize CI/DevOps Platforms for Systems and Software

1 Shift Systems Team Mindset toward Automation and DevOps

Aligning and Automating MBD and Code-Centric Approaches



Automation of MBD Workflows

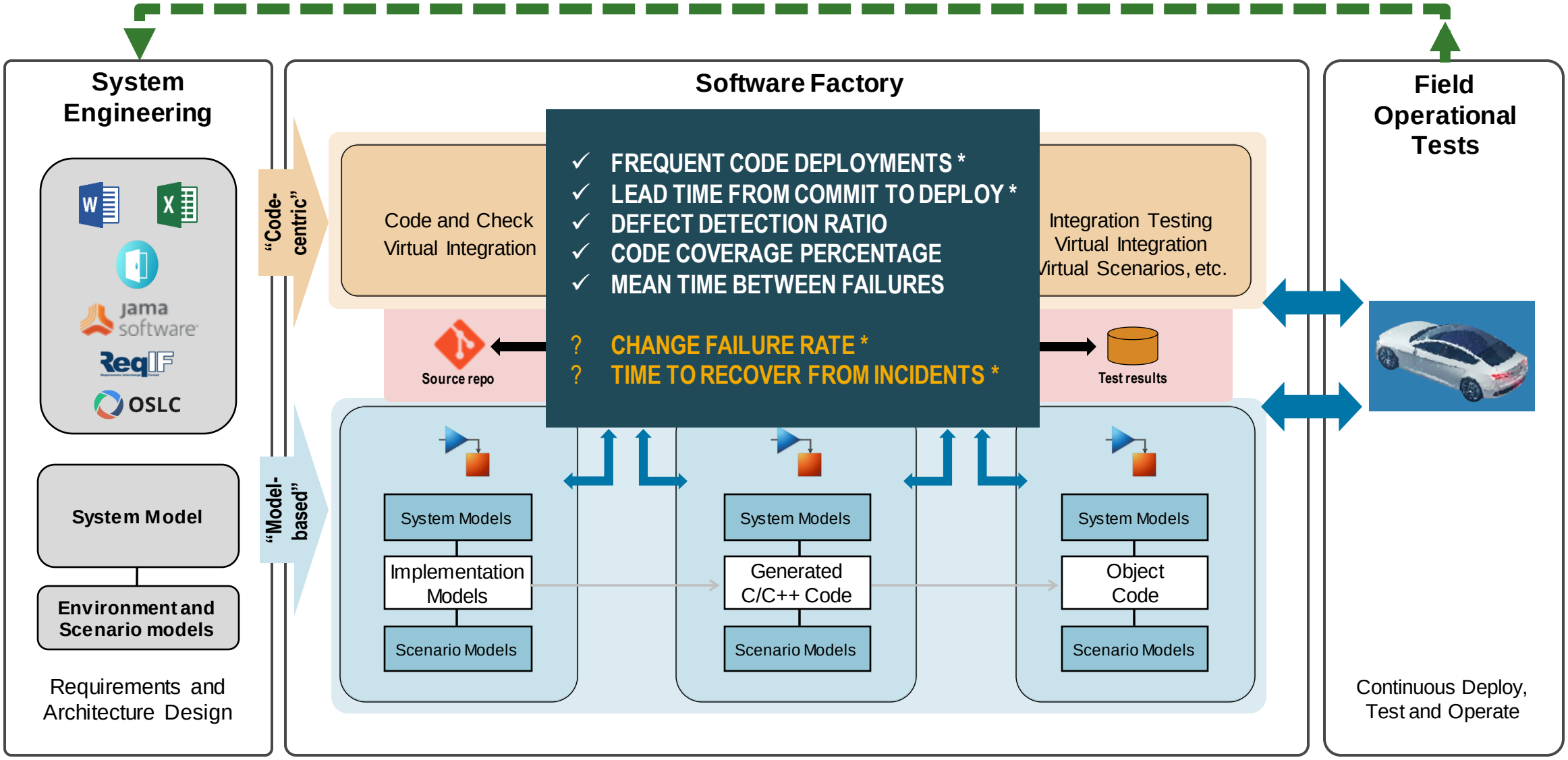


* DORA Metrics

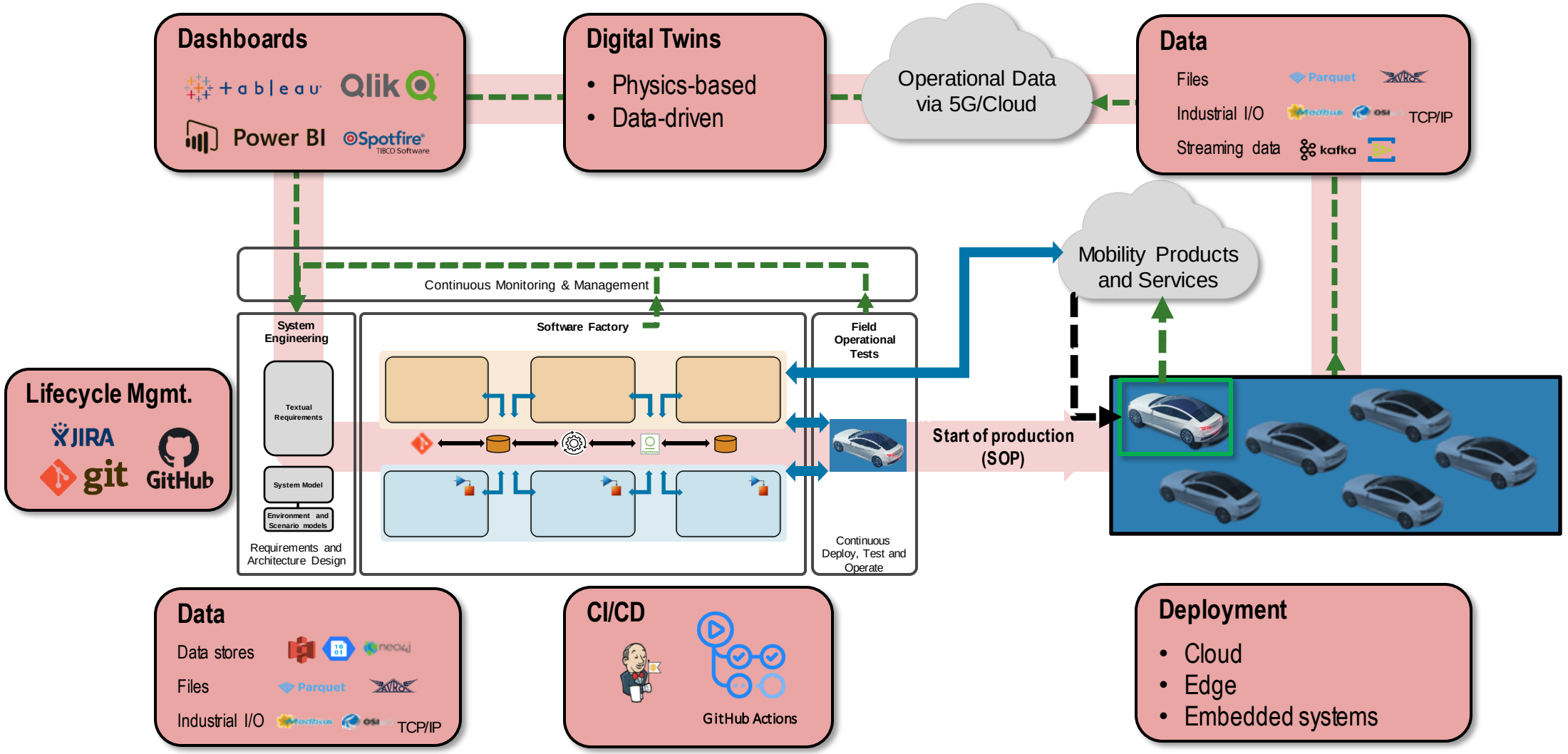
Team Interactions: Complexity in Technology, Culture, and Processes

* DORA Metrics

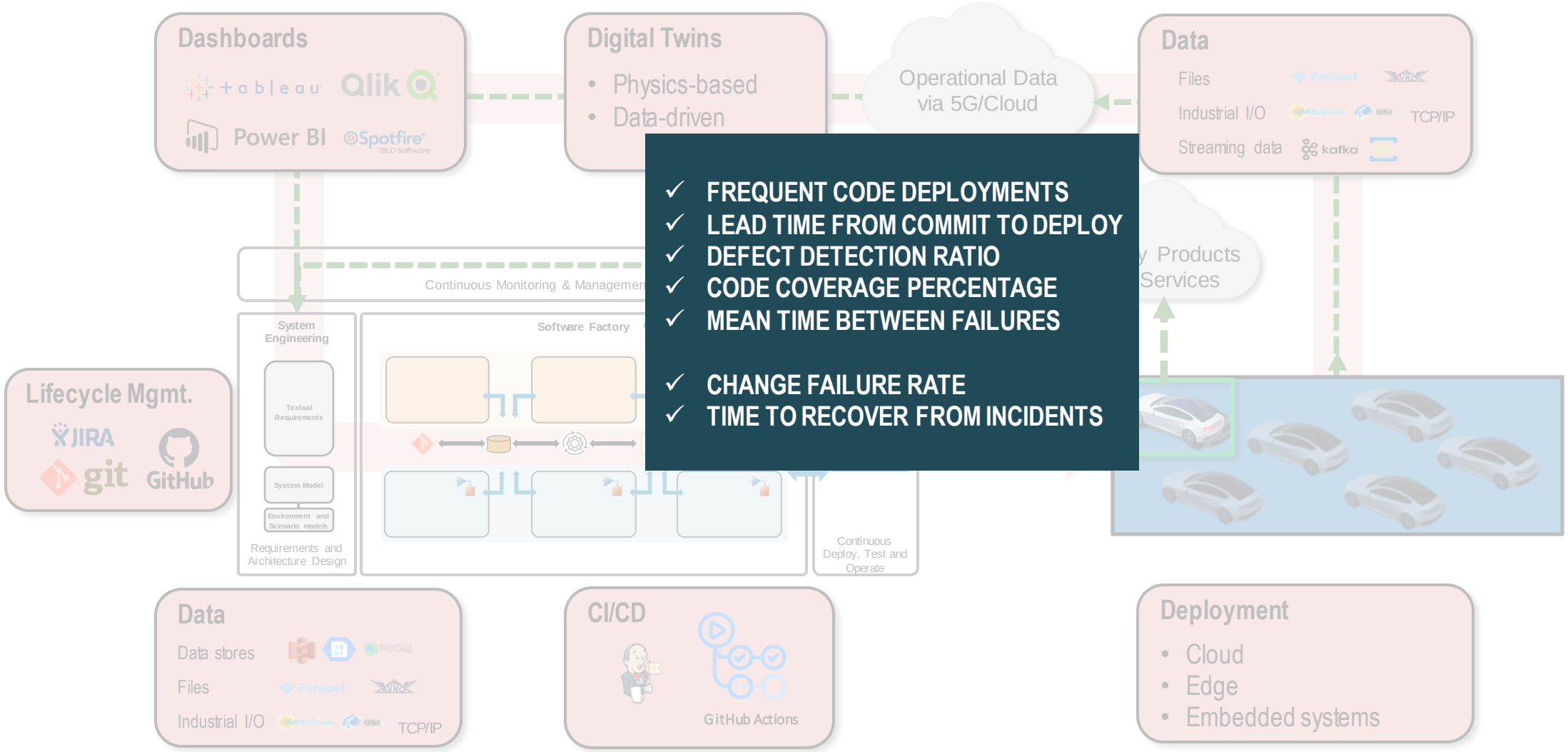
However, this only addresses part of the Big-Picture Goal



Accelerating Time to Recover from Incidents

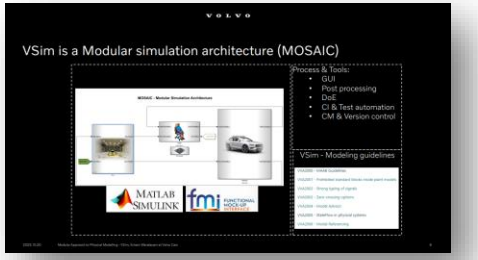


A Broader Set of SDV DevOps KPIs

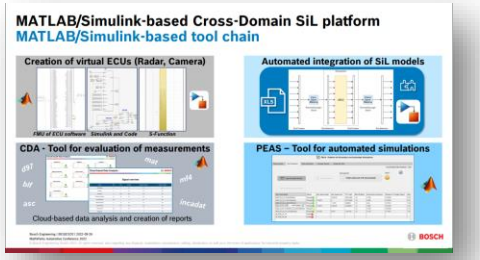


Enabling Automotive Customers to Leverage MBD in New Ways

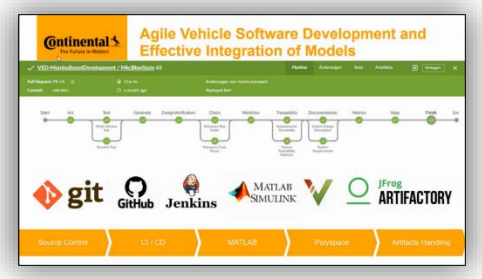
Established Automotive Companies



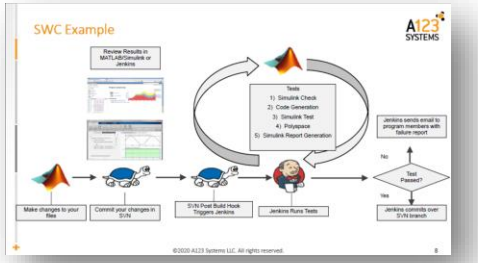
Volvo Cars



Bosch



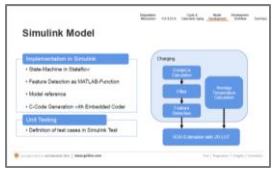
Continental



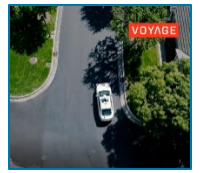
A123

Tech Companies and Start-ups

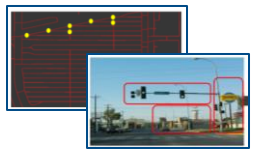
Lightyear One
Optimized energy flow from the solar panel and accompanying electronics to the vehicle



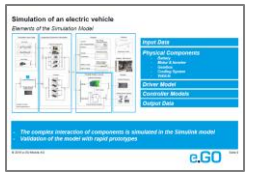
Gotion
High accuracy onboard battery state of health estimation



Voyage (acquired by Cruise)
Longitudinal controls for self-driving taxis



NIO
Battery management and vision guided localization



e.GO
Urban electric vehicle

TuSimple
Brake-by-wire system for autonomous truck

MathWorks

Service-oriented architectures (SOA) with Model-Based Design

Simulink is evolving to address the changing architectures

Modeling

Simulation

+

Automation

Coding | Verification

- **Fast design iterations** of service-oriented architectures and applications
- **Maximize reuse** of existing skills and assets
- **Ensure traceability** across all the stages

- **Generate code** compliant to automotive standards
- Deploy to **multiple targets**
- Enable **continuous integration**

Service Oriented Architecture

MathWorks

Comprehensive static analysis tools for increased efficiency

Catch and fix bugs while you code

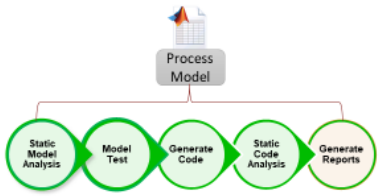
Automate with CI Workflows

Collaborate with Team Members

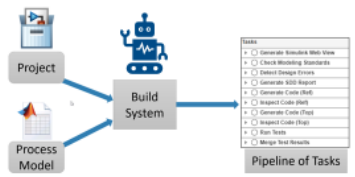
Code Quality

MathWorks

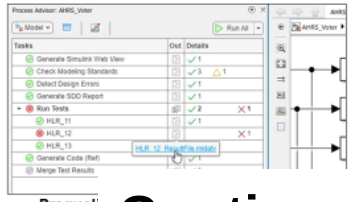
Simplify CI Setup by using the Automation Support Package



Prebuilt & Tailorable Model-Based Design Pipeline

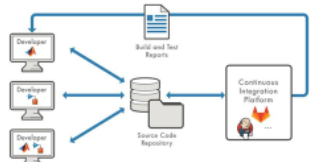


Build system to generate and optimally execute the process in your CI system



Prequali

Learn mor



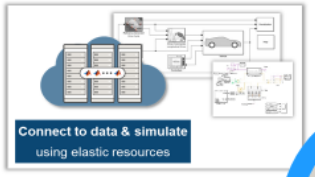
nmon CI Systems

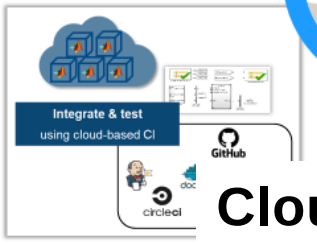
Continuous Integration

MathWorks

SDV Development in Action

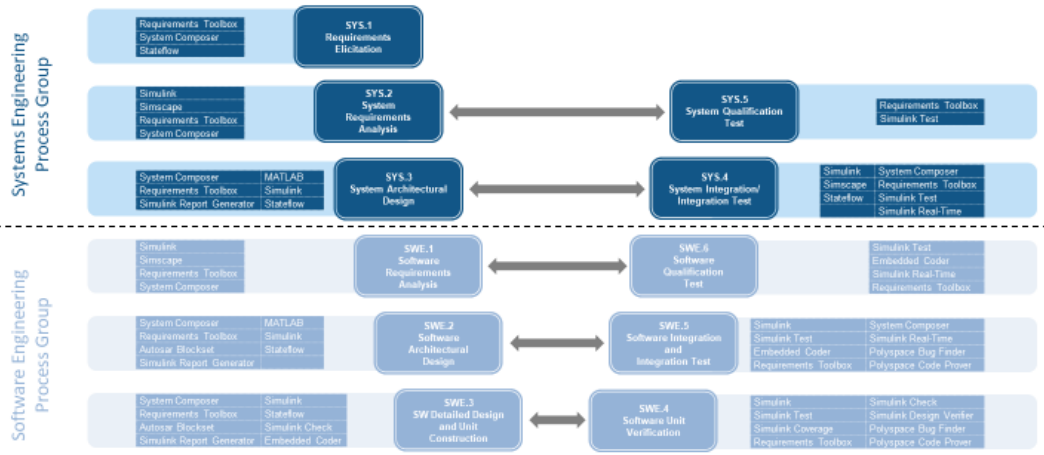
MathWorks
Elektrobit
aws automotive



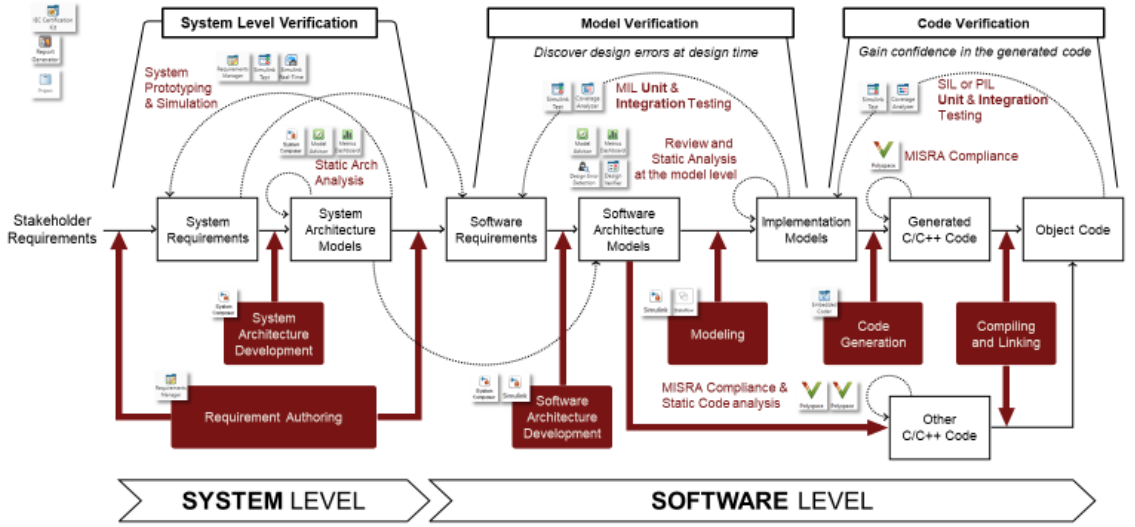


Cloud-Based Development

MathWorks MBD Toolchain Supports A-SPICE® System and Software Engineering



Reference Workflow for A-SPICE®



Model-Based Design & Automotive SPICE

Model & Detailed Design

- Thesis: „My model is my detailed design!“
- Model = Detailed Design, if fulfills:
  - Design Decisions documentation
  - Interfaces definition
  - Dynamic behavior description
  - Design review
  - Bidirectional requirements traceability
  - Consistency check
- Software Units
  - Implementation model
  - Code generation
  - Model has much more value than a static drawing

Base Practice	Measure	Recommended Tool or Functionality	Artif
BP1: Develop software detailed design.	+ Use Model Reference Blocks, Atomic Subsystems, Function-Call Subsystems or Simulink Functions to achieve functional decomposition into testable units + Use Interface view to assess signal flow and decomposition + Adhere to MAAB Modeling Standards, e.g. avoid mixing basic blocks and subsystems	Simulink® Stateflow® Simulink Verification and Validation® - Model Advisor MAAB Checks	N/A *Part and Desc*
BP2: Define interfaces of software units.	+ Use unambiguous names for Signals and Ports + Definition of complex interfaces with multiple signals through non-virtual busses (Bus Objects) + Link Interface Requirements to Data Dictionary Elements	Simulink® Simulink® - Data Dictionary Simulink Verification and Validation® - Requirements	N/A *Part and Desc*

MBD ASPICE Compliance Guideline

- **Result of collaboration:**
  - **Guideline for efficient ASPICE-conform Model-Based Design development.**
  - **MathWorks Expertise for customer support.**

VW-MathWorks joint project to establish guidelines for Simulink models to be used as Software Detailed Design

Bosch Improves Engineering Capability for ASPICE with MBD and System Composer

Challenge: need much more effective tooling than legacy tools for system architectural design (SYS.3) so Bosch Power Systems can increase project load and handle high design variance.

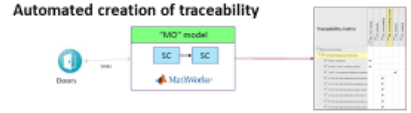
Solution: extending MBD toolchain to system engineering with System Composer

All Automotive Spice is SYS.3 system architectural design base practices are met:

- BP1 Develop system architectural design. → System Composer
- BP2 Allocate system requirements. → System Requirements
- BP3 Define interfaces for system elements. → System Composer
- BP4 Describe dynamic behavior. → Simulink & Stateflow
- BP5 Evaluate alternative system architectural design. → Option to use System Composer.
- BP6 Establish bidirectional traceability. → System Requirements
- BP7 Ensure consistency. → Simulink Requirements
- BP8 Communicate agreed system architectural design. → Report Generator

Automated creation of architecture models

Automated creation of traceability



Performing architecture design tradeoff with engineering tools

Automated generation of documentation

The Model Based Systems Engineering solution using the Mathworks toolchain achieves high user acceptance. It is proven by use in several customer accounts including an Automotive Spice assessment.

- High amount of projects → Re-use of reference models in both System Composer and Simulink.
- Design variance → Automated creation of models with System Composer architecture import meeting the individual design.

The Path Forward Requires Four Strategic Clusters of Action:

1. **Process**
   - Align software development and system engineering approaches
2. **People**
   - Collaborative, building synergies with new teams
   - Domain skills, re- and up-skilling the existing work force
3. **Methods**
   - Agile, DevOps
   - Parallelize and virtualize development
   - “Software factory” mindset of development-process automation
4. **Standards**
   - Legislative regulations, functional safety, cyber-security, AUTOSAR compliance, etc.