MathWorks
AUTOMOTIVE CONFERENCE 2024
Korea

미래자동차 개발을 위한 Model Based Design의 진화

유재흥 프로, MathWorks Korea

MathWorks

Headquarters

Natick, MA USA

North America

United States

Europe

Finland
France
Germany
Ireland
Italy
Netherlands
Spain
Sweden
Switzerland
UK

Asia-Pacific

Australia
China
India
Japan
Korea

6,500+ staff

in 34 offices around the world

$1.25+ billion

in revenues

Privately held

and profitable every year

A Brief History of MathWorks

First Natick office, 1988

1984 – **Founded in Palo Alto, CA**
by Jack Little and Cleve Moler

1985 – **First MATLAB purchase order by MIT**

1988 – **Move to first Natick office**
at 21 Eliot Street

1990 – **Simulink released**

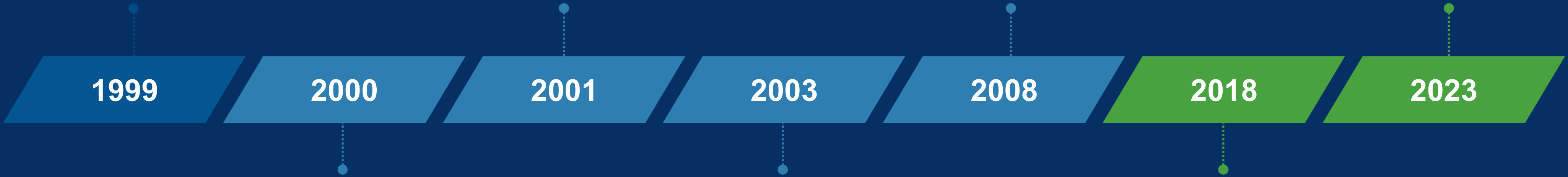
1993 – **Launch mathworks.com,**
one of the first 500 registered domain names

1997 – **First international office opens**
in Cambridge (UK)

1998 – **First $100M year**

A Brief History of MathWorks

Apple Hill headquarters



- **1999**: Move into Apple Hill headquarters
- **2000**: **Six new offices open:** Michigan, France, Germany, Spain, the Netherlands, and Switzerland
- **2001**: 1,000 staff members
- **2003**: L-shaped membrane officially becomes MathWorks logo
- **2008**: India office opens
- **2018**: First $1 billion year and Lakeside campus opens
- **2023**: 6,000 staff members

A Brief History of MathWorks

1999

Move into Apple Hill headquarters

2000

Six new offices open: Michigan, France, Germany, Spain, the Netherlands, and Switzerland

2001

1,000 staff members

2003

L-shaped membrane officially becomes MathWorks logo

2008

India office opens

2018

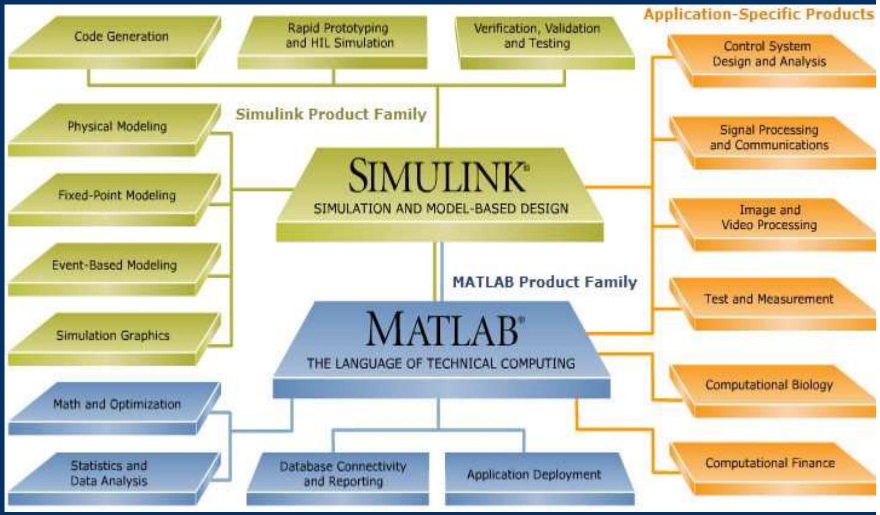
First $1 billion year and Lakeside campus opens

2023

6,000 staff members

MathWorks Products and Services

Use the right features based on your application



Code Generation

Rapid Prototyping and HIL Simulation

Verification, Validation and Testing

Application-Specific Products

Control System Design and Analysis

Signal Processing and Communications

Image and Video Processing

Test and Measurement

Computational Biology

Computational Finance

Simulink Product Family

Physical Modeling

Fixed-Point Modeling

Event-Based Modeling

Simulation Graphics

SIMULINK®

SIMULATION AND MODEL-BASED DESIGN

MATLAB Product Family

MATLAB®

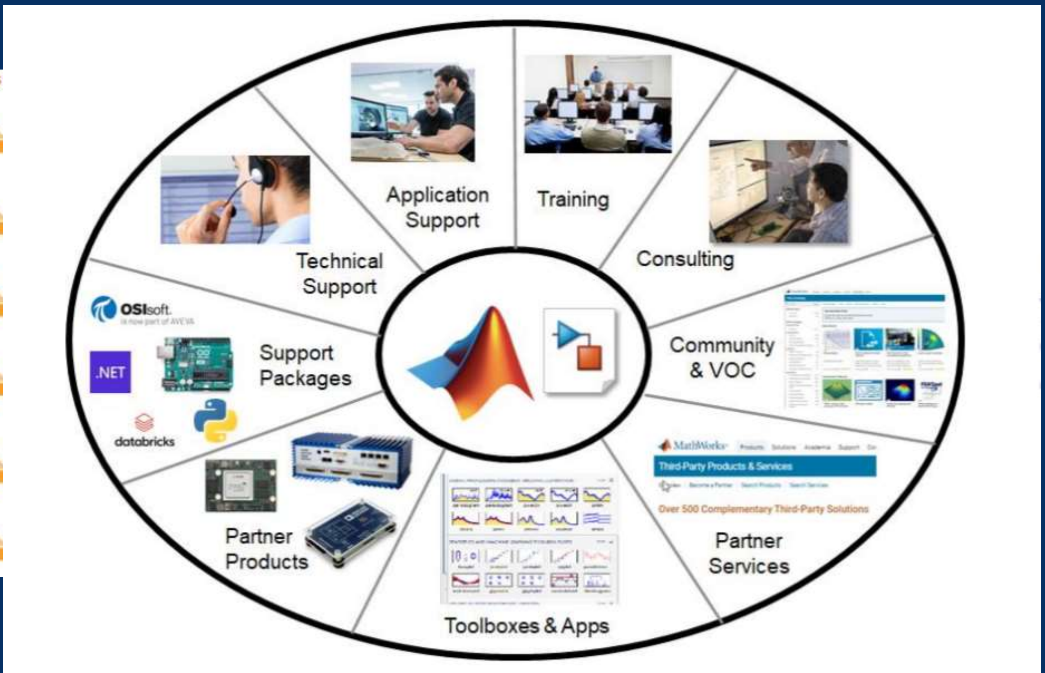
THE LANGUAGE OF TECHNICAL COMPUTING

Math and Optimization

Statistics and Data Analysis

Database Connectivity and Reporting

Application Deployment



Application Support

Training

Technical Support

Consulting

Support Packages

Community & VOC

Partner Products

Partner Services

Toolboxes & Apps

OSIsoft.
.NET
databricks
MathWorks
Third-Party Products & Services
Over 500 Complementary Third-Party Solutions

MathWorks®

How are Megatrends transforming Automotive R&D?

Electrification

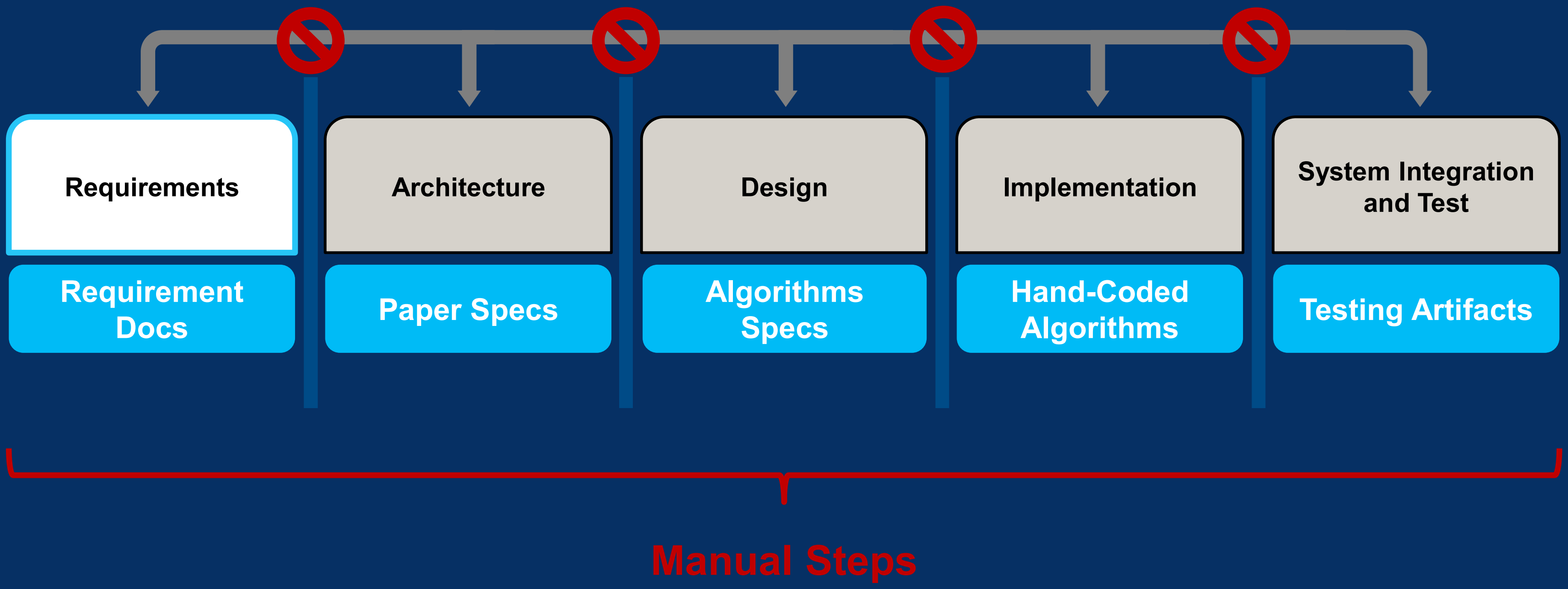
Automated Driving

Connected Vehicles

- Handling increasing system and software complexity
- Building innovative features with speed and quality
- Need for virtual development and test grounds

Why use Model Based Design for Embedded SW?

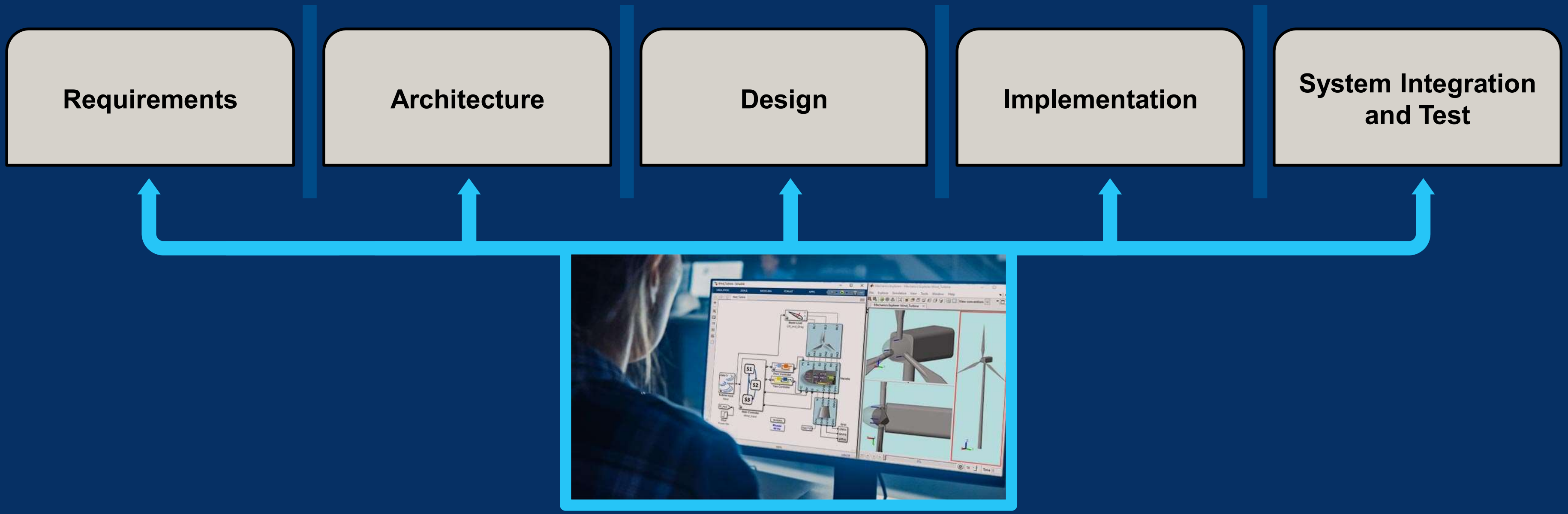
Requirements and artifacts are **hard** to **manage**, **change**, and **trace**



Manual steps introduce **errors** and **slow down** the development process

Requirements **capture** and artifact **traceability** throughout the process

Model-Based Design



Requirements

Architecture

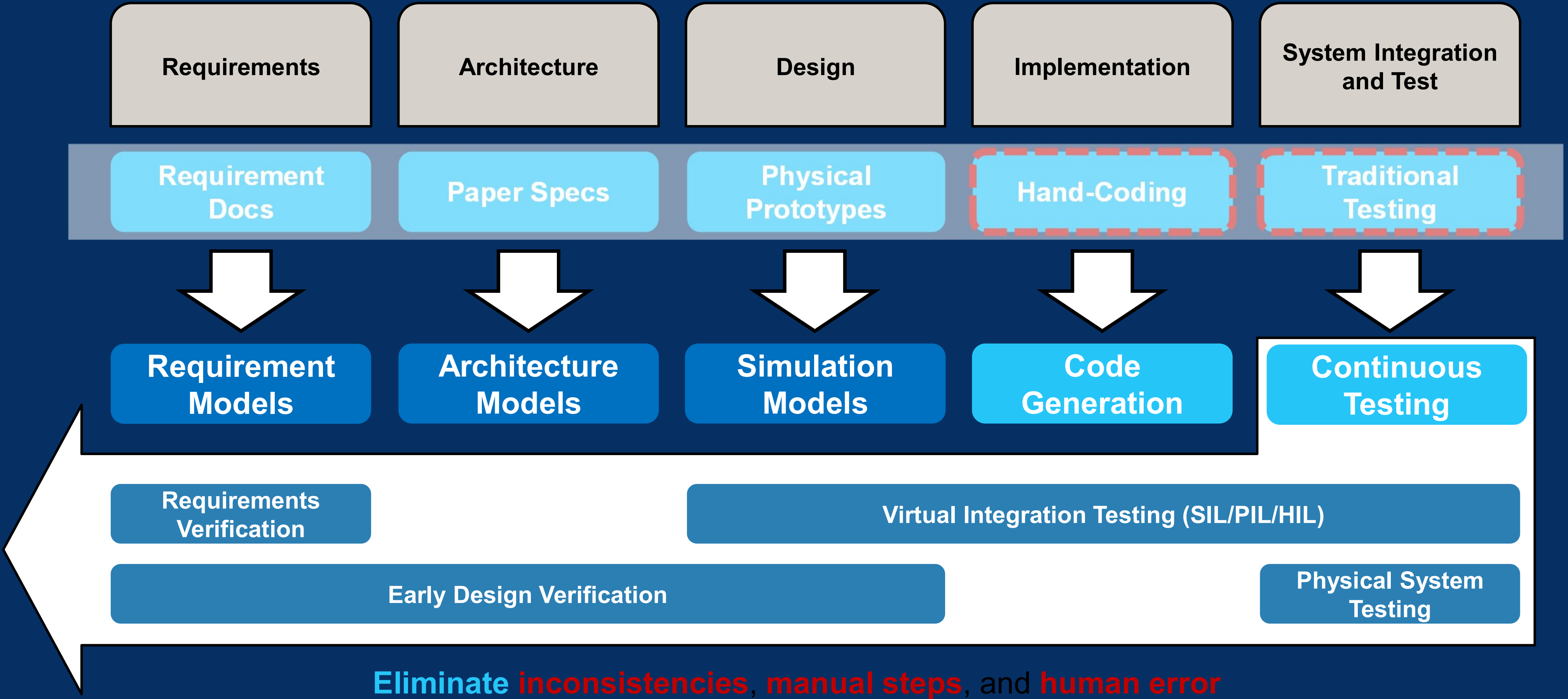
Design

Implementation

System Integration and Test

Models are at the **center** of your development process
Create a **digital thread**

Automate steps in the design process with **code generation** and **model-based verification**

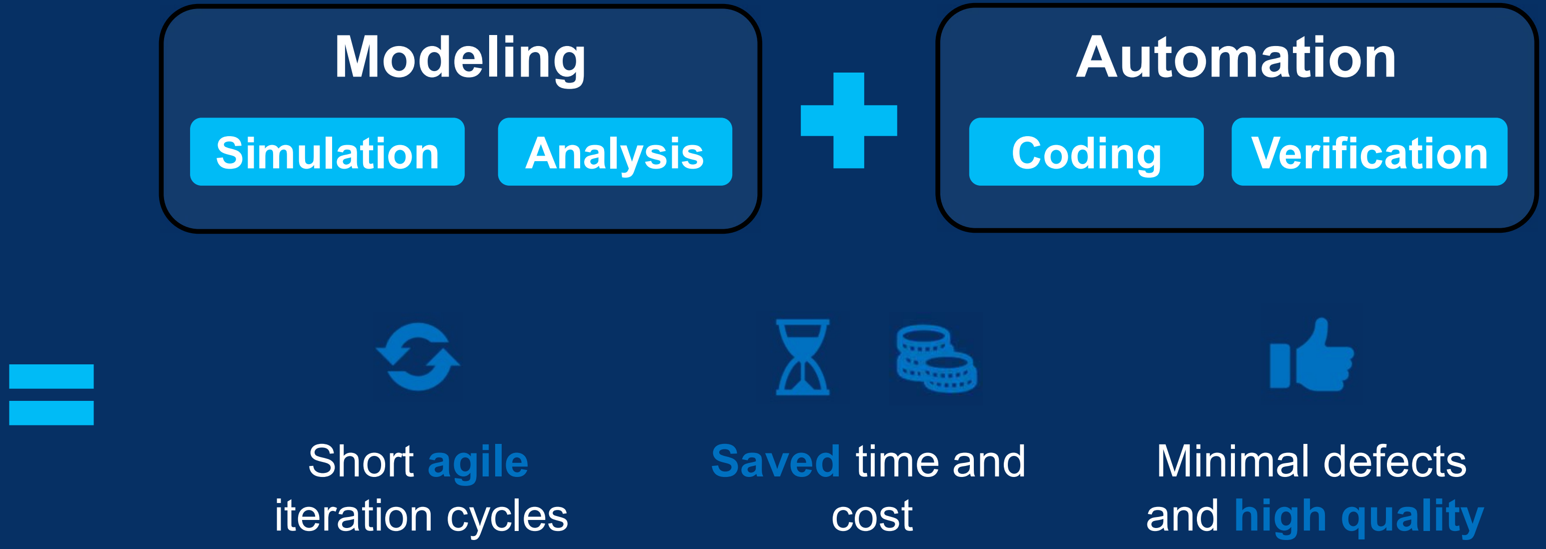


Eliminate inconsistencies, manual steps, and human error

So that you can **focus on** creating **innovative design** solutions

Model-Based Design

Systematic use of **models** throughout the development process



Modeling

Simulation **Analysis**

+

Automation

Coding **Verification**

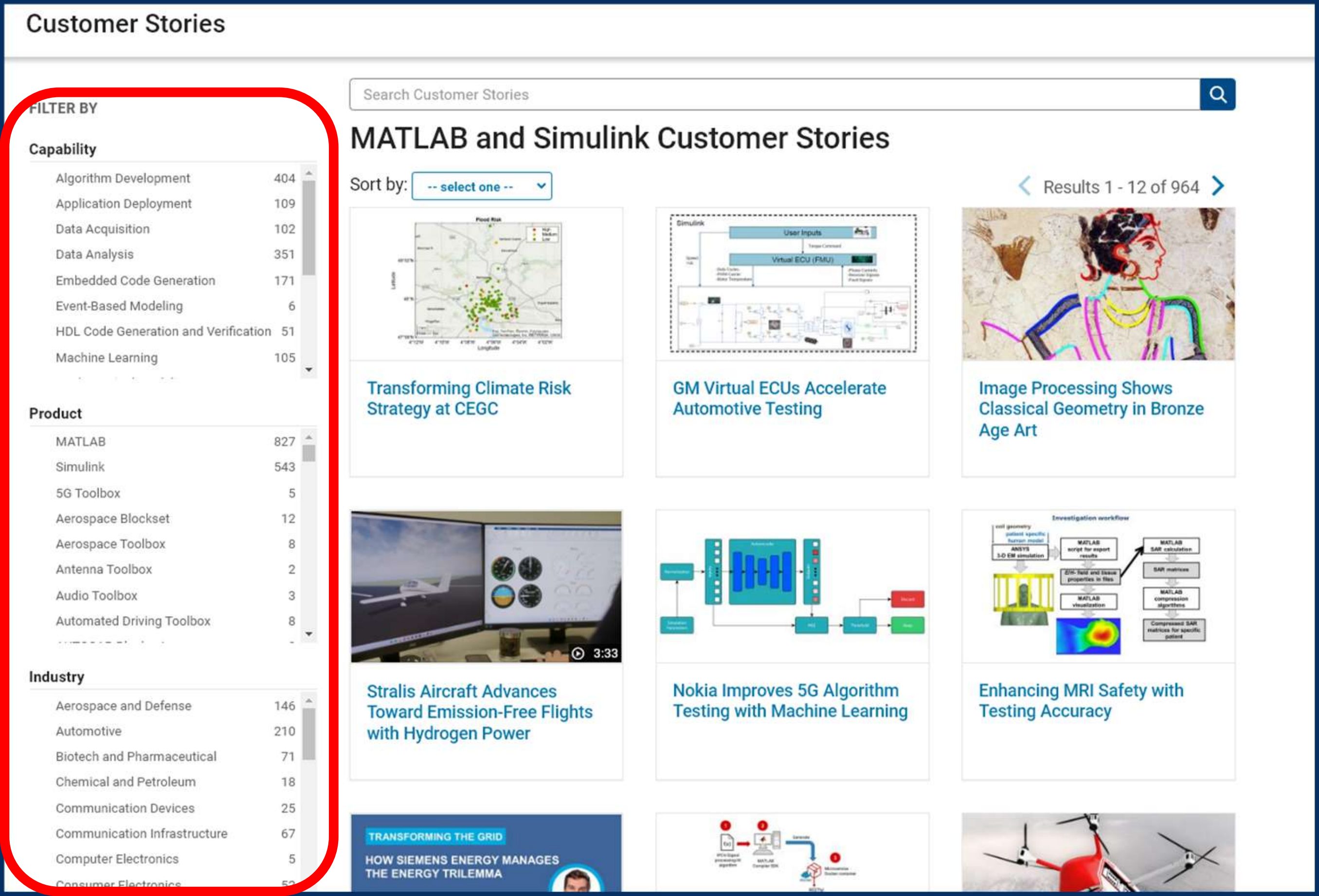
=

Short **agile** iteration cycles

Saved time and cost

Minimal defects and **high quality**

User Stories



Customer Stories

Search Customer Stories

FILTER BY

Capability

Algorithm Development	404
Application Deployment	109
Data Acquisition	102
Data Analysis	351
Embedded Code Generation	171
Event-Based Modeling	6
HDL Code Generation and Verification	51
Machine Learning	105

Product

MATLAB	827
Simulink	543
5G Toolbox	5
Aerospace Blockset	12
Aerospace Toolbox	8
Antenna Toolbox	2
Audio Toolbox	3
Automated Driving Toolbox	8

Industry

Aerospace and Defense	146
Automotive	210
Biotech and Pharmaceutical	71
Chemical and Petroleum	18
Communication Devices	25
Communication Infrastructure	67
Computer Electronics	5
Consumer Electronics	52

MATLAB and Simulink Customer Stories

Sort by: -- select one --

Results 1 - 12 of 964

Transforming Climate Risk Strategy at CEGC

GM Virtual ECUs Accelerate Automotive Testing

Image Processing Shows Classical Geometry in Bronze Age Art

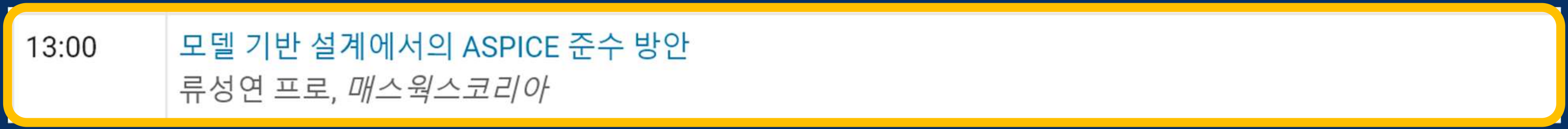
Stralis Aircraft Advances Toward Emission-Free Flights with Hydrogen Power

Nokia Improves 5G Algorithm Testing with Machine Learning

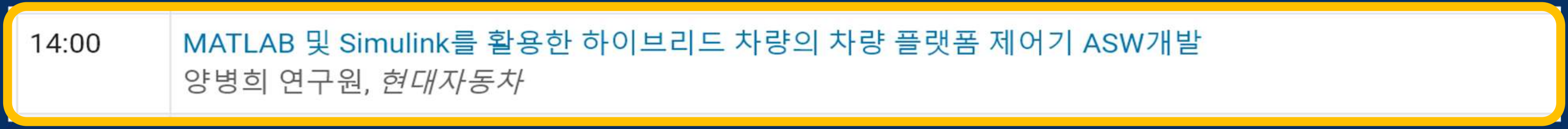
Enhancing MRI Safety with Testing Accuracy

https://www.mathworks.com/company/customer-stories/search.html?q=&page=1

More details coming :



13:00	모델 기반 설계에서의 ASPICE 준수 방안 류성연 프로, *매스웍스코리아*

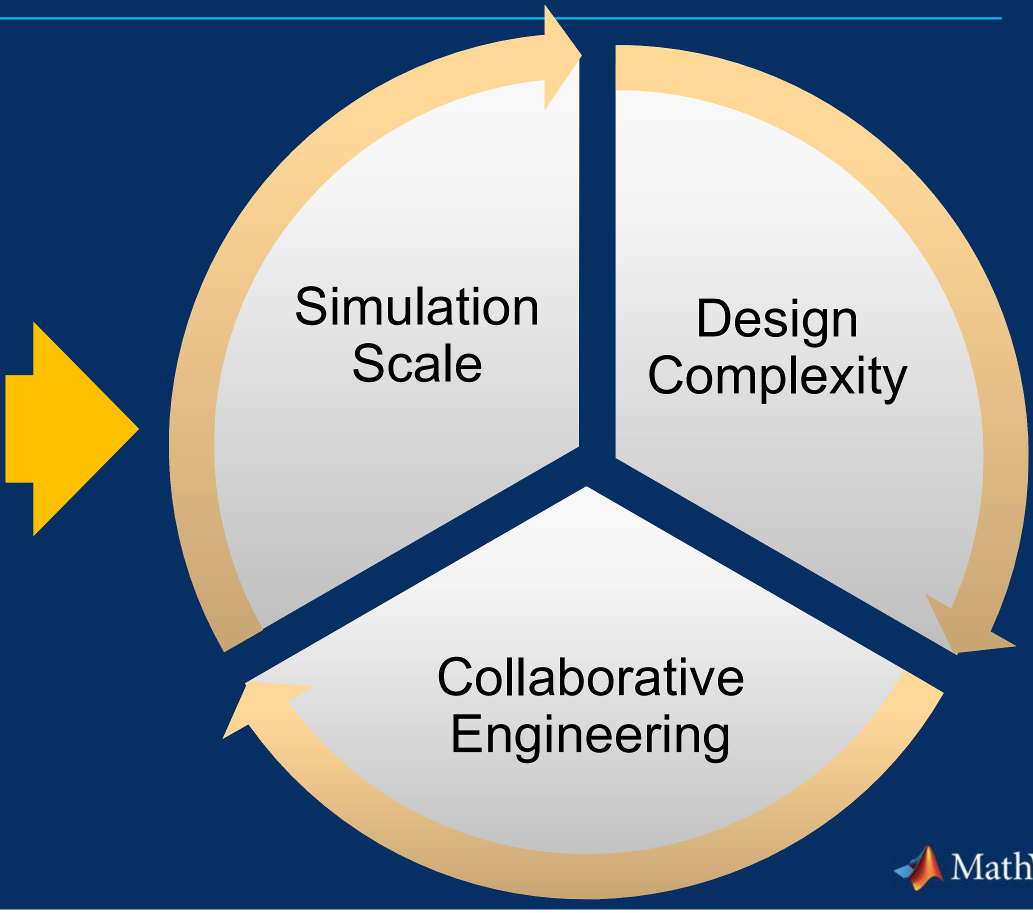


14:00	MATLAB 및 Simulink를 활용한 하이브리드 차량의 차량 플랫폼 제어기 ASW개발 양병희 연구원, *현대자동차*

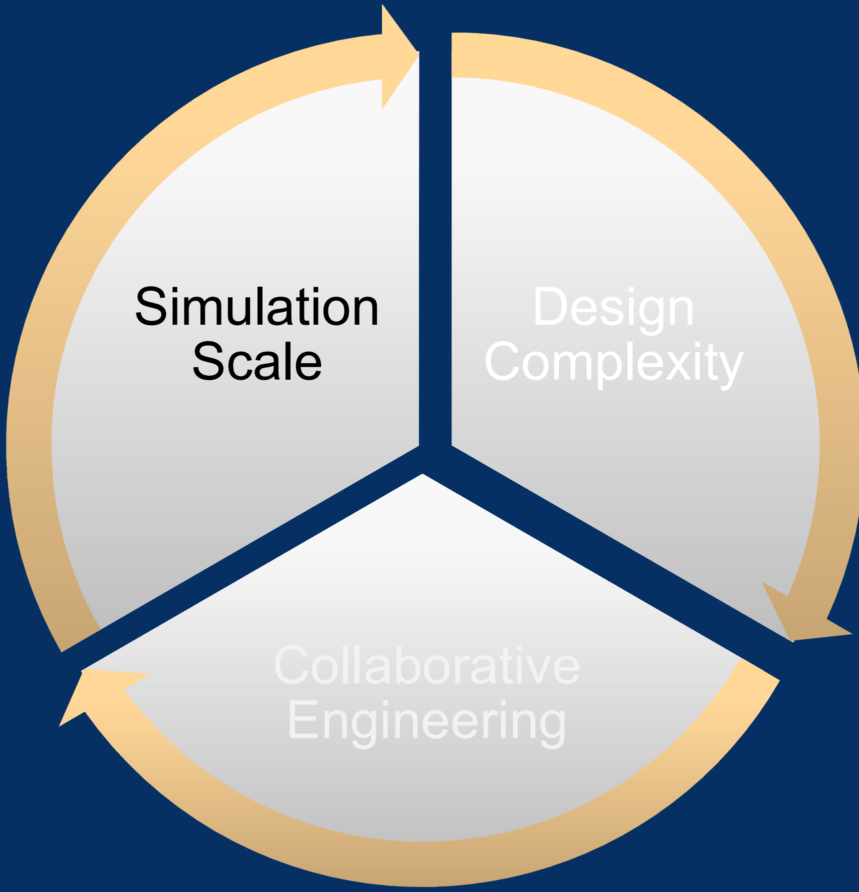
How is Model-Based Design Evolving?

The Three Evolutionary Forces at Play

- Handling increasing system and software complexity
- Building innovative features with speed and quality
- Need for virtual development and test grounds



The Three Evolutionary Forces at Play



Why are these trends important?

What are customers doing today about these trends?

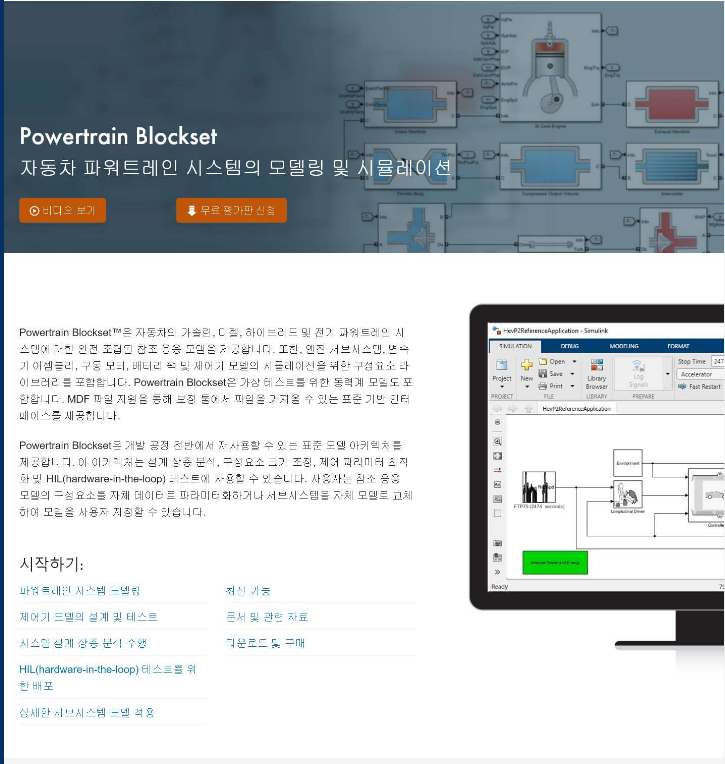
How does Model-Based Design evolve to meet the needs of future mobility?

Trend: Systems → Full Vehicle Simulation

Full Vehicle Simulation

2 Products for Vehicle Level Simulation Which Help You

Powertrain Blockset



Powertrain Blockset

자동차 파워트레인 시스템의 모델링 및 시뮬레이션

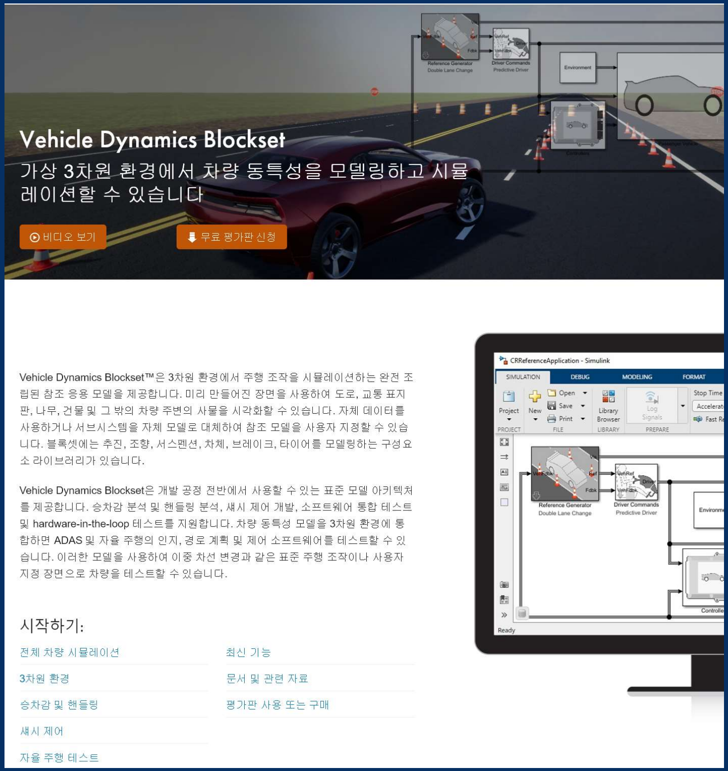
Powertrain Blockset™은 자동차의 가솔린, 디젤, 하이브리드 및 전기 파워트레인 시스템에 대한 완전 조립된 참조 응용 모델을 제공합니다. 또한, 엔진 서브시스템, 변속기 어셈블리, 구동 모터, 배터리 팩 및 제어기 모델의 시뮬레이션을 위한 구성요소 라이브러리를 포함합니다. Powertrain Blockset은 가상 테스트를 위한 동력계 모델도 포함합니다. MDF 파일 지원을 통해 보정 툴에서 파일을 가져올 수 있는 표준 기반 인터페이스를 제공합니다.

Powertrain Blockset은 개발 공정 전반에서 재사용할 수 있는 표준 모델 아키텍처를 제공합니다. 이 아키텍처는 설계 상충 분석, 구성요소 크기 조정, 제어 파라미터 최적화 및 HIL(hardware-in-the-loop) 테스트에 사용할 수 있습니다. 사용자는 참조 응용 모델의 구성요소를 자체 데이터로 파라미터화하거나 서브시스템을 자체 모델로 교체하여 모델을 사용자 지정할 수 있습니다.

시작하기:

- 파워트레인 시스템 모델링
- 제어기 모델의 설계 및 테스트
- 시스템 설계 상충 분석 수행
- HIL(hardware-in-the-loop) 테스트를 위한 배포
- 상세한 서브시스템 모델 적용
- 최신 기능
- 문서 및 관련 자료
- 다운로드 및 구매

Vehicle Dynamics Blockset



Vehicle Dynamics Blockset

가상 3차원 환경에서 차량 동특성을 모델링하고 시뮬레이션할 수 있습니다

Vehicle Dynamics Blockset™은 3차원 환경에서 주행 조작을 시뮬레이션하는 완전 조립된 참조 응용 모델을 제공합니다. 미리 만들어진 장면을 사용하여 도로, 교통 표지판, 나무, 건물 및 그 밖의 차량 주변의 사물을 시각화할 수 있습니다. 자체 데이터를 사용하거나 서브시스템을 자체 모델로 대체하여 참조 모델을 사용자 지정할 수 있습니다. 블록셋에는 추진, 조향, 서스펜션, 차체, 브레이크, 타이어를 모델링하는 구성요소 라이브러리가 있습니다.

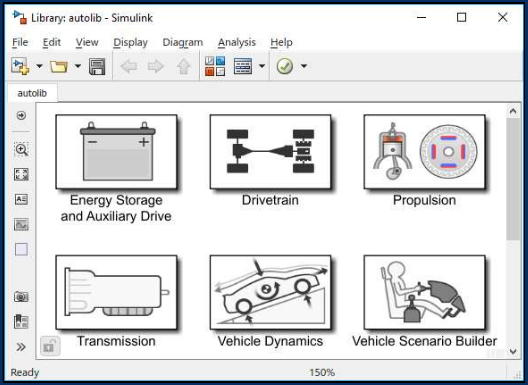
Vehicle Dynamics Blockset은 개발 공정 전반에서 사용할 수 있는 표준 모델 아키텍처를 제공합니다. 승차감 분석 및 핸들링 분석, 섀시 제어 개발, 소프트웨어 통합 테스트 및 hardware-in-the-loop 테스트를 지원합니다. 차량 동특성 모델을 3차원 환경에 통합하면 ADAS 및 자율 주행의 인지, 경로 계획 및 제어 소프트웨어를 테스트할 수 있습니다. 이러한 모델을 사용하여 이중 차선 변경과 같은 표준 주행 조작이나 사용자 지정 장면으로 차량을 테스트할 수 있습니다.

시작하기:

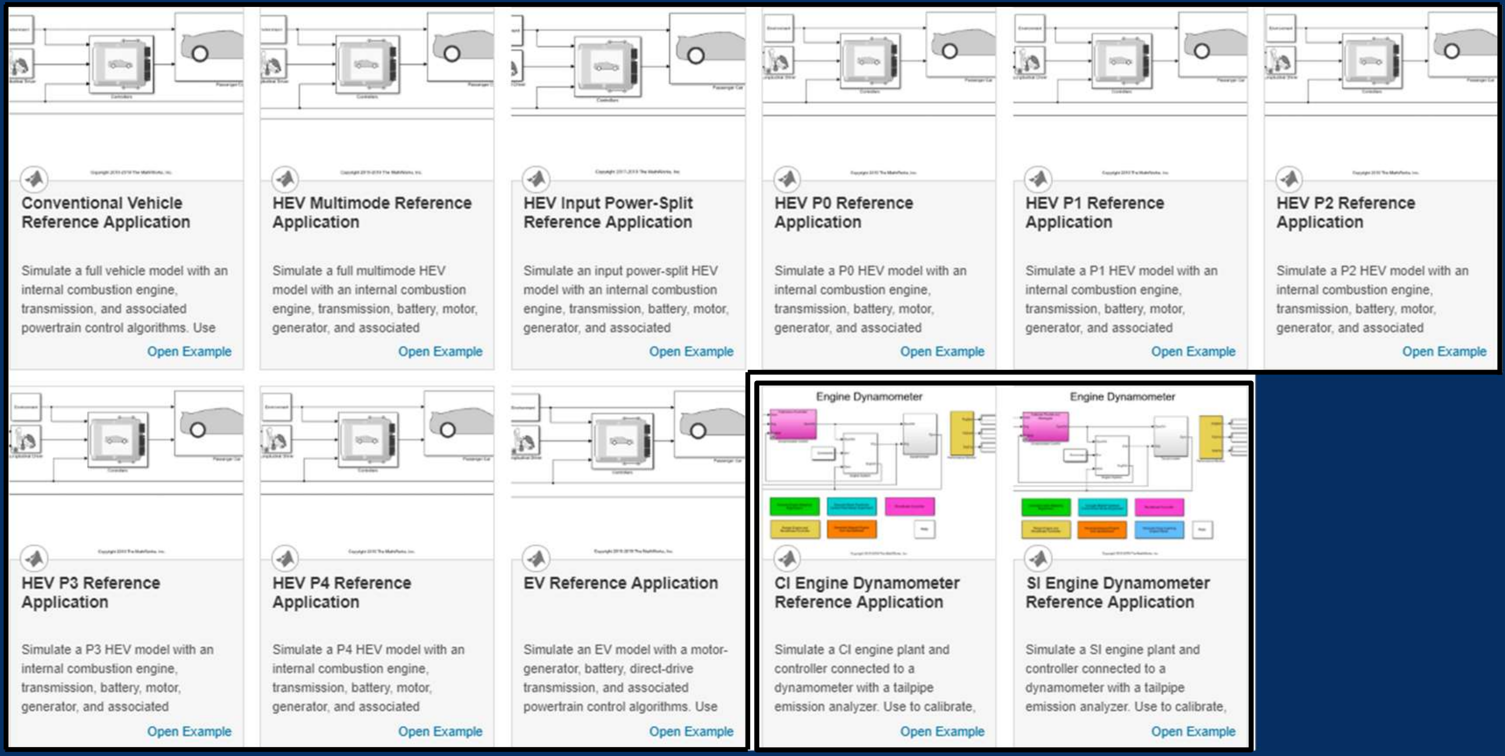
- 전체 차량 시뮬레이션
- 3차원 환경
- 승차감 및 핸들링
- 섀시 제어
- 자율 주행 테스트
- 최신 기능
- 문서 및 관련 자료
- 평가판 사용 또는 구매

Composition of Powertrain Blockset & Vehicle Dynamics Blockset

Library of blocks



Reference Applications



Conventional Vehicle Reference Application

Simulate a full vehicle model with an internal combustion engine, transmission, and associated powertrain control algorithms. Use

Open Example

HEV Multimode Reference Application

Simulate a full multimode HEV model with an internal combustion engine, transmission, battery, motor, generator, and associated

Open Example

HEV Input Power-Split Reference Application

Simulate an input power-split HEV model with an internal combustion engine, transmission, battery, motor, generator, and associated

Open Example

HEV P0 Reference Application

Simulate a P0 HEV model with an internal combustion engine, transmission, battery, motor, generator, and associated

Open Example

HEV P1 Reference Application

Simulate a P1 HEV model with an internal combustion engine, transmission, battery, motor, generator, and associated

Open Example

HEV P2 Reference Application

Simulate a P2 HEV model with an internal combustion engine, transmission, battery, motor, generator, and associated

Open Example

HEV P3 Reference Application

Simulate a P3 HEV model with an internal combustion engine, transmission, battery, motor, generator, and associated

Open Example

HEV P4 Reference Application

Simulate a P4 HEV model with an internal combustion engine, transmission, battery, motor, generator, and associated

Open Example

EV Reference Application

Simulate an EV model with a motor-generator, battery, direct-drive transmission, and associated powertrain control algorithms. Use

Open Example

CI Engine Dynamometer Reference Application

Simulate a CI engine plant and controller connected to a dynamometer with a tailpipe emission analyzer. Use to calibrate,

Open Example

SI Engine Dynamometer Reference Application

Simulate a SI engine plant and controller connected to a dynamometer with a tailpipe emission analyzer. Use to calibrate,

Open Example

Virtual Vehicle Models   Virtual Engine Dynamometers

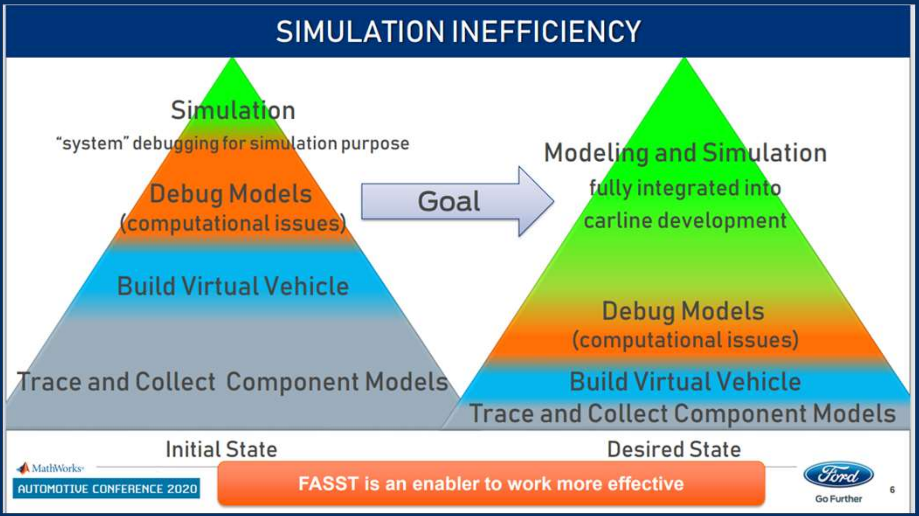
FORD uses an automated system simulation toolchain to build a virtual vehicle in minutes and Detect System Issues Early in Development

Ford & MathWorks collaborated on a standard framework Ford Automated System Simulation Toolchain (FASST) which has 500+ users today

FASST

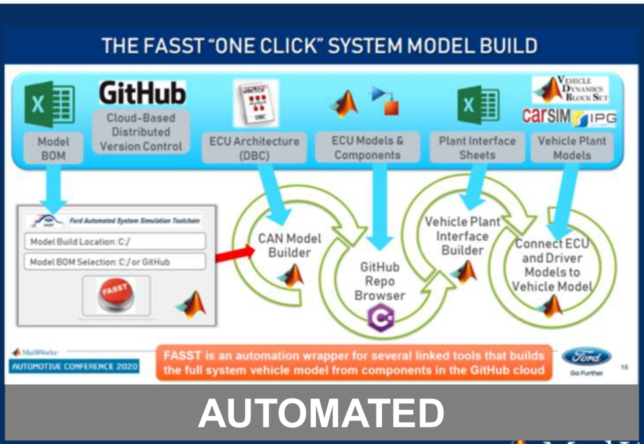
- reduced virtual vehicle build time from months to minutes
- enabled groups needing to perform different analysis tasks to build their own virtual vehicles

Model-Based Agility with Ford Automated System Simulation Toolchain (FASST) - MATLAB & Simulink



SIMULATION INEFFICIENCY

FASST is an enabler to work more effective



THE FASST "ONE CLICK" SYSTEM MODEL BUILD

FASST is an automation wrapper for several linked tools that builds the full system vehicle model from components in the GitHub cloud

Tesla Maximizes Efficiency with Model-Based Design

“[Tesla] spent the past 10 years building MATLAB models of where all the energy is flowing. From that, it's determined where the vehicles experience losses due to inefficiencies. The team then goes in and continually tweaks the hardware to increase efficiency. Additional secondary improvements can be pushed via over-the-air updates.”
– Roberto Baldwin, Car and Driver

SCAN ME

Tesla Tells Us How It Keeps Beating Nearly Everyone in the Range Game
https://www.caranddriver.com/news/a34046953/tesla-range-strategy-details/

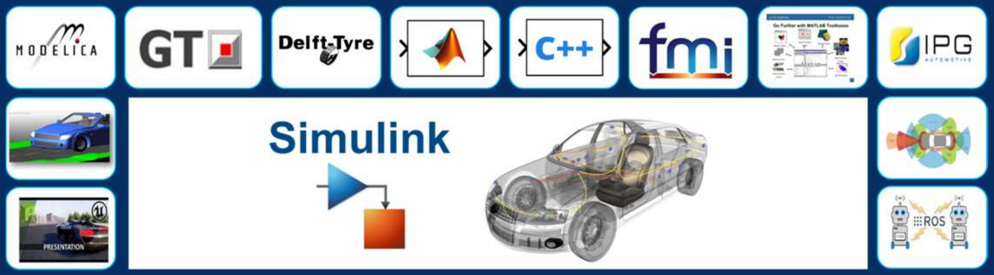
MathWorks Vision for Virtual Vehicle

Every function designer can ***create a virtual vehicle within minutes*** with desired details in physics and software, and prototype, calibrate, and validate their functions in simulation

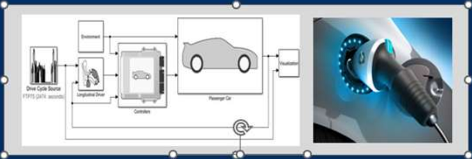
Goals

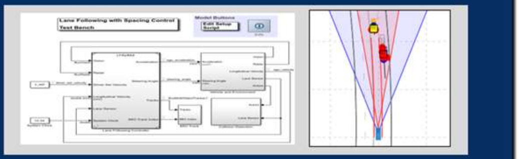
- Maximize frontloading via simulation
- Deliver rich out-of-box capabilities and openness for tailoring
- Provide world class simulation integration platform (SIP)

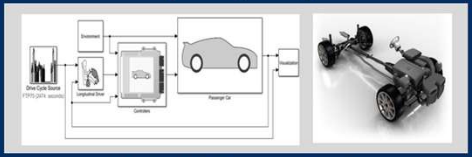
How is Model-Based Design Evolving?

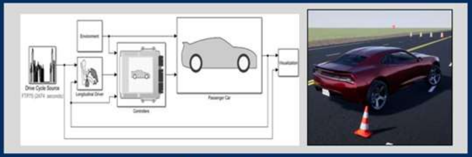


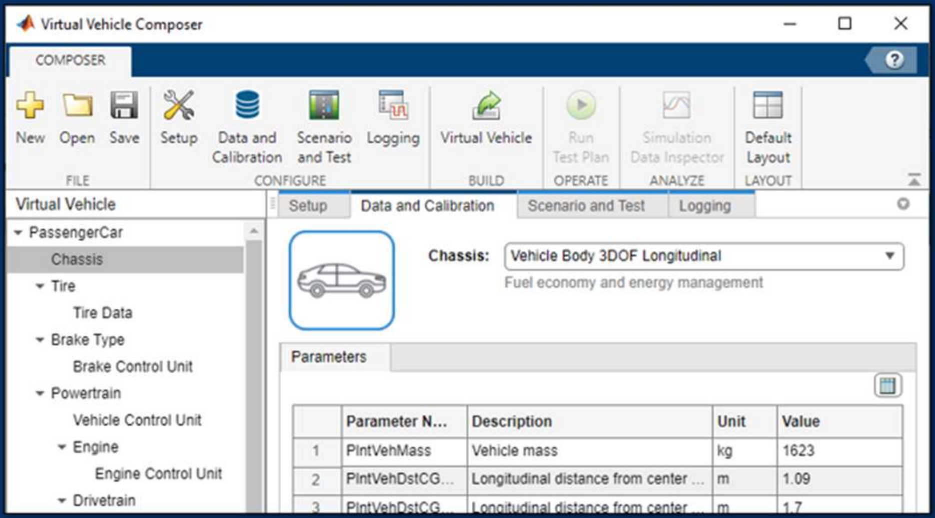
Automotive Reference Applications


Pure EV


Lane Keeping Assist


Hybrid Powertrain


Car Vehicle Dynamics

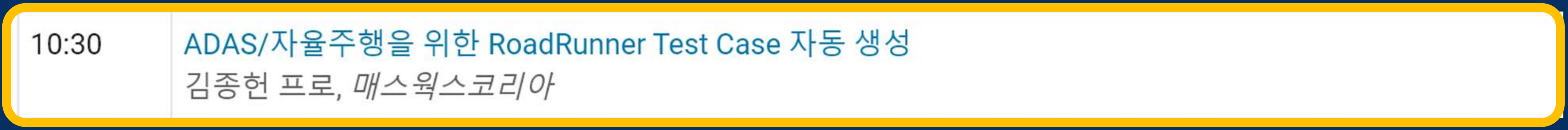


Chassis: Vehicle Body 3DOF Longitudinal

Fuel economy and energy management

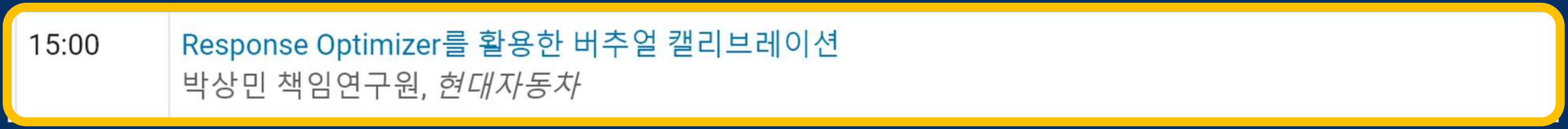
	Parameter N...	Description	Unit	Value
1	PlntVehMass	Vehicle mass	kg	1623
2	PlntVehDstCG...	Longitudinal distance from center ...	m	1.09
3	PlntVehDstCG...	Longitudinal distance from center ...	m	1.7

More details coming :



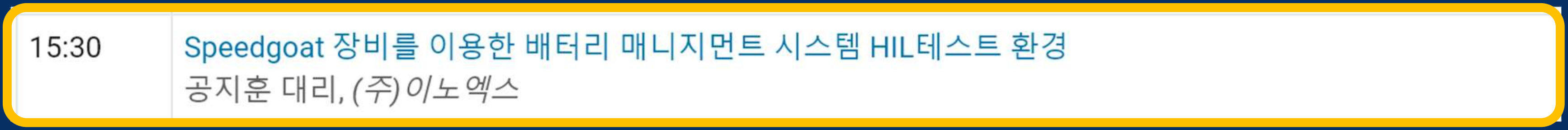
10:30 ADAS/자율주행을 위한 RoadRunner Test Case 자동 생성

김종헌 프로, *매스웍스코리아*



15:00 Response Optimizer를 활용한 버추얼 캘리브레이션

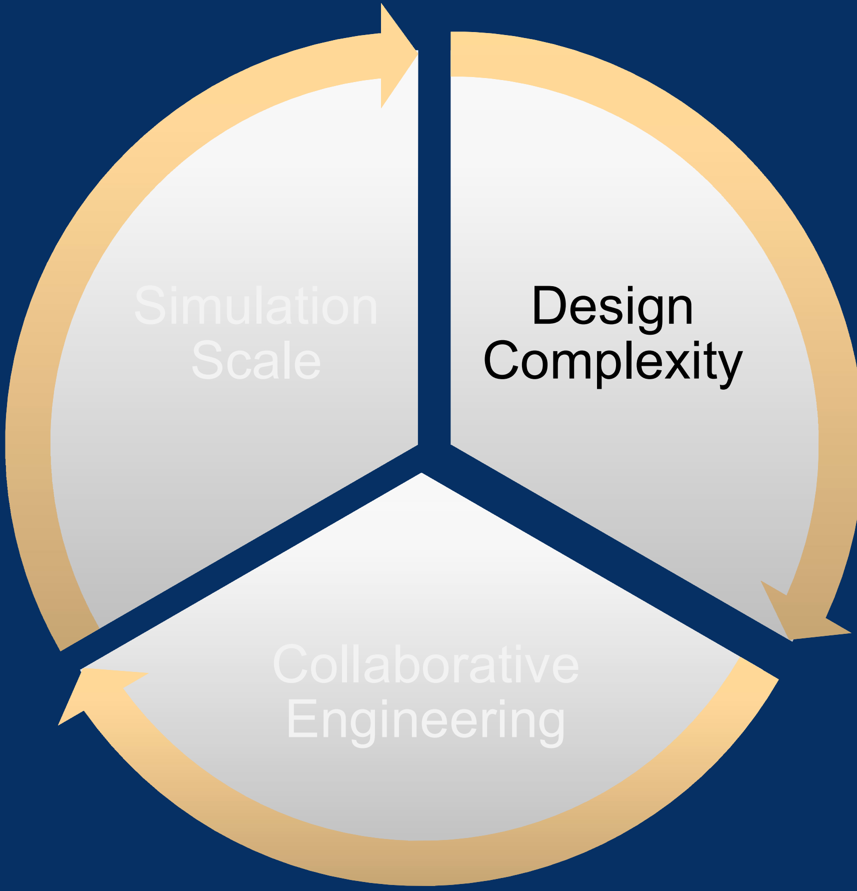
박상민 책임연구원, *현대자동차*



15:30 Speedgoat 장비를 이용한 배터리 매니지먼트 시스템 HIL테스트 환경

공지훈 대리, *(주)이노엑스*

The Three Evolutionary Forces at Play



Why are these trends important?

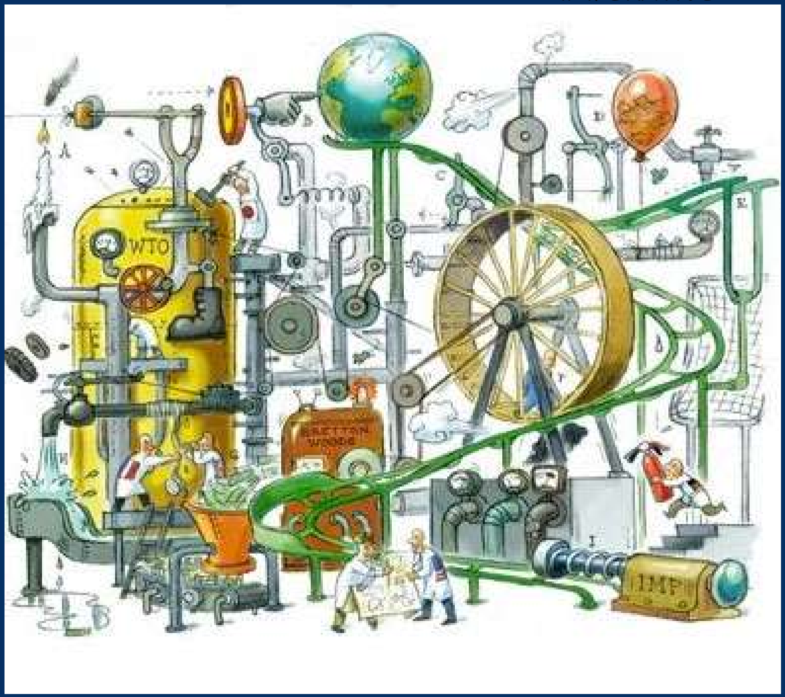
What are customers doing today about these trends?

How does Model-Based Design evolve to meet the needs of future mobility?

System Engineering – Why?

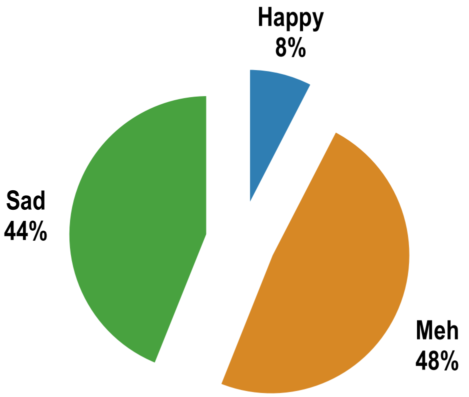
MODEL-BASED SYSTEM ENGINEERING IMPERATIVE

Systems Engineering is about **coping with complexity**. Systems Engineering helps **avoid omissions and invalid assumptions**, helps to manage real world changing issues, and produce the most **efficient, economic and robust solution**.



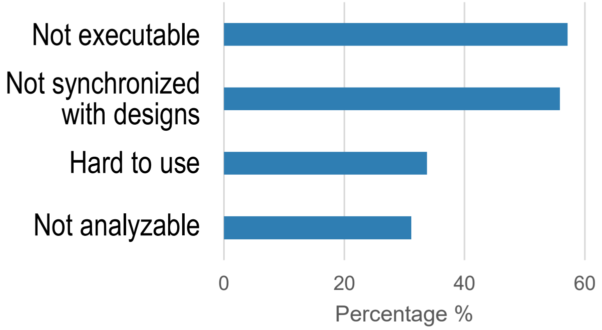
What we’ve heard from *you*

Are you happy with your current tool choices for modeling system architecture?



Survey: MathWorks Advisory Board

What are the problems with your current system architecture solution?

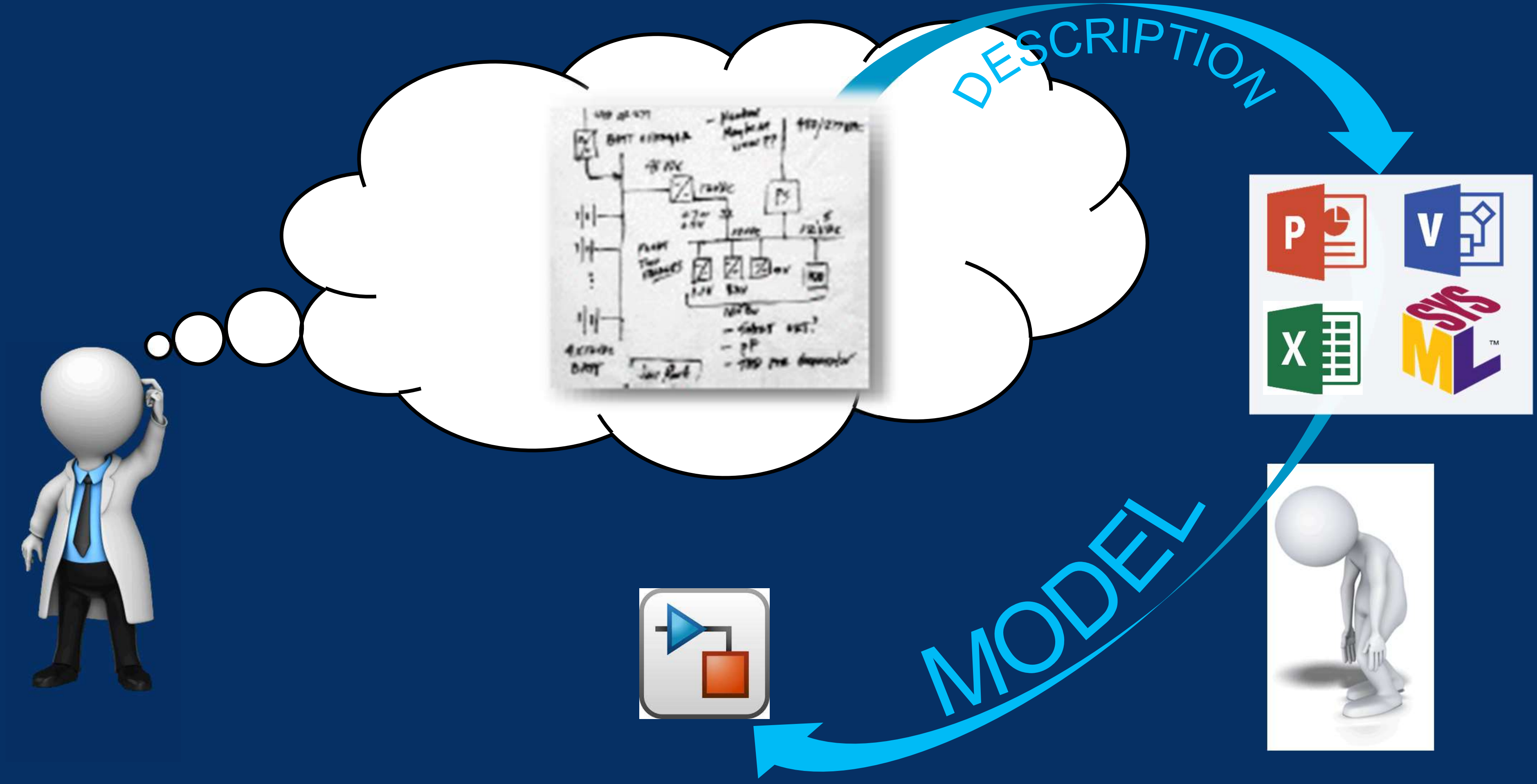


Survey: MathWorks Advisory Board

Customer:
“We have tried to build the architecture model in SysML and connect it to the design in Simulink …

*… [It] does not work without **rework** both in the **architecture** and **design** worlds whenever a change is needed. It is broken and we need a more integrated approach”*

What problem are we trying to address?

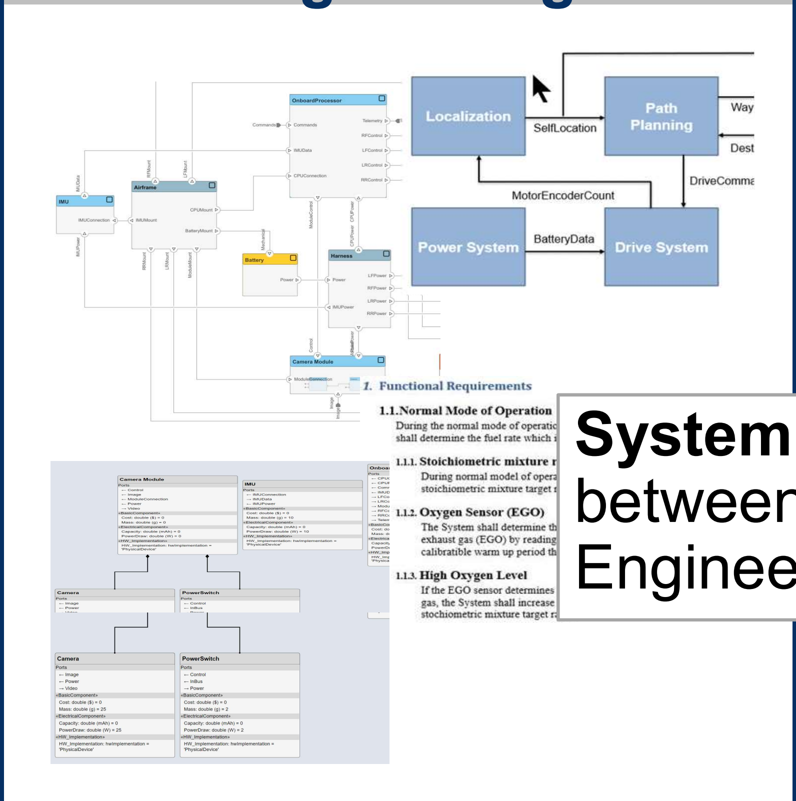


DESCRIPTION

MODEL

Trend : Bridging the gap between Model-Based Systems Engineering and Model-Based Design

Model-Based Systems Engineering



1. Functional Requirements

1.1. Normal Mode of Operation

During the normal mode of operatio
shall determine the fuel rate which i

1.1.1. Stoichiometric mixture r

During normal model of opera
stoichiometric mixture target r

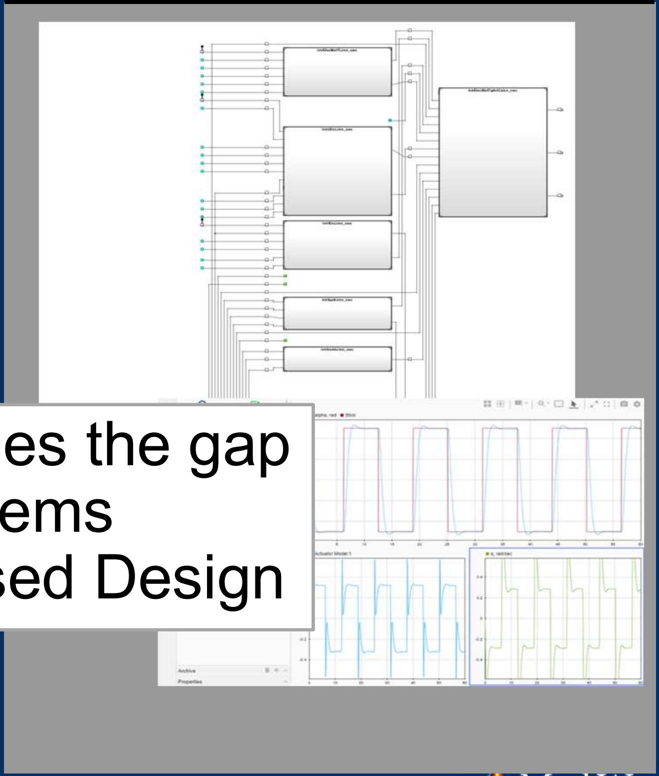
1.1.2. Oxygen Sensor (EGO)

The System shall determine th
exhaust gas (EGO) by reading
calibratible warm up period th

1.1.3. High Oxygen Level

If the EGO sensor determines
gas, the System shall increase
stochiometric mixture target ra

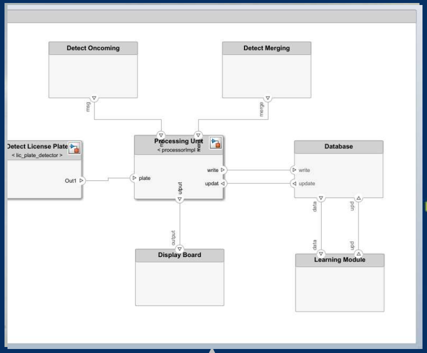
Model-Based Design



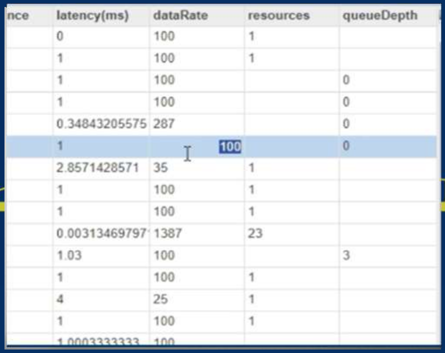
System Composer™ bridges the gap between Model-Based Systems Engineering and Model-Based Design

How is Model-Based Design Evolving to support the needs of System Engineers?

Be Intuitive

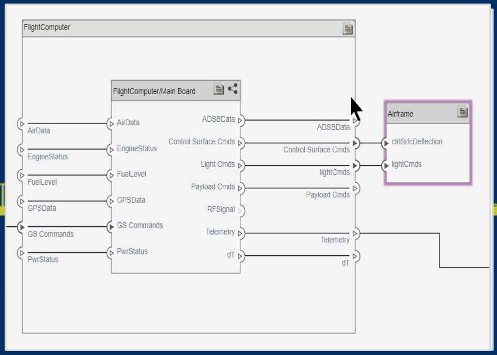


Facilitate Analysis

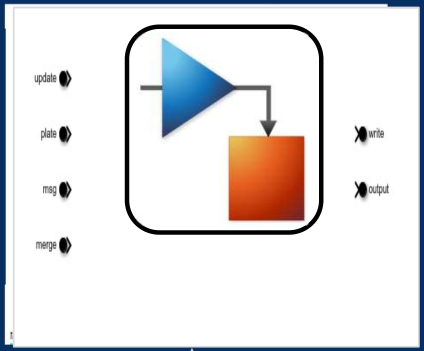


nce	latency(ms)	dataRate	resources	queueDepth
	0	100	1	
	1	100	1	
	1	100		0
	1	100		0
	0.34843205575	287		0
	1	100		0
	2.8571428571	35	1	
	1	100	1	
	1	100	1	
	0.00313469797	1387	23	
	1.03	100		3
	1	100	1	
	4	25	1	
	1	100	1	
	1.0003333333	100		

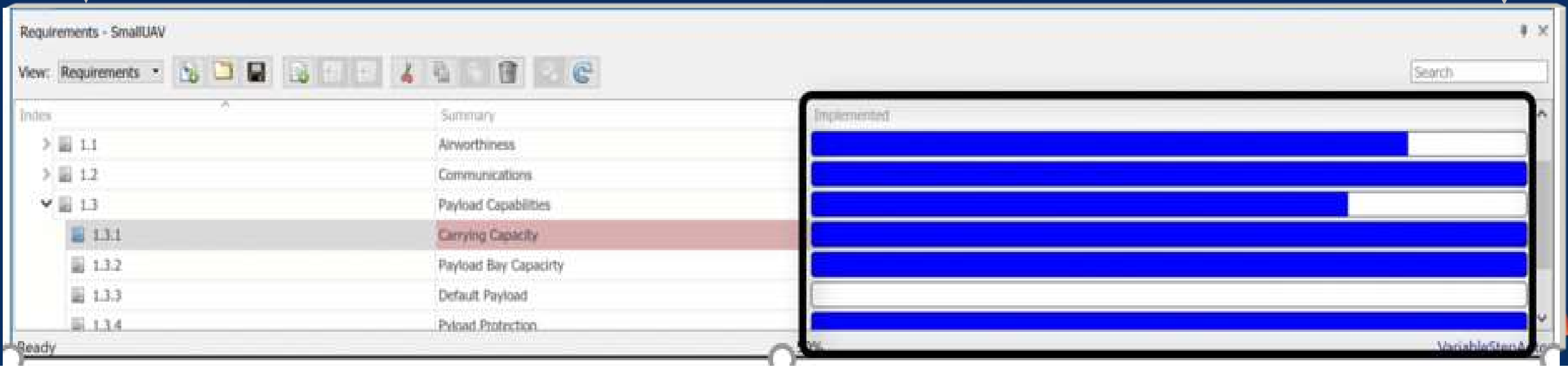
Tackle Complexity



Enable Implementation



Requirements Coverage Reporting and Impact Analysis



Index	Summary	Implemented
1.1	Airworthiness	
1.2	Communications	
1.3	Payload Capabilities	
1.3.1	Carrying Capacity	
1.3.2	Payload Bay Capacirty	
1.3.3	Default Payload	
1.3.4	Pyload Protection	

Mercedes-Benz : System Architecture using System Composer

Challenge :

- What’s missing here is that the developer won’t understand the whole picture of the requirement being implemented on how it is affecting the entire system

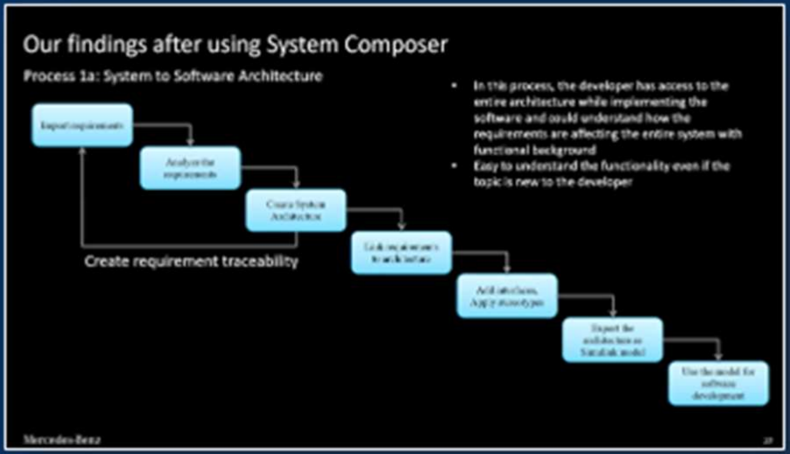
- Might miss implementing some of the requirements due to human error

Solution : With System composer, this void could be addressed, as it is a MATLAB toolbox, the conventional process is not altered much

Results :

- Communication between stakeholders and functional developers will be improved
- The tool is assistive in complex system development

System Architecture creation using System Composer



Our findings after using System Composer

Process 1a: System to Software Architecture

- In this process, the developer has access to the entire architecture while implementing the software and could understand how the requirements are affecting the entire system with functional background
- Easy to understand the functionality even if the topic is new to the developer

Import requirements

Analyze the requirements

Create System Architecture

Link requirements to architecture

Add interfaces, Apply stereotypes

Export the architecture as Simulink model

Use the model for software development

Create requirement traceability

Mercedes-Benz

https://www.mathworks.com/videos/system-architecture-creation-using-system-composer-1622075047576.html?s_tid=srchtitle_site_search_1_benz%20system%20composer

Delphi Technologies : AUTOSAR Architecture Modeling of Multi-core Electric Powertrain Controller for Next Generation Inverter

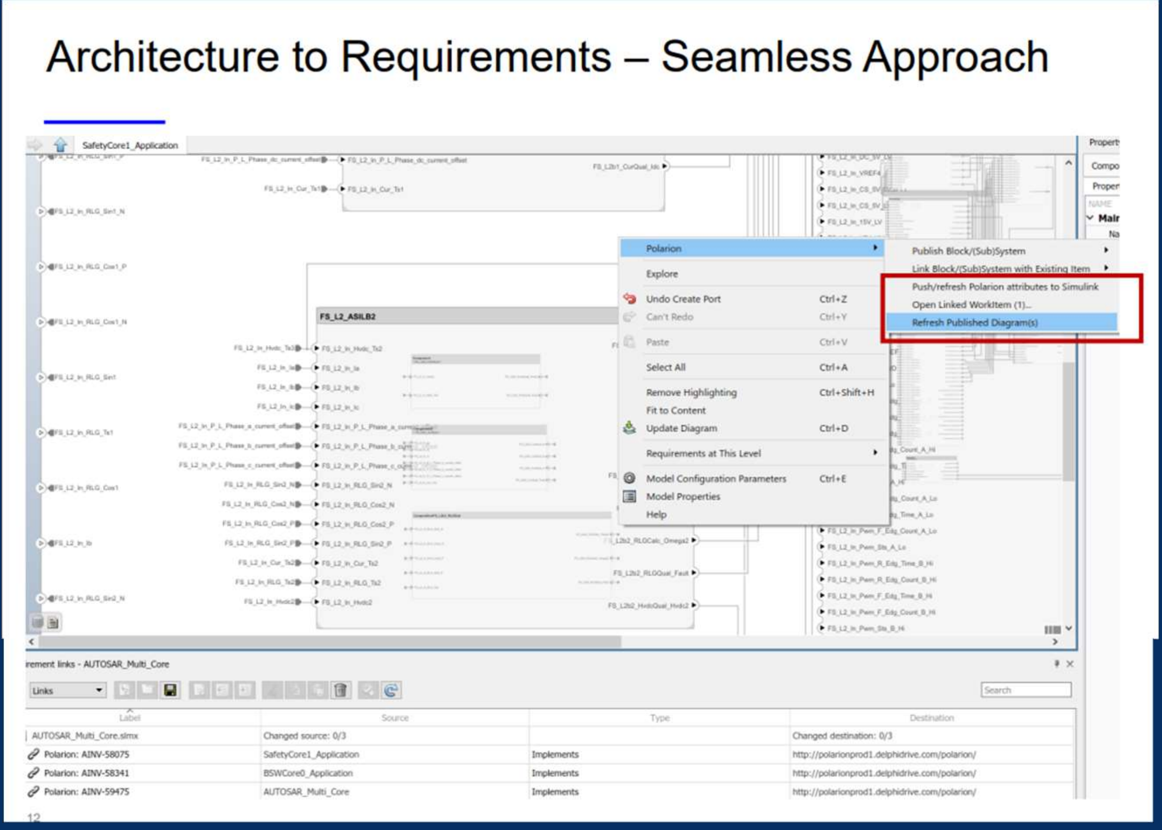
Challenge :

- Gap between architecture and design models
- Gaps in requirement traceability
- Lack of support for intuitive and performance analysis

Solution : Delphi Technologies used System Composer and AUTOSAR Blockset for AUTOSAR Based System Engineering

Results :

- Architecture to Requirements –Seamless Approach
- Intuitive and Performance Analysis



Architecture to Requirements – Seamless Approach

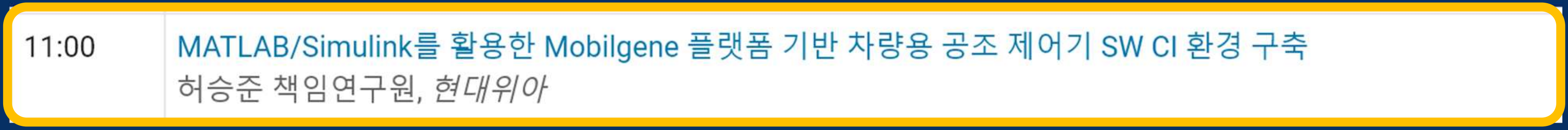
SafetyCore1_Application
FS_L2_ASILB2
Polarion
Explore
Undo Create Port Ctrl+Z
Can't Redo Ctrl+Y
Paste Ctrl+V
Select All Ctrl+A
Remove Highlighting Ctrl+Shift+H
Fit to Content
Update Diagram Ctrl+D
Requirements at This Level
Model Configuration Parameters Ctrl+E
Model Properties
Help
Publish Block/(Sub)System
Link Block/(Sub)System with Existing Item
Push/refresh Polarion attributes to Simulink
Open Linked WorkItem (1)...
Refresh Published Diagram(s)
rement links - AUTOSAR_Multi_Core
Links
Search

Label	Source	Type	Destination
AUTOSAR_Multi_Core.slmx	Changed source: 0/3		Changed destination: 0/3
Polarion: AINV-58075	SafetyCore1_Application	Implements	http://polarionprod1.delphidrive.com/polarion/
Polarion: AINV-58341	BSWCore0_Application	Implements	http://polarionprod1.delphidrive.com/polarion/
Polarion: AINV-59475	AUTOSAR_Multi_Core	Implements	http://polarionprod1.delphidrive.com/polarion/

12

https://www.mathworks.com/content/dam/mathworks/mathworks-dot-com/company/events/conferences/automotive-conference-stuttgart/2020/autosar-software-architecture-modeling-of-multicore-electric-powertrain-software.pdf

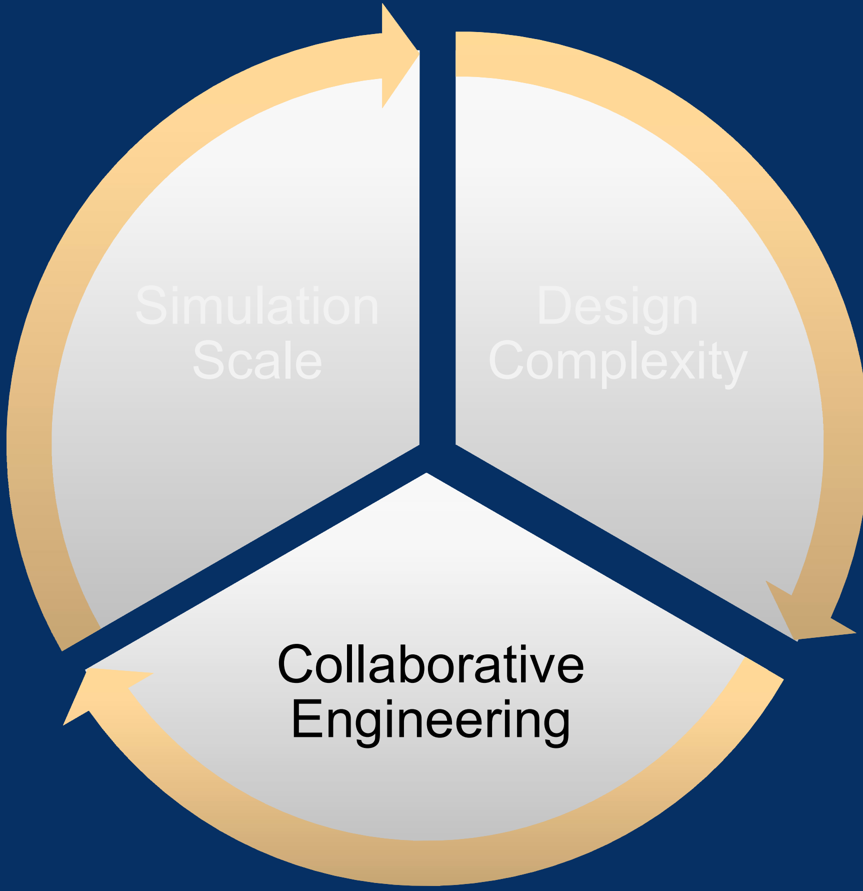
More details coming :



11:00 MATLAB/Simulink를 활용한 Mobilgene 플랫폼 기반 차량용 공조 제어기 SW CI 환경 구축

허승준 책임연구원, *현대위아*

The Three Evolutionary Forces at Play

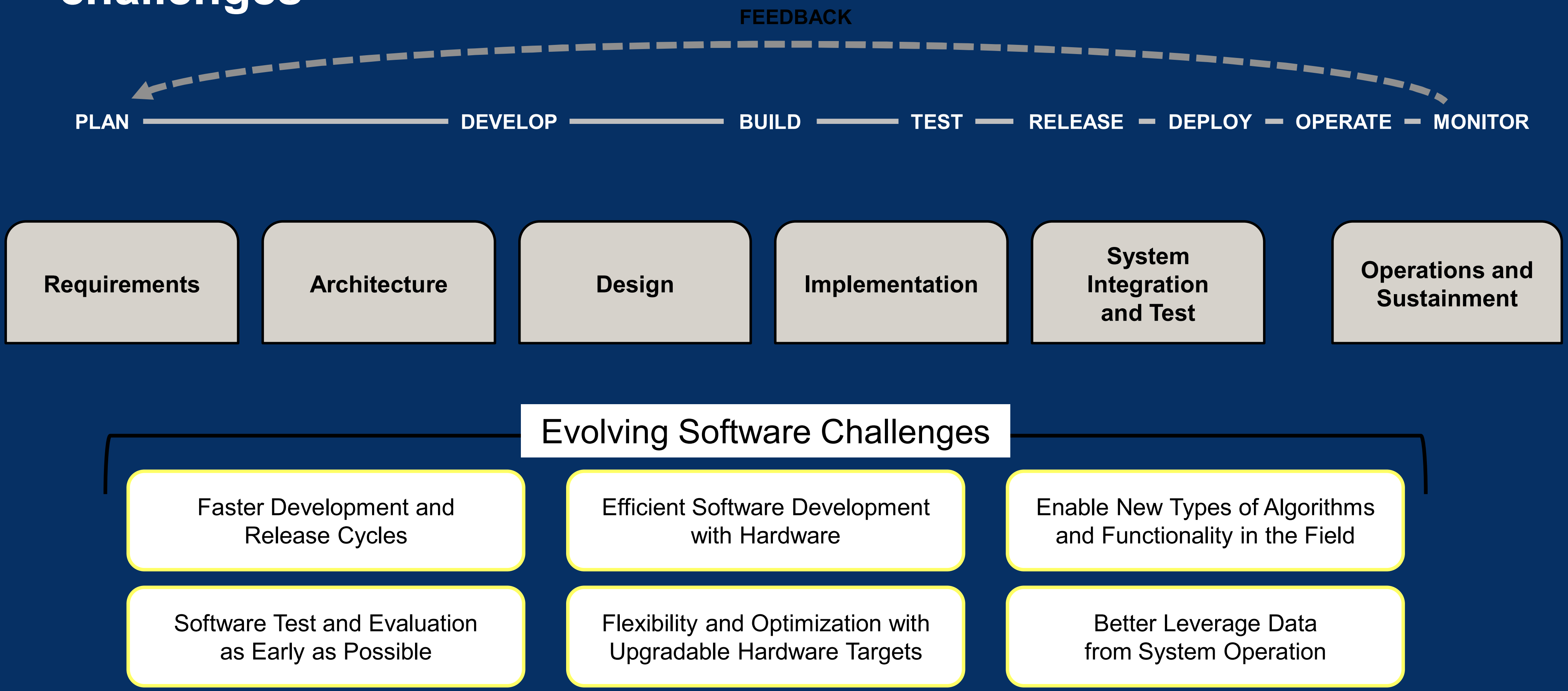


Why are these trends important?

What are customers doing today about these trends?

How does Model-Based Design evolve to meet the needs of future mobility?

Adapting development process under evolving software challenges



FEEDBACK

PLAN — DEVELOP — BUILD — TEST — RELEASE — DEPLOY — OPERATE — MONITOR

Requirements

Architecture

Design

Implementation

System Integration and Test

Operations and Sustainment

Evolving Software Challenges

Faster Development and Release Cycles

Efficient Software Development with Hardware

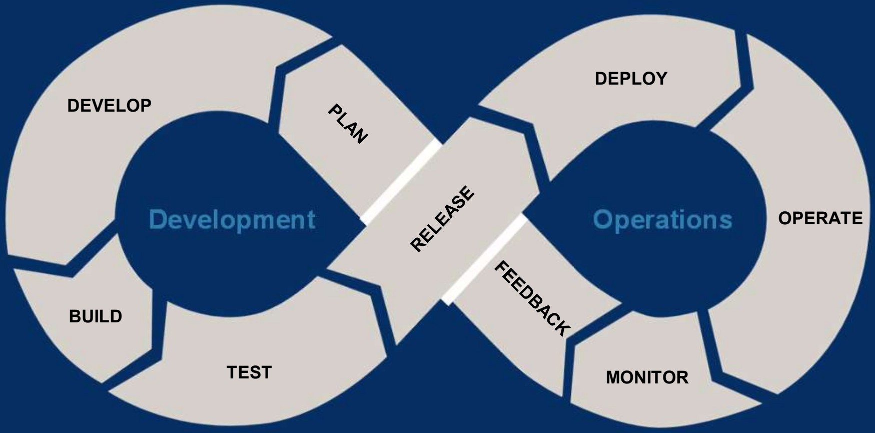
Enable New Types of Algorithms and Functionality in the Field

Software Test and Evaluation as Early as Possible

Flexibility and Optimization with Upgradable Hardware Targets

Better Leverage Data from System Operation

DevOps: A set of practices to automate and integrate processes between Development and Operations (typically software)



Perpetually Upgradeable Machines

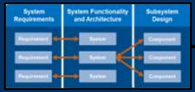
- Faster development and release cycles
- Better leverage data from system operation
- Enable new types of algorithms and functionality in the field

Extending DevOps to Systems, Not Only Software

Cloud

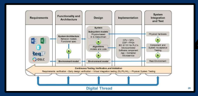
Edge

Desktop


MBSE ↔ MBD

System Simulation

Code Generation


Model- and
Code-based V&V

Lifecycle Mgmt

git

Data

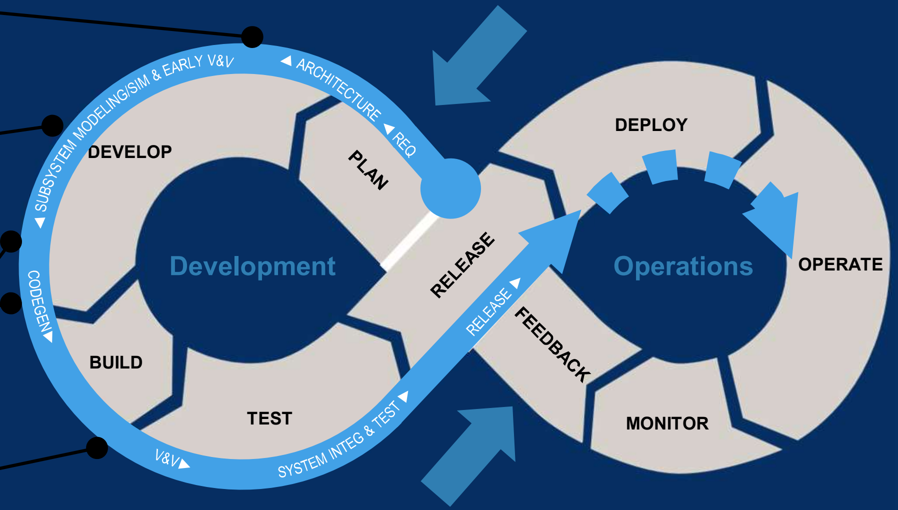
Data stores

Files

Industrial I/O TCP/IP

Deployment

- Cloud
- Edge
- Embedded systems



Containers/Virtual Machines

kubernetes

Digital Twins

- Physics-based
- Data-driven

CI/CD

circleci

Data

Files

Industrial I/O TCP/IP

Streaming data kafka

Dashboards

Power BI

Building Battery State-of-Health Estimation Pipelines for Electrified Vehicles at NIO

Challenges

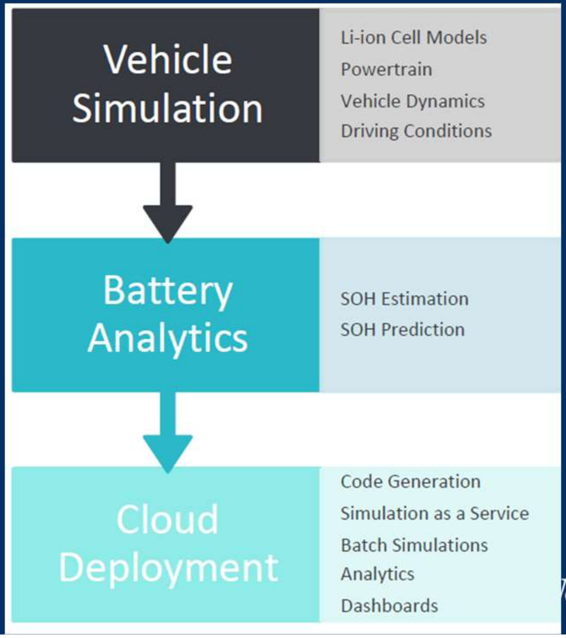
- Large amount of data is required for developing battery state-of-health estimation algorithms. Data is limited during early design phase.

Solution

- A cloud-based scalable simulator to generate synthetic data.

Result

- Generate synthetic data before fleet is available or when edge case data is needed
- Available to many engineers; simulations are run on request
- Integrates synthetic data and data from test vehicles



Vehicle Simulation

Li-ion Cell Models
Powertrain
Vehicle Dynamics
Driving Conditions

Battery Analytics

SOH Estimation
SOH Prediction

Cloud Deployment

Code Generation
Simulation as a Service
Batch Simulations
Analytics
Dashboards

Building Battery State of Health Estimation Pipelines for Electric vehicles

More details coming :

13:30

제품 개발 프로세스 성숙화를 위한 코드비머 활용 방안

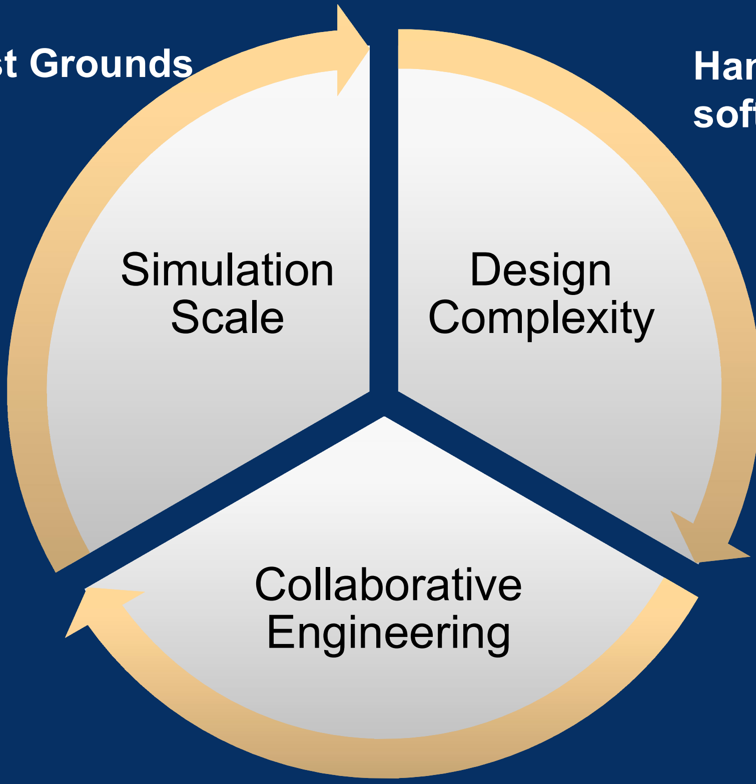
정규진 매니저, *PTC Korea*

Simulink 모델 및 코드 검증을 위한 CI/CD (Continuous Integration / Continuous Delivery) 적용

Summary :Evolution of Model-Based Design for Next Generation E-Mobility

Need for Virtual Development and Test Grounds

- Systems simulation to Full Vehicle Simulations



Handling increasing system and software complexity

- Bridging the gap between Model-Based System Engineering and Model-Based Design

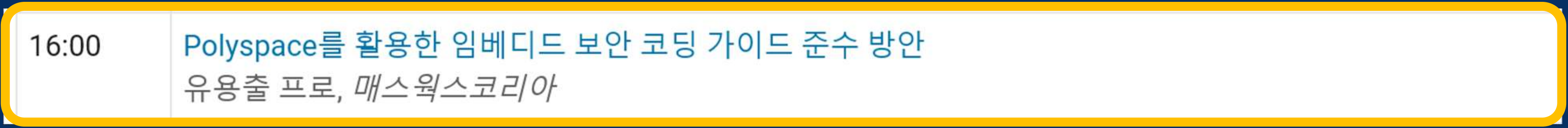
Building innovative features with speed and quality

- Automate and integrate processes using data and models

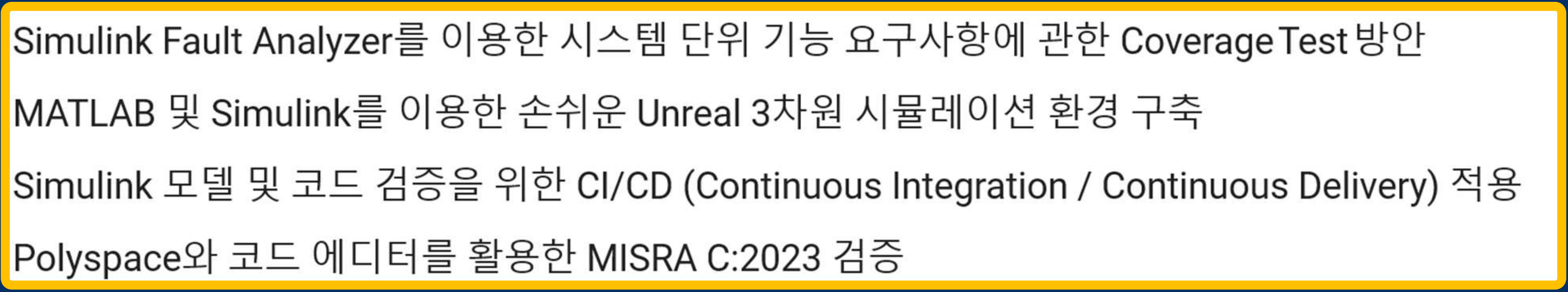
Call to action

Identify a problem statement where you would like to apply these technologies

Team would be happy to discuss about your goals and define an implementation plan for enabling you to move from current state to desired state



16:00 Polyspace를 활용한 임베디드 보안 코딩 가이드 준수 방안
유용출 프로, *매스웍스코리아*



Simulink Fault Analyzer를 이용한 시스템 단위 기능 요구사항에 관한 Coverage Test 방안

MATLAB 및 Simulink를 이용한 손쉬운 Unreal 3차원 시뮬레이션 환경 구축

Simulink 모델 및 코드 검증을 위한 CI/CD (Continuous Integration / Continuous Delivery) 적용

Polyspace와 코드 에디터를 활용한 MISRA C:2023 검증

Thank You