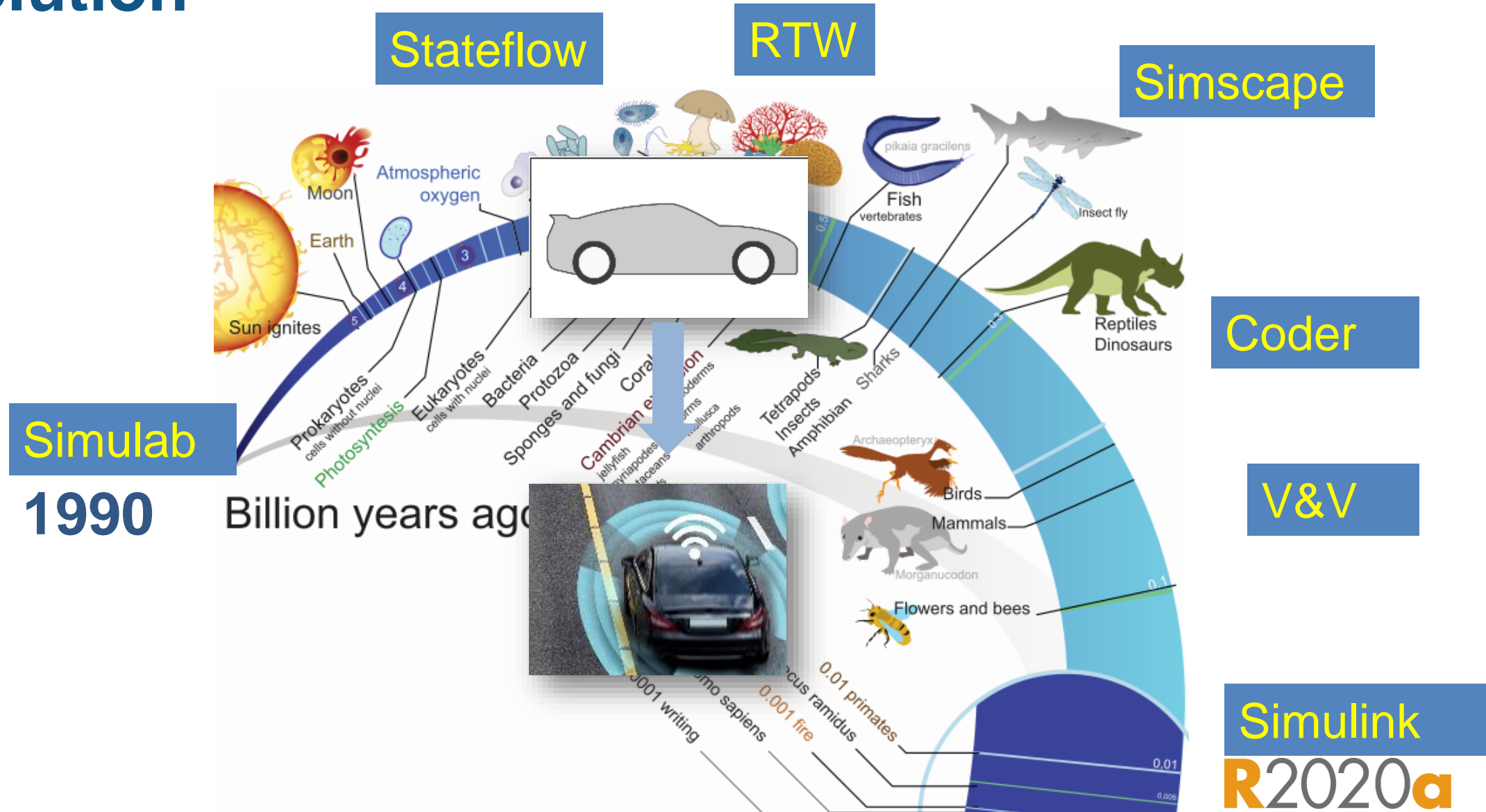


# Model-Based Engineering Platform to Manage Complexity and Scale

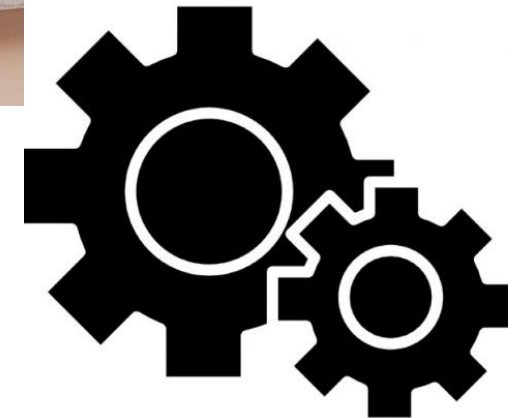
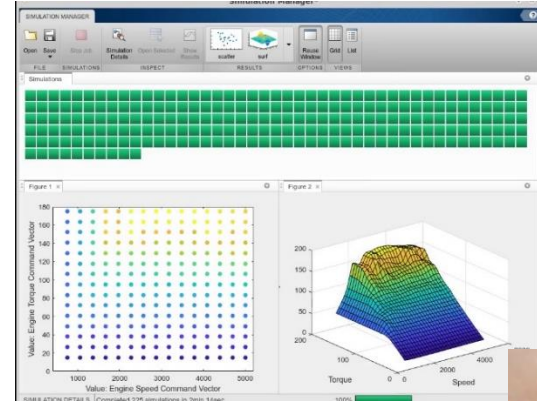
Ramamurthy Mani  
Engineering Director, Simulink Semantics

# Our theme today: Evolution



# The Three Evolutionary Forces at Play

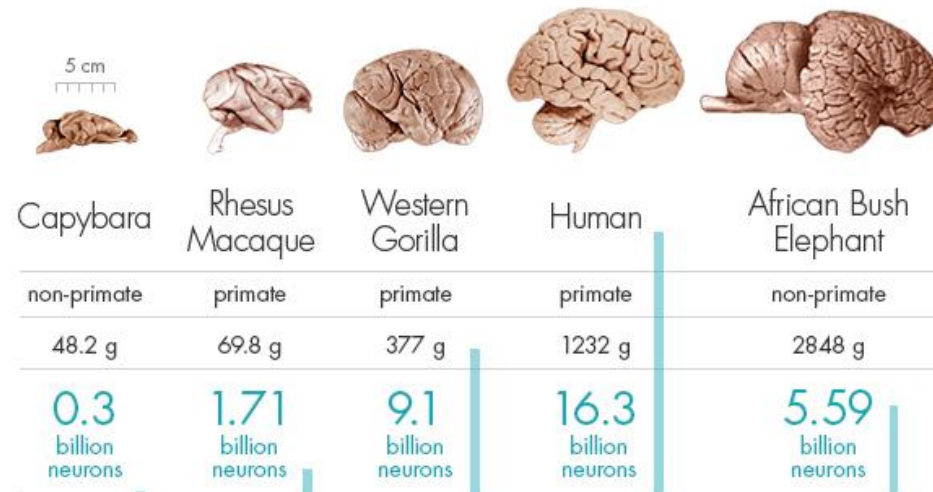
1. Simulation Scale
2. Design Complexity
3. Collaborative Engineering



# Evolving for **Simulation Scale**

## BRAIN SIZE AND NEURON COUNT

Cerebral cortex mass and neuron count for various mammals.



<https://www.quantamagazine.org/how-humans-evolved-supersize-brains-20151110/>

# Trend: Demand for scaled up simulation capabilities

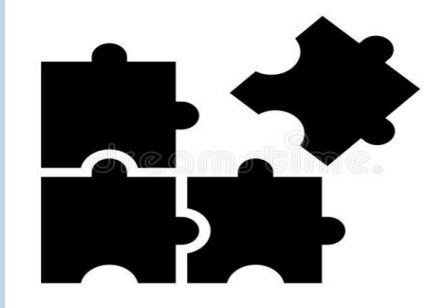


Full Vehicle Simulation

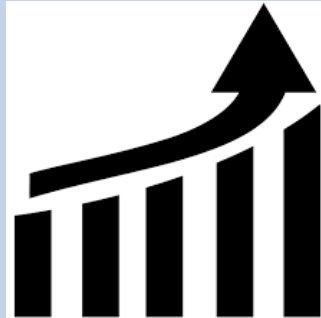
# Strategy: Continuously evolve Simulink to be a best in class Simulation Integration Platform



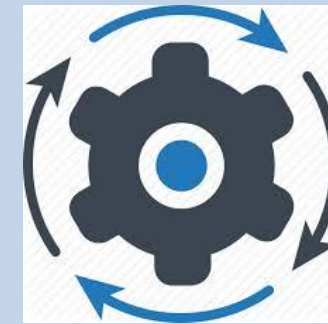
# The primary challenges for simulation scale



Integration

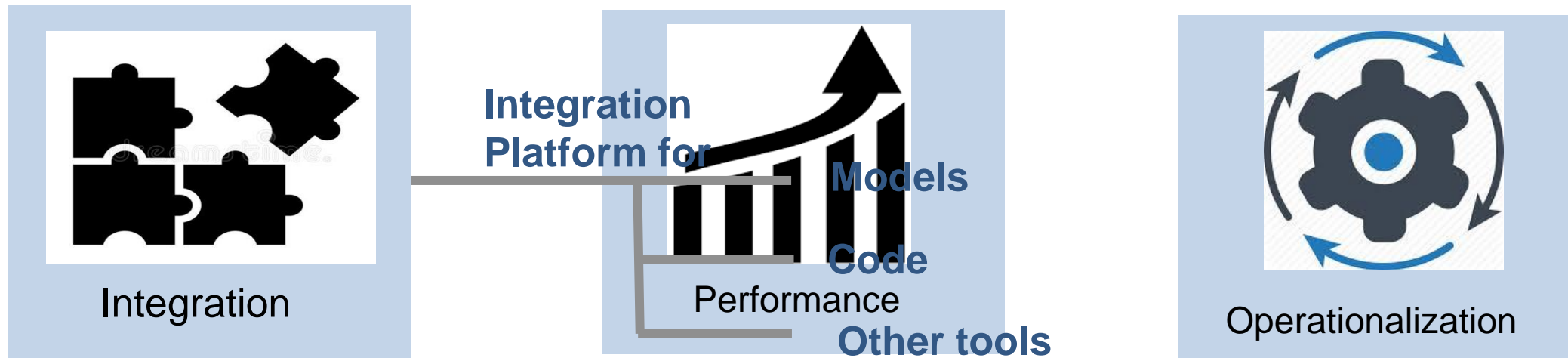


Performance



Operationalization

# Integration of algorithms with multiple simulation interfaces is key

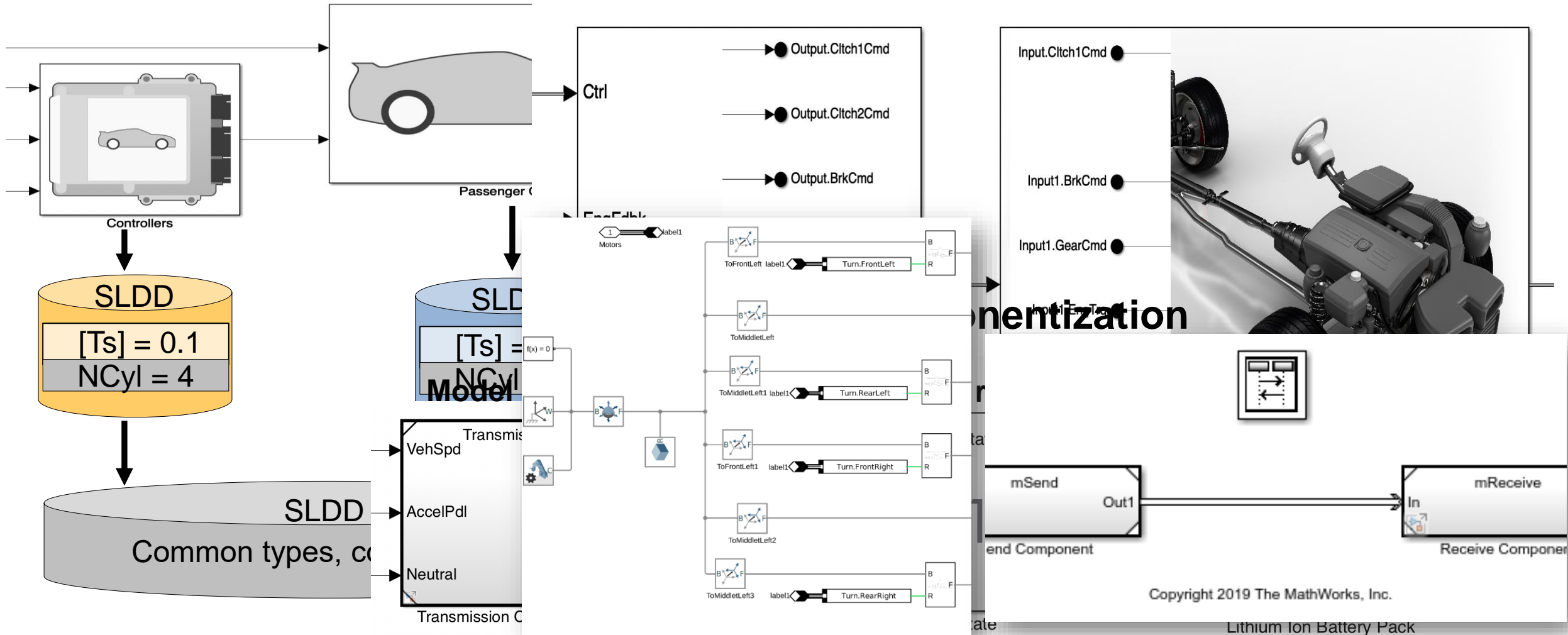




# For Models, core modularity principles underpin integration

## Data Encapsulation

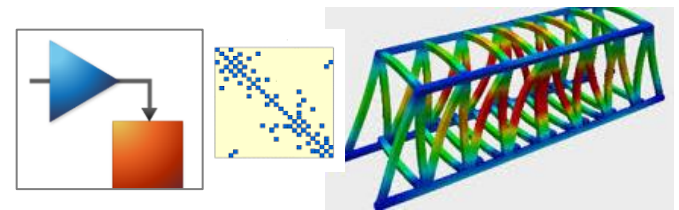
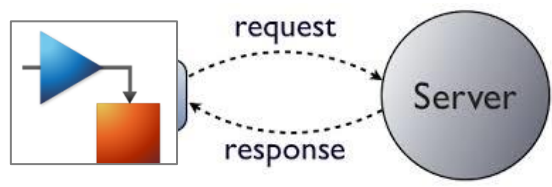
## Interface Management



# You can easily bring C/C++ code into Simulink

Models
<b>Code</b>
Tools

```
void function_name() {
    .....
    .....
}
```

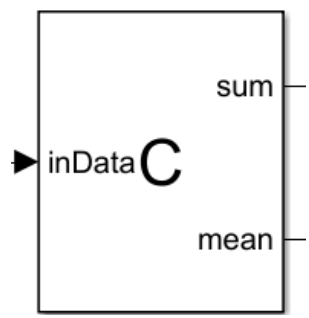


Basic Advanced



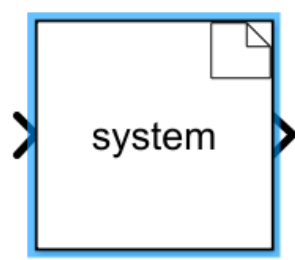
**C Caller**

**R2018b**

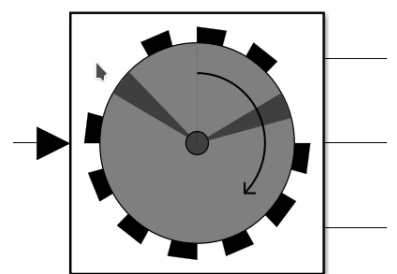


**C Function**

**R2020a**

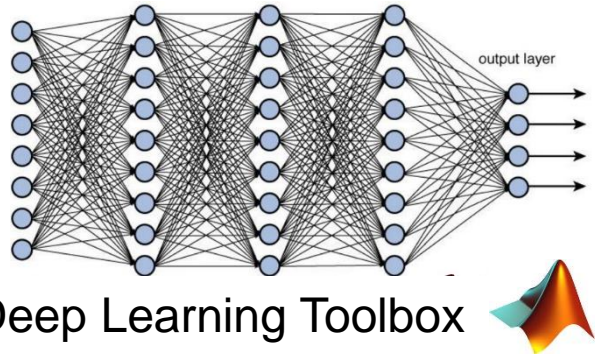


**S-Function  
Builder**



**S-Function**

# You can use MATLAB algorithms like the Deep Learning Toolbox in Simulink

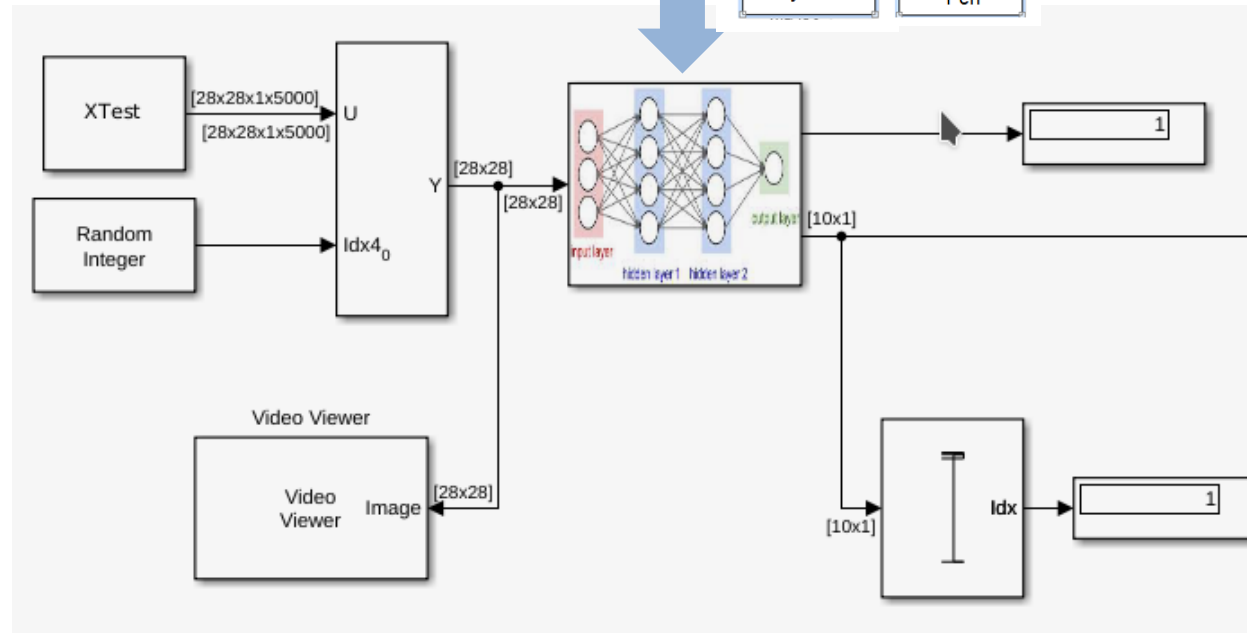


```

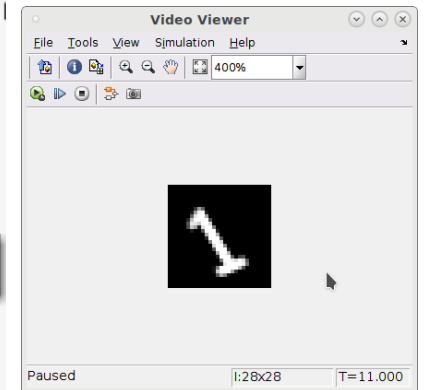
1 classdef DL_load < matlab.System
2
3     properties (Access = private)
4         % Trained deep learning model
5         DLModel
6     end
7
obj.DLModel = coder.loadDeepLearningNetwork('mydnn.mat', 'network')
    
```

- Models
- Code
- Tools

Tensorflow-  
Keras Importer



R2020a

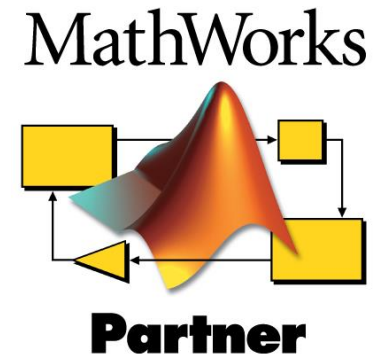


# Simulink has simulation interfaces to 190 connection partner products and services primarily through the S-Function interface

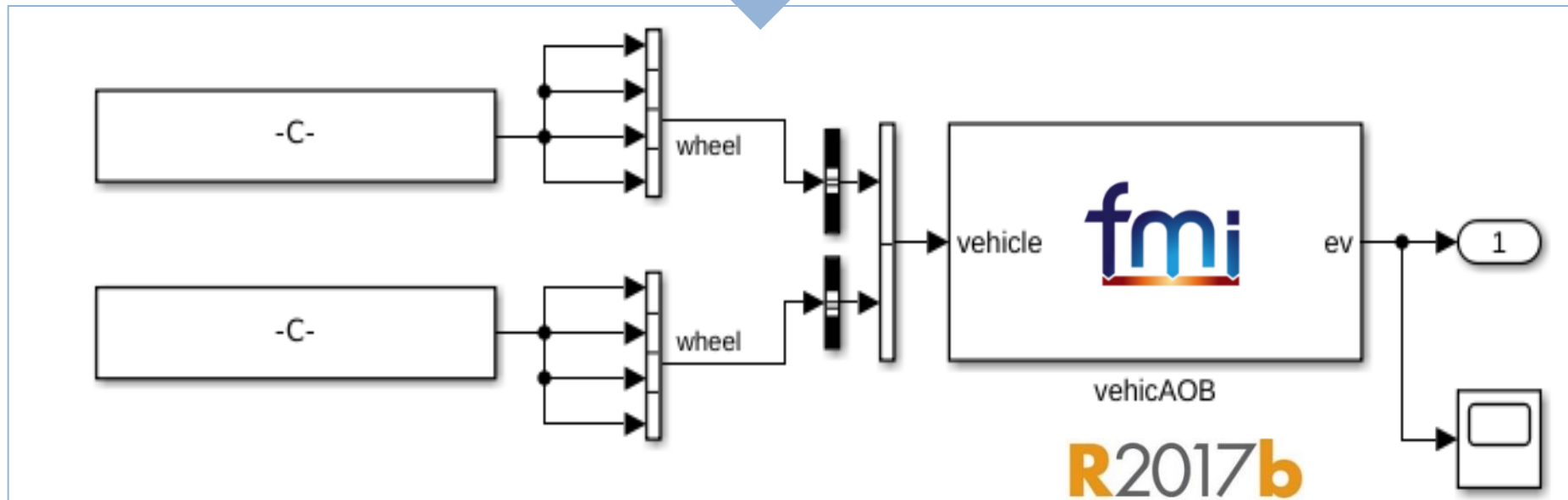
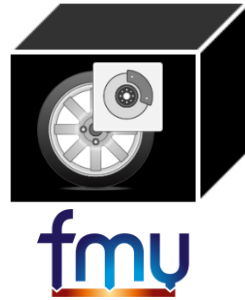
- Models
- Code
- Tools

The screenshot shows the MathWorks website's 'Third-Party Products & Services' section. The search filters are set to 'Modeling and Simulation Tools' and 'System Modeling and Simulation'. The results list 76-97 of 97 items. The following table summarizes the visible results:

Tool Name	Description	Provider
<b>SIMBA</b>	Software for the description and dynamic simulation of wastewater systems	Ifak System GmbH
<b>SimHPN</b>	Toolbox for analysis and simulation of hybrid Petri nets	University of Zaragoza
<b>SIMPACK</b>	Complete multibody simulation in combination with MATLAB	SIMPACK AG
<b>SimulationX</b>	High-end modeling tool for simulating nonlinear, dynamic effects	ITI GmbH
<b>SimWise 4D</b>	Simulation and validation of functional performance for mechanical parts and assemblies	Design Simulation Technologies
<b>SMASH</b>	A mixed-signal, multi-language, and multi-level electronic simulator	Dolphin Integration
<b>Structural Dynamics Toolbox</b>	Finite element modeling and modal analysis with MATLAB	SDTOOLS
<b>SystemVision</b>	Mechatronics system modeling and analysis software	Mentor Graphics Corporation
<b>Tactical Engagement Simulation Software (TESS)</b>	ECM evaluation tools using terminal phase engagement simulations	Tactical Technologies
<b>Thermolib</b>	Toolbox for thermodynamic calculations and thermodynamic systems simulations in MATLAB® and Simulink®	EUtech Scientific Engineering GmbH



# Using FMUs inside Simulink is easy and expressive

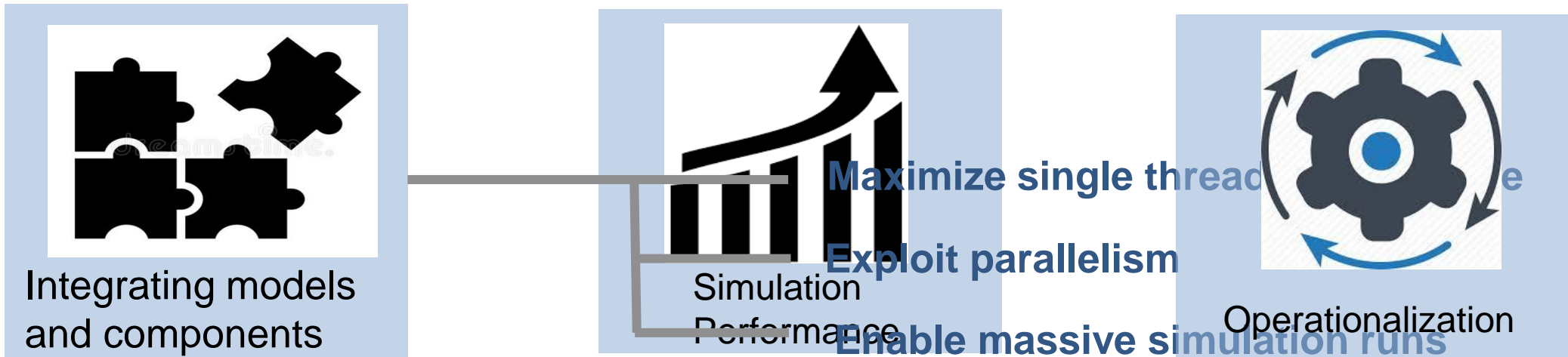


Models

Code

Tools

# System-level simulations are computationally expensive



# Maximizing performance by discovering speed-up opportunities: Performance Advisor

Filter checks

- ✔ Passed
- ✘ Failed
- ⚠ Warning
- 📄 Not Run

Navigation

- Performance Advisor
- 1 Baseline
- 2 Simulation
  - 2.1 Checks Occurring Before Update
  - 2.2 Checks that Require Update Diagram
  - 2.3 Checks that Require Simulation to Run
- 3 Simulation Targets
  - 3.1 Check Simulation Modes Settings
  - 3.2 Check Compiler Optimization Settings

## Simulink Performance Advisor Report - vdp

**Simulink version: 8.3**  
**System: vdp**

Performance Advisor

1 Baseline ✔1 ✘0 ⚠0 📄0

✔ **Create baseline**

✔ **Passed** Baseline generated successfully. Simulation took 00:00:00.580 seconds.

**Input Parameters Selection**

Name	Value
Stop Time	10
Check to view baseline signals and set their tolerances.	false

2 Simulation ✔2 ✘0 ⚠2 📄8

2.1 Checks Occurring Before Update ✔1 ✘0 ⚠2 📄6

⚠ **Identify resource-intensive diagnostic settings**

Some diagnostics incur run-time overhead during simulation. Review the following parameters in the ... for these parameters.

Click link(s) to make changes manually. Alternatively, click the 'Modify all' button below to have Perf

	Severity	Diagnostics checked	Origin
<b>Solver</b>	✔	<a href="#">Diagnostics &gt; Solver data inconsistency</a>	none
<b>Signals</b>	⚠	<a href="#">Diagnostics &gt; Data Validity &gt; Signal resolution</a>	Explici

- Consolidated advice on performance
- Gives advice that works!
- Helps discover performance focused capabilities

# Invest in multiple parallelization techniques for boosting performance

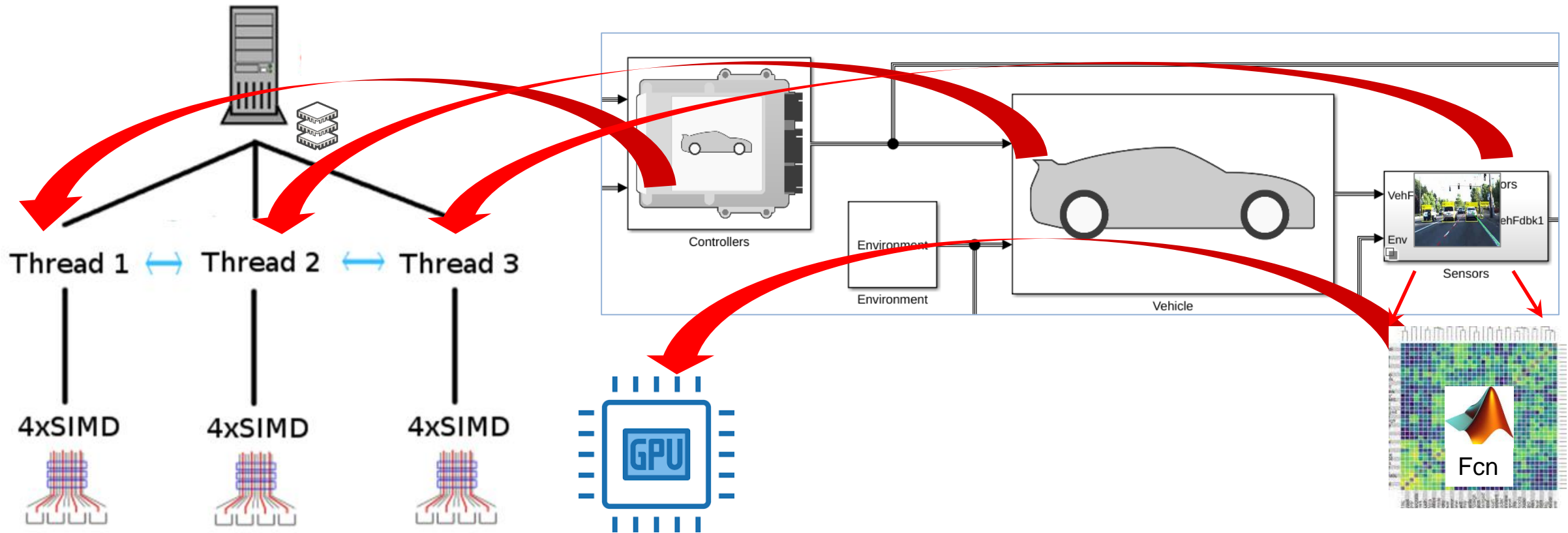
Model block, S-function, FMU import

**R2018a**

**R2018a**  
Dataflow SIMD

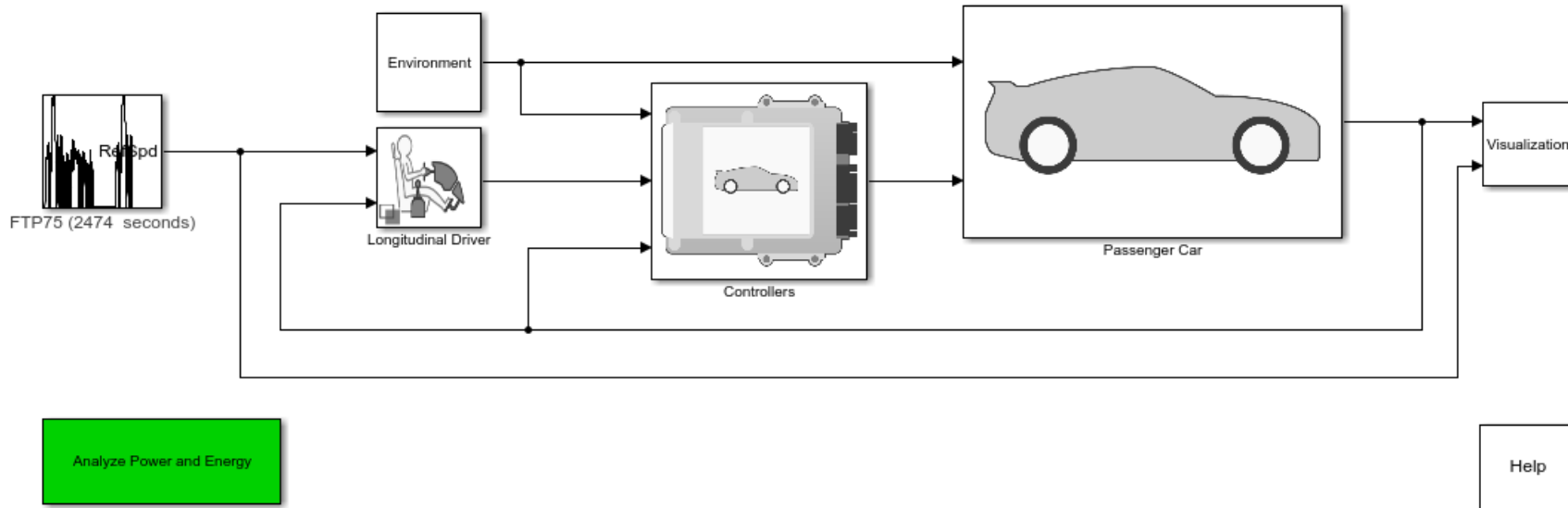
ForEach Subsystem Parallelization  
MATLAB Function GPU acceleration  
Compute Clusters

**FUTURE  
RELEASE**

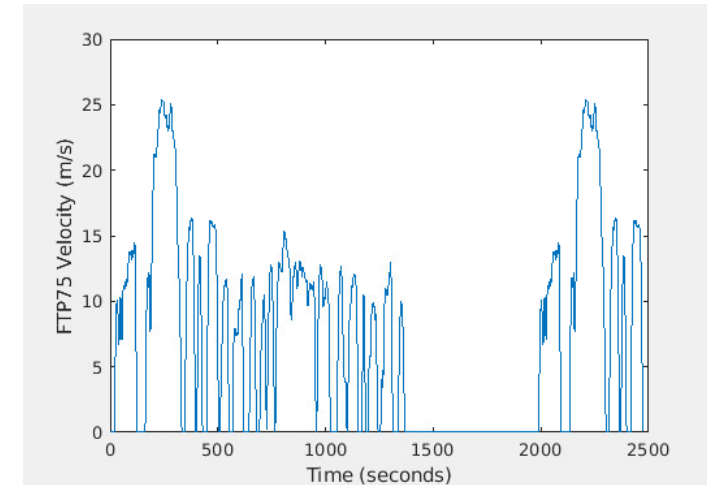




# Design envelope studies require a large number of simulations



Copyright 2019 The MathWorks, Inc.



Full vehicle model

Driving cycle

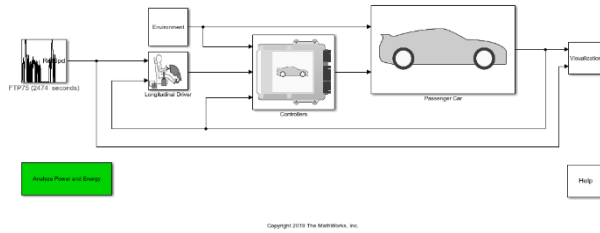
100 drive cycles × 10 vehicle loadings × 10 weather conditions

10,000 simulations

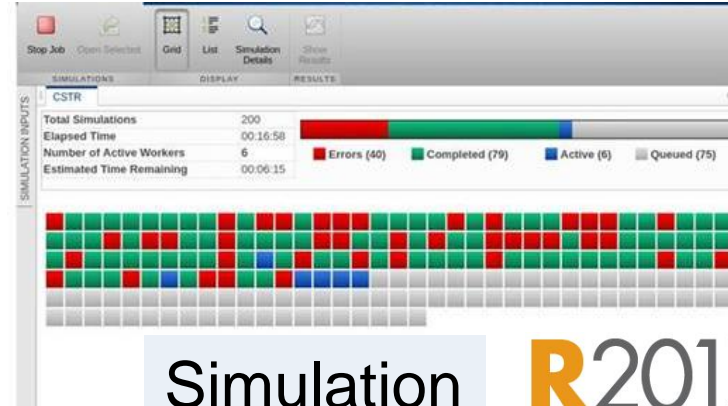
Optimize gear ratios

# Simulink enables massive simulation workflows

## Setup



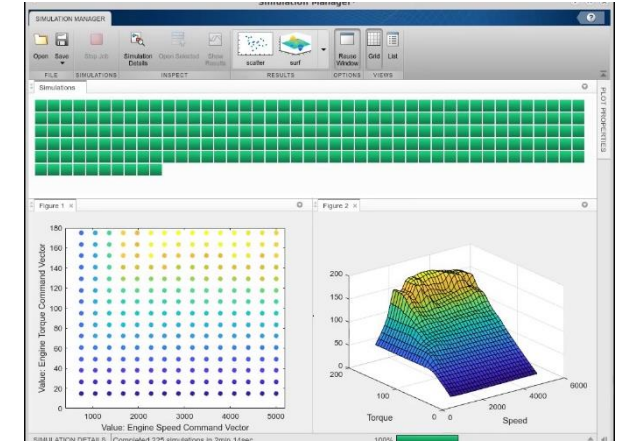
## Simulate



Simulation Manager

R2017b

## Analyze



Simulation Manager

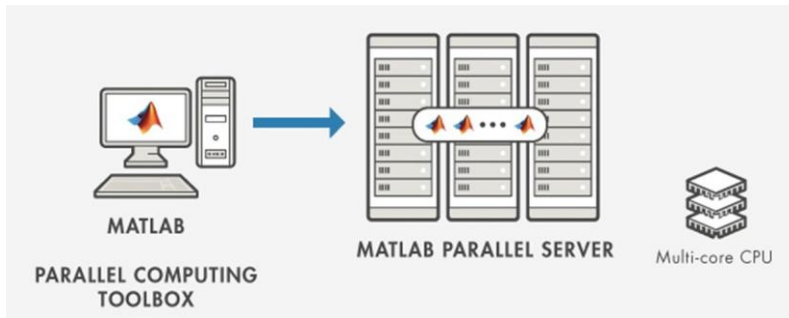
R2019b  
FUTURE RELEASE

parsim

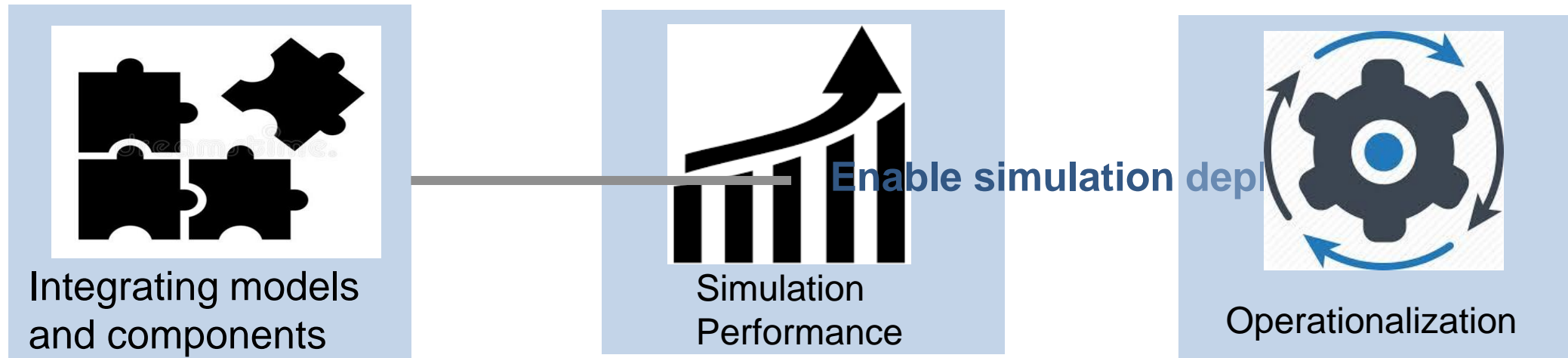
R2017b

batchsim

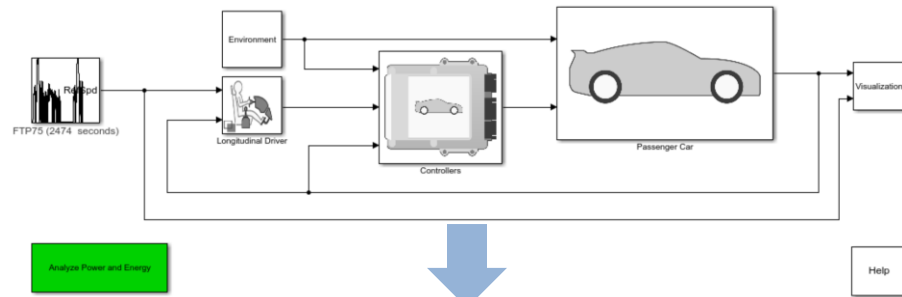
R2018b



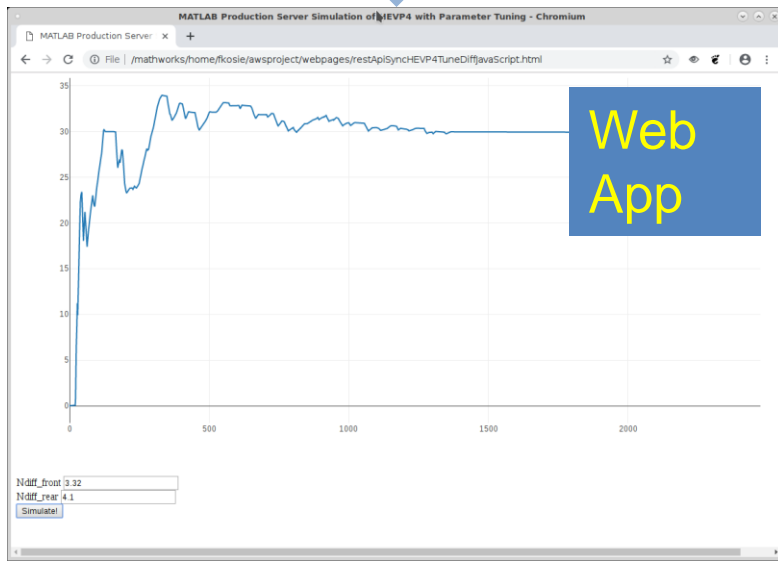
# Extend simulations to Operational phases of the system



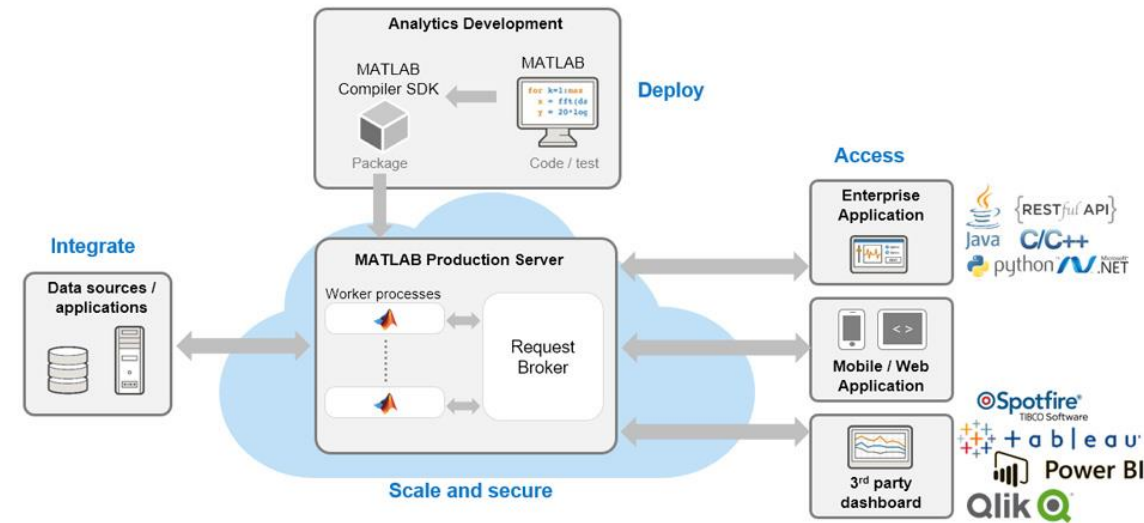
# Simulink Compiler enables deployment of simulations



**Simulink Compiler**  
**R2020a**



Integrate  
as  
Enterprise  
Application

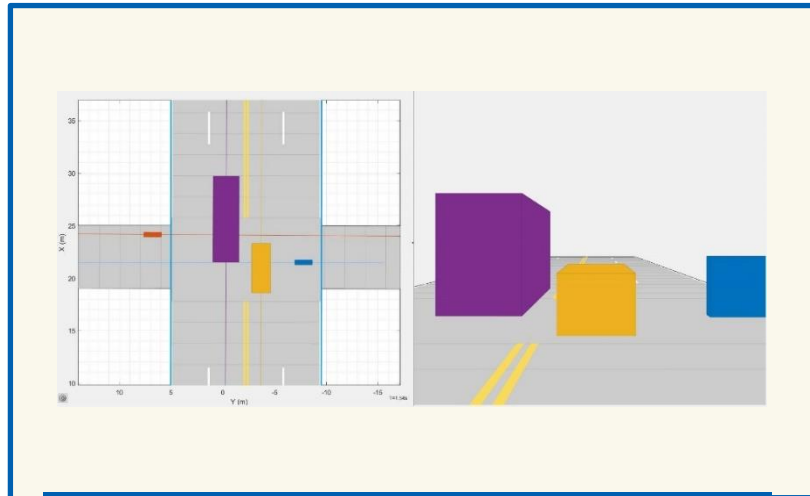


Trend: Demand for simulating complex scenarios with multiple actors is increasing

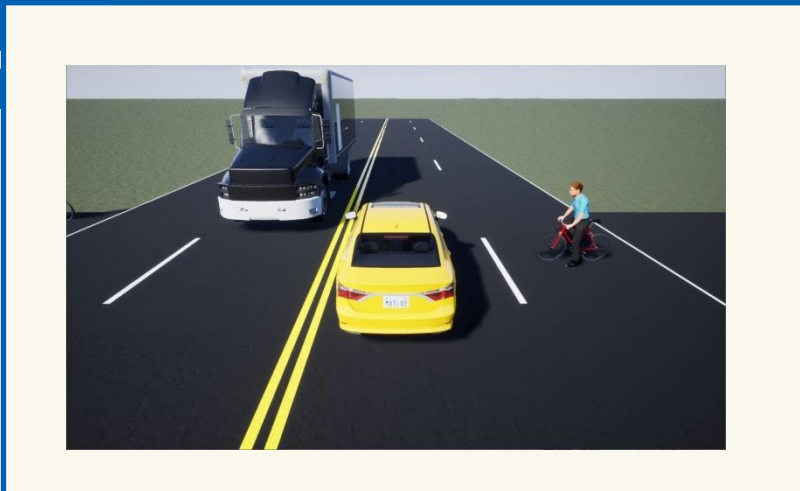


Scenario Simulations for Autonomy

# Strategy: Create a platform for system-of-systems (scenario) simulations



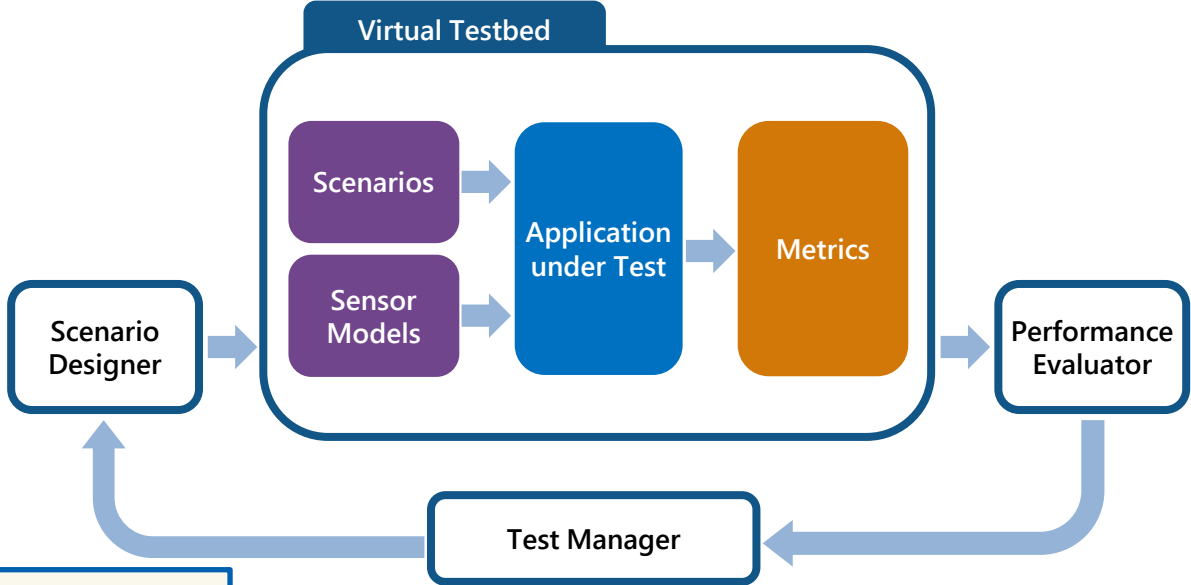
Cuboid Driving Simulation



Unreal Engine Driving Simulation



RoadRunner



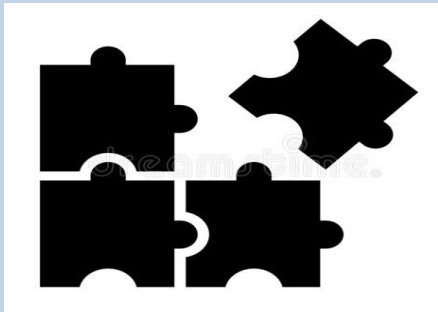
# Simulink platform is evolving to meet the demands of scaled up simulations



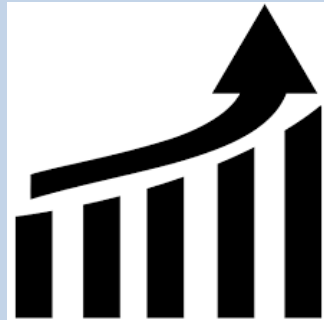
Full Vehicle Simulation



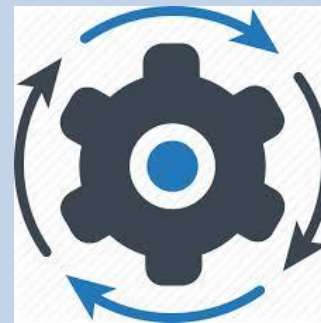
Scenario Simulations for Autonomy



Integrating models and components



Simulation Performance



Operationalization



Scenario Simulation

# Evolving for **Design Complexity**



<https://en.wikipedia.org/wiki/Tiktaalik>



*Trend: Some rumblings in the force*

# **MAB Breakout session 2012 on *System Architecture***

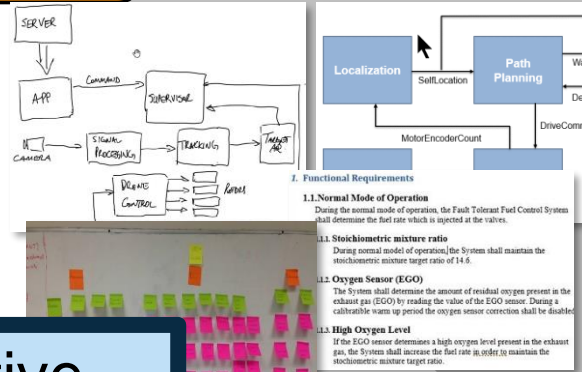


“Not sure you get it...”

Wonder what's  
for lunch?

# Why the discontent?

Stakeholder Needs

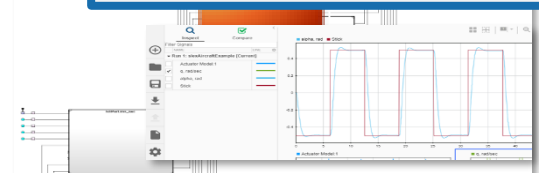


Descriptive Architectures



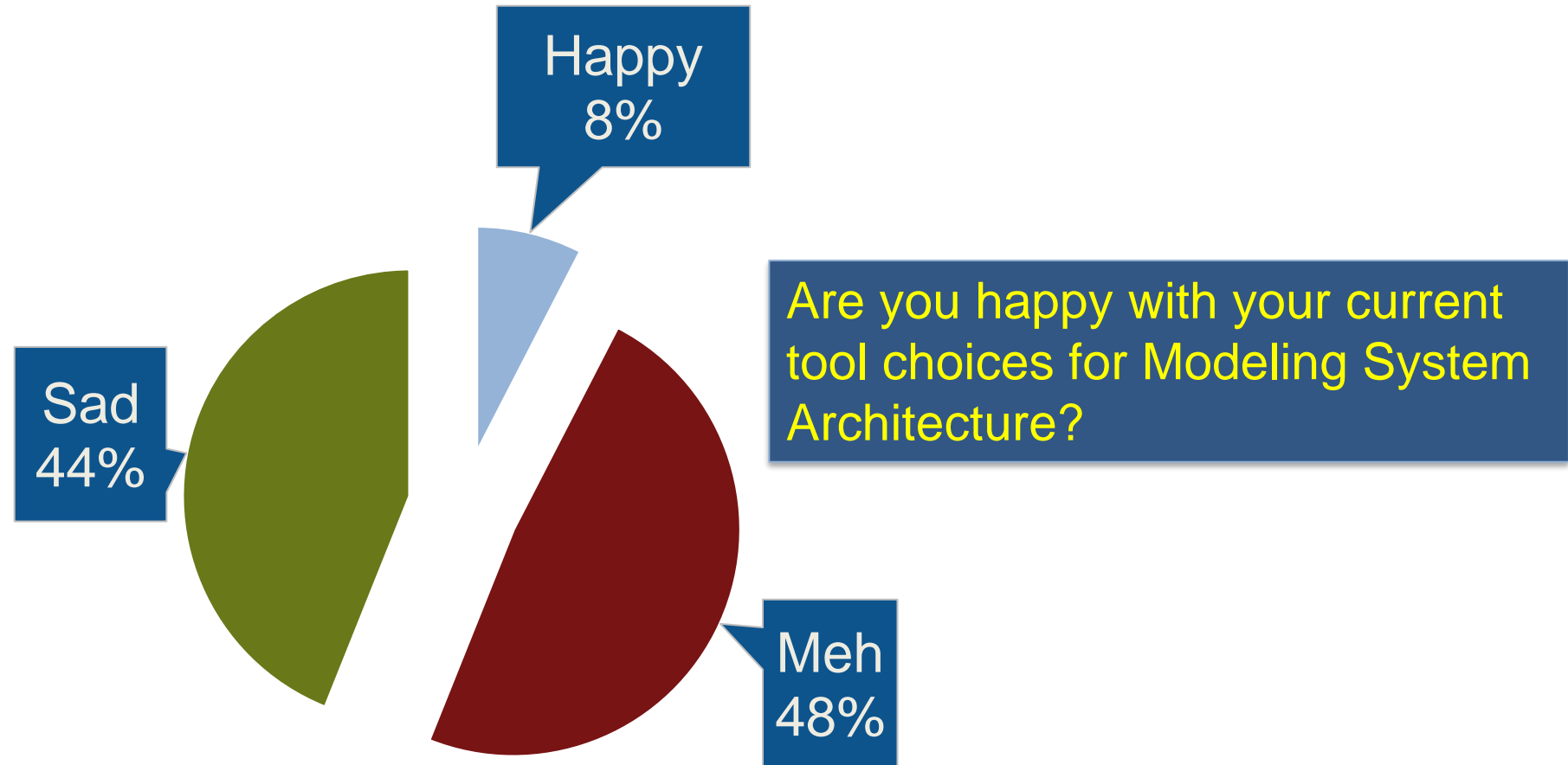
Implementations

**Customer quote:**  
 “We have tried to build the architecture model in SysML and connect it to the design in Simulink ...  
 ... 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”



# Survey @ Modeling System Architecture Breakout

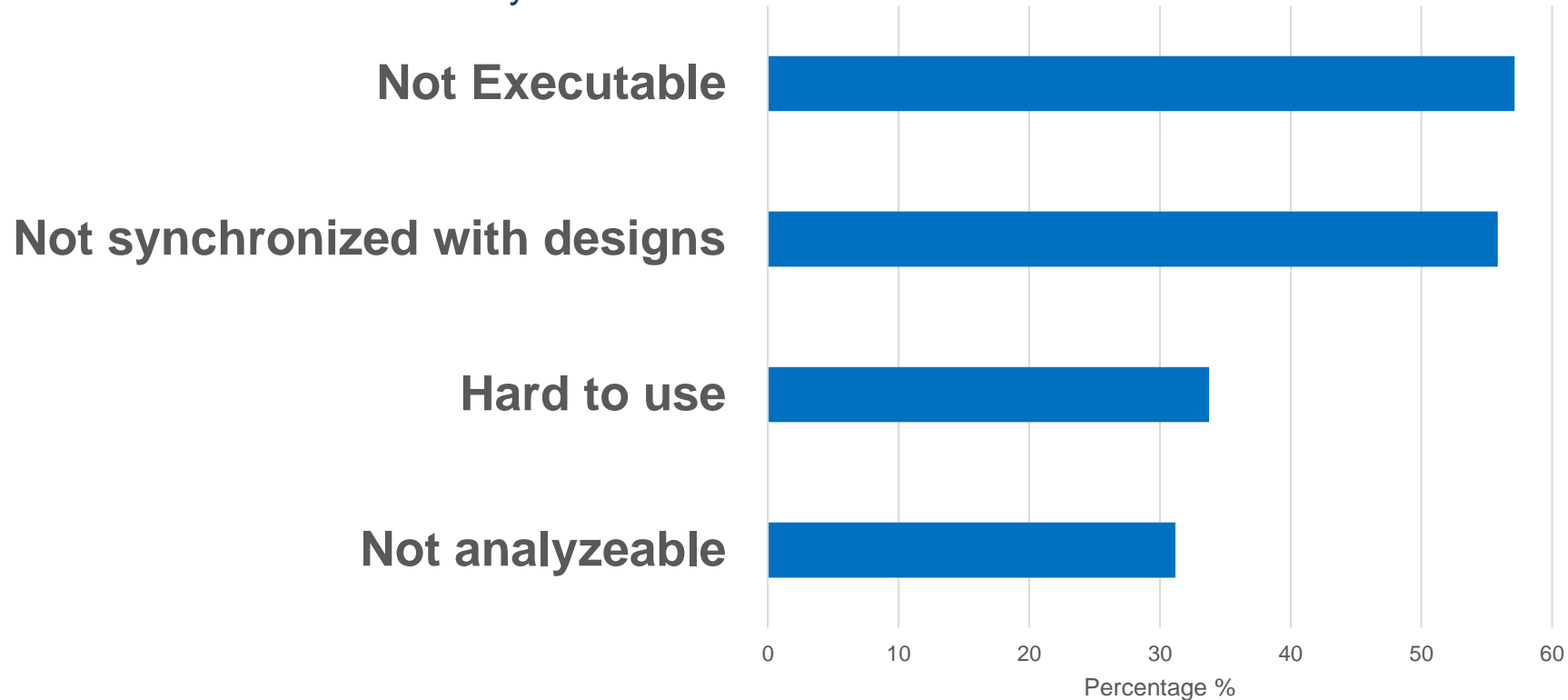
## Newton MAB - 2018



# More specifically, what are the pains?

“We do not like our current System Architecture solution because they are:”

Newton MAB Survey 2019



# Strategy: Build an MBSE Ecosystem that fits with MBD

Be Intuitive

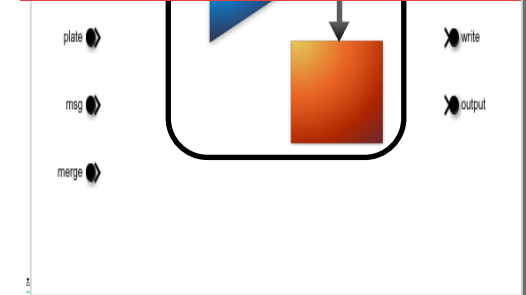
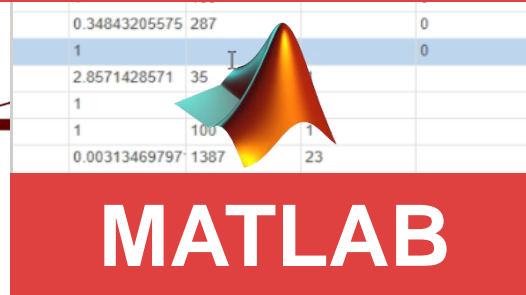
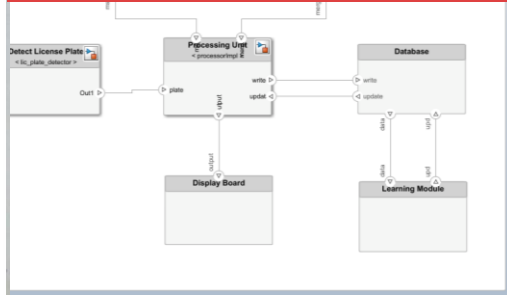
Facilitate Analysis

Tackle Complexity

Enable Implementation

## System Composer

## Simulink

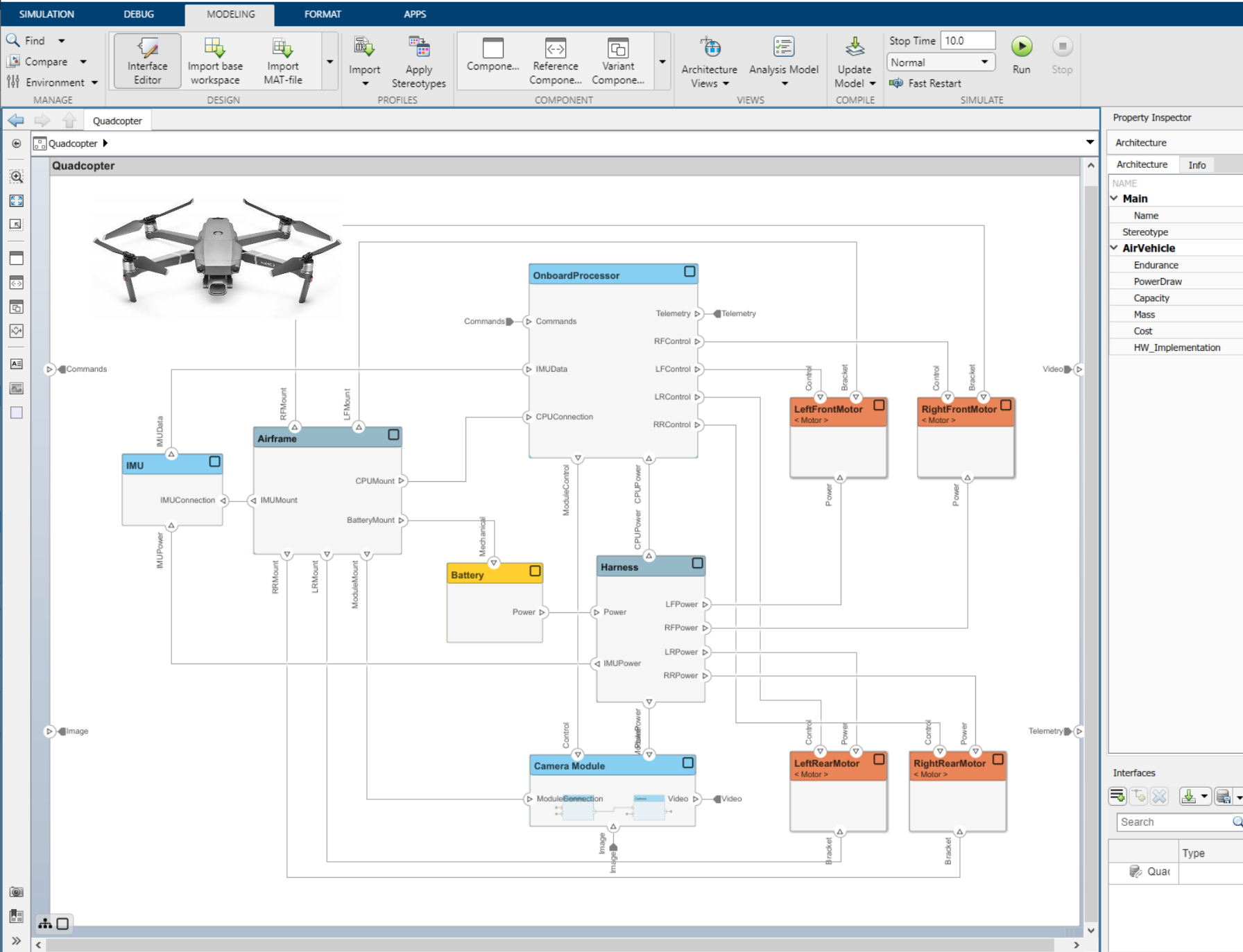


## MATLAB

Requirements Coverage Reporting and Impact Analysis

## Simulink Requirements

Index	Summary	Implemented
> 1.1	Airworthiness	<div style="width: 80%; background-color: blue;"></div>
> 1.2	Communications	<div style="width: 100%; background-color: blue;"></div>
▼ 1.3	Payload Capabilities	<div style="width: 70%; background-color: blue;"></div>
1.3.1	Carrying Capacity	<div style="width: 100%; background-color: blue;"></div>
1.3.2	Payload Bay Capacity	<div style="width: 100%; background-color: blue;"></div>
1.3.3	Default Payload	<div style="width: 100%; background-color: blue;"></div>
1.3.4	Payload Protection	<div style="width: 100%; background-color: blue;"></div>



Property Inspector

Architecture

Architecture Info

NAME

- ▼ Main
  - Name
  - Stereotype
- ▼ AirVehicle
  - Endurance
  - PowerDraw
  - Capacity
  - Mass
  - Cost
  - HW\_Implementation

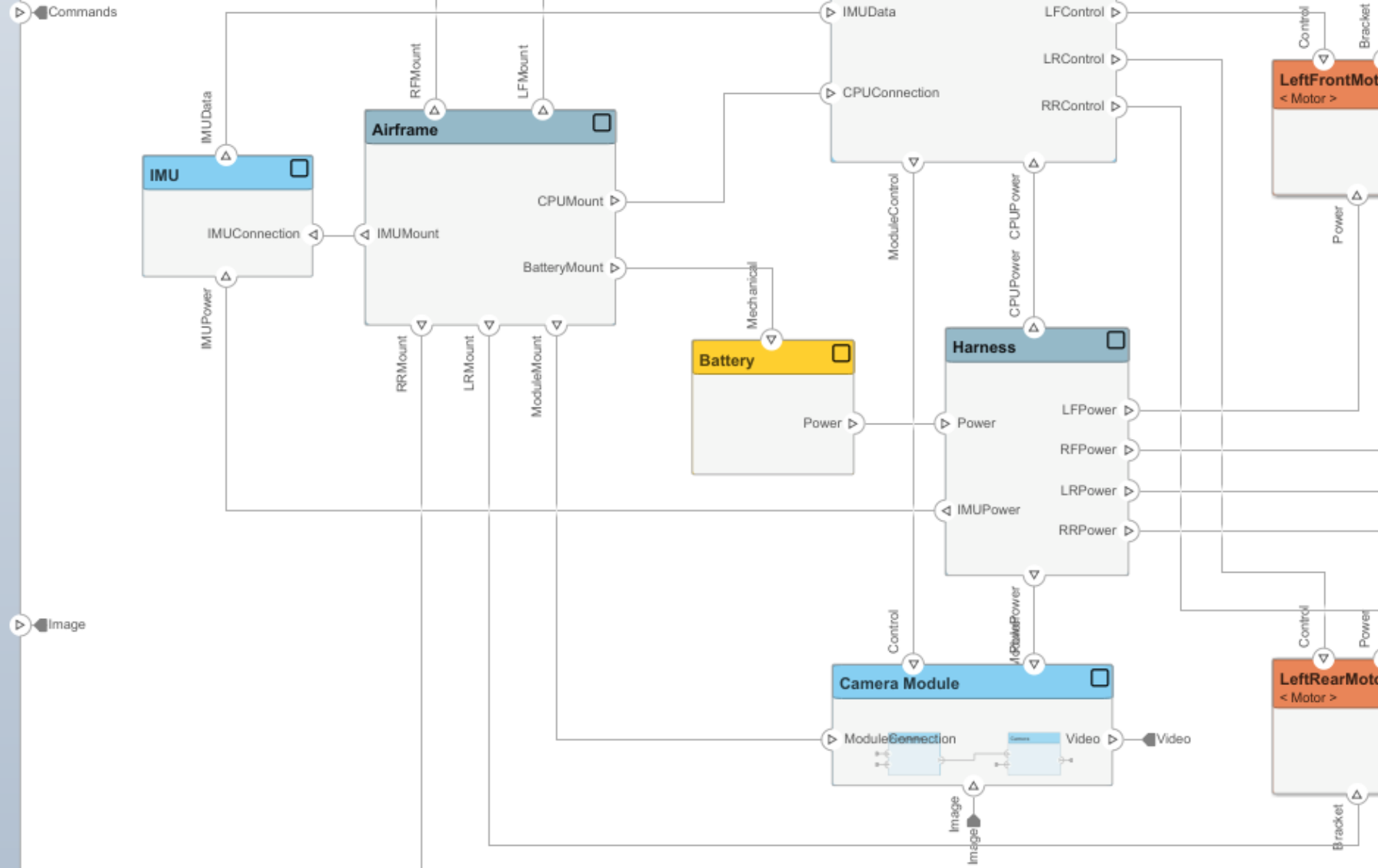
Interfaces

Search

Quar	Type

R2019a

# “Sketch” system interfaces and elaborate incrementally



R2019a

# Extend elements with your own custom metadata using Profiles & Stereotypes

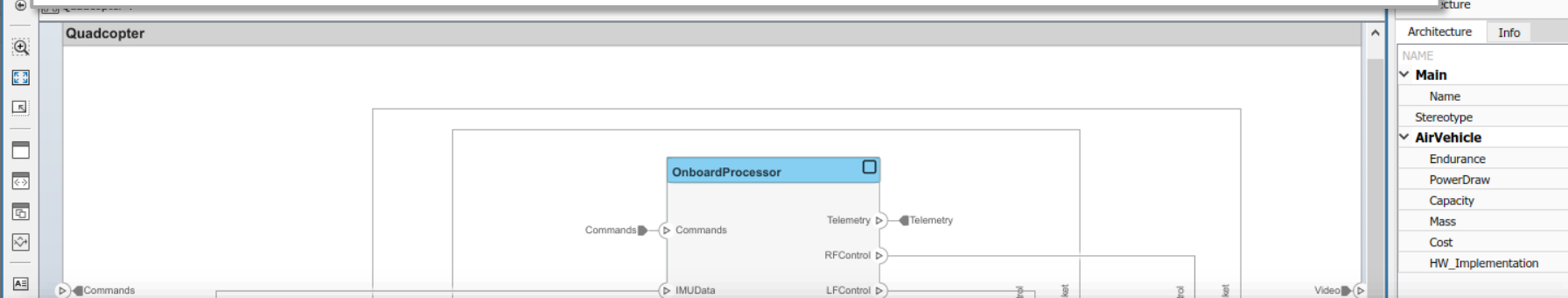
The screenshot shows the Simulink Property Inspector window for a Quadcopter model. The main workspace displays a block diagram with two motor blocks, 'LeftFrontMotor' and 'RightFrontMotor', each connected to 'Control' and 'Bracket' ports. The Property Inspector is open to the 'Architecture' tab, showing a table of properties for the 'AirVehicle' stereotype.

NAME	VALUE
<b>Main</b>	
Name	Quadcopter
Stereotype	Add
<b>AirVehicle</b>	
Endurance	0 min
PowerDraw	0 W
Capacity	0 mAh
Mass	0 g
Cost	0 \$
HW_Implementation	PhysicalDevice

R2019a



# Analyze system characteristics and quantitatively evaluate choices using MATLAB



Architecture	
Info	
NAME	
▼	<b>Main</b>
	Name
	Stereotype
▼	<b>AirVehicle</b>
	Endurance
	PowerDraw
	Capacity
	Mass
	Cost
	HW_Implementation

## Analysis Viewer (Technical Preview)

HOME

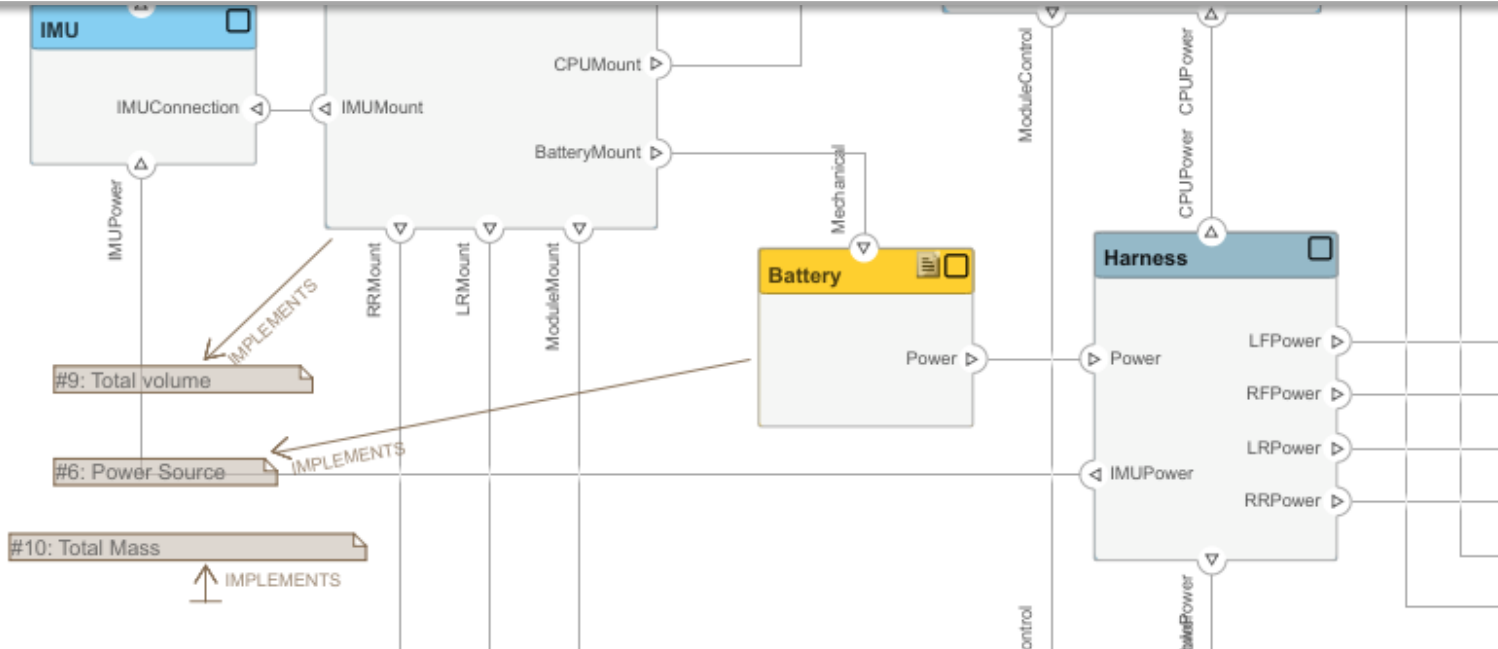
New Open Save Delete Analyze Arguments Refresh Automatic Overwrite Update

INSTANCE MODEL	Endurance	Mass	PowerDraw	...
Instances				
EnduranceModel	4.0997877	85	40	137
Airframe				
Battery		5	3.7	2000
Camera Module		27		27
Camera		25		25
PowerSwitch		2		2
Harness		2		
IMU		10		10
LeftFrontMotor		25		0
LeftRearMotor		25		0
OnboardProcessor		100		100
RightFrontMotor		25		0
RightRearMotor		25		0

R2019a

# Trace to system requirements

## Refine requirements alongside the architecture



R2019a

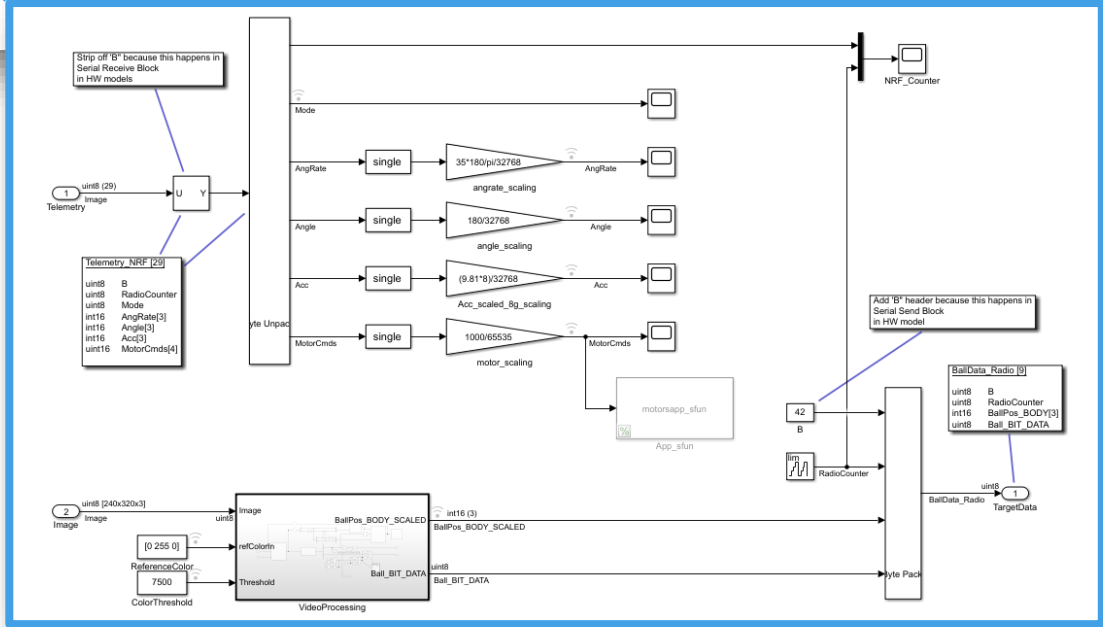
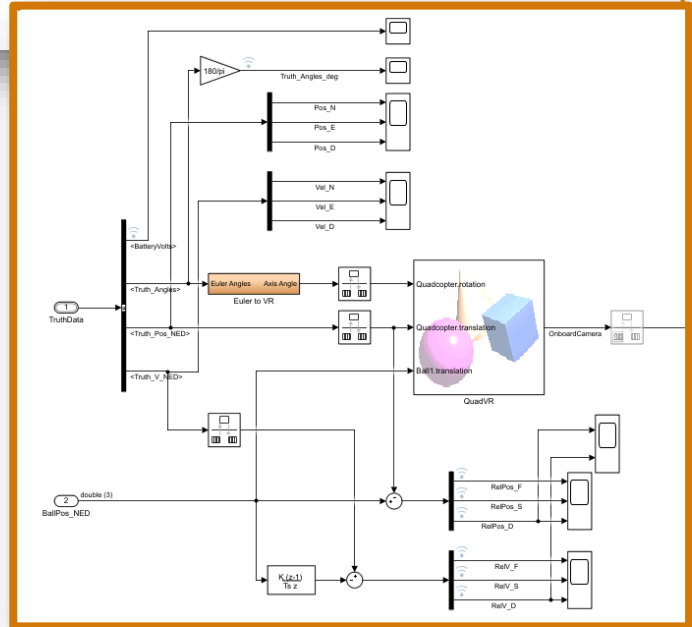
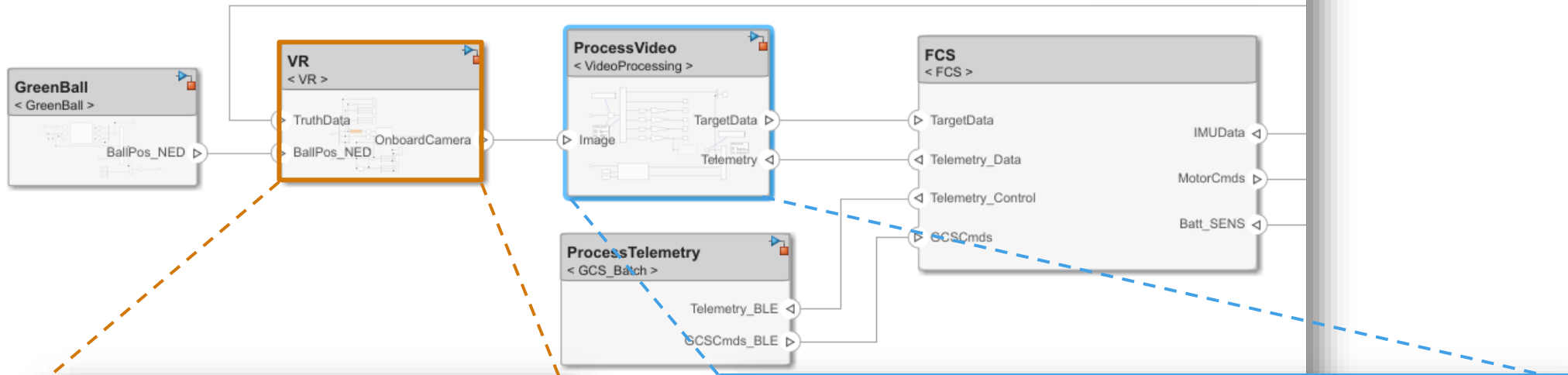
With Simulink Requirements

Requirements - Quadcopter

View: Requirements

Index	ID	Summary	Implemented
quadcopter			<div style="width: 10%; background-color: blue;"></div>
1	#1	Aircraft Performance	<div style="width: 0%; background-color: blue;"></div>
1.1	#14	Aircraft horizontal velocity	<div style="width: 0%; background-color: blue;"></div>
1.2	#15	Aircraft vertical velocity	<div style="width: 0%; background-color: blue;"></div>
2	#2	Power System	<div style="width: 10%; background-color: blue;"></div>
2.1	#6	Power Source	<div style="width: 100%; background-color: blue;"></div>

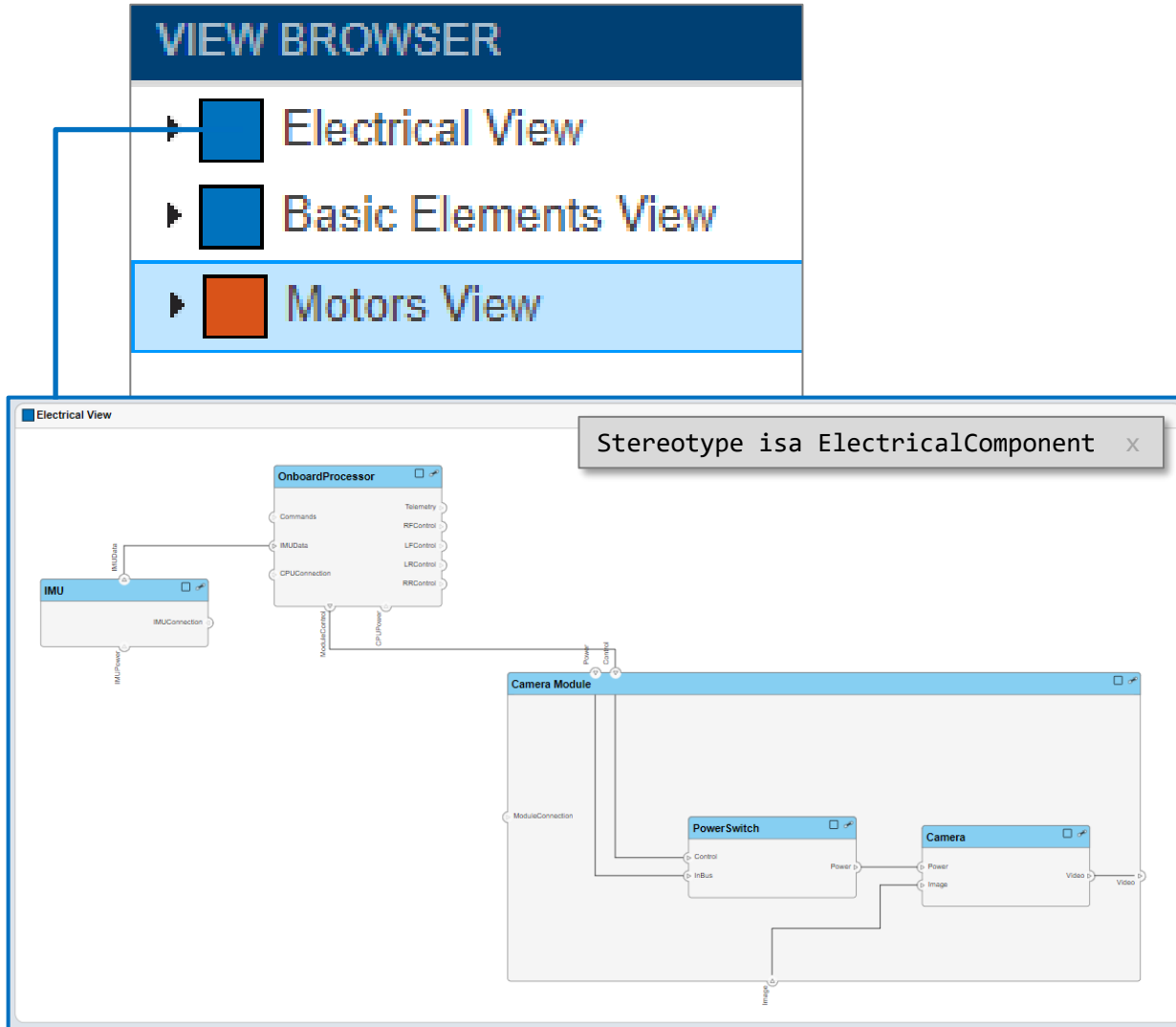
# Link design models to components and ensure consistent interfaces



R2019a

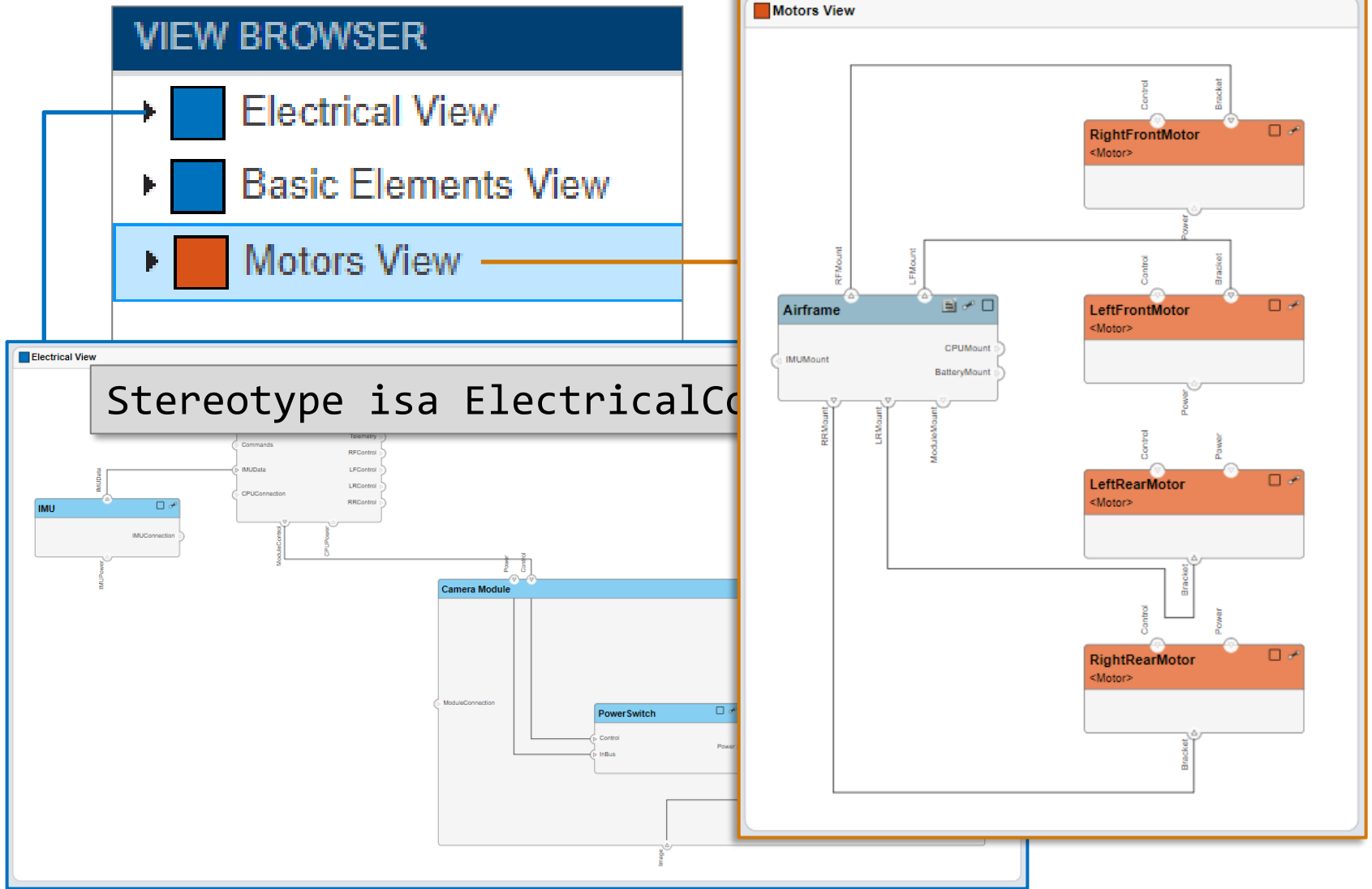
# Simplify the complex with Filters and autogenerated Views

## R2019b



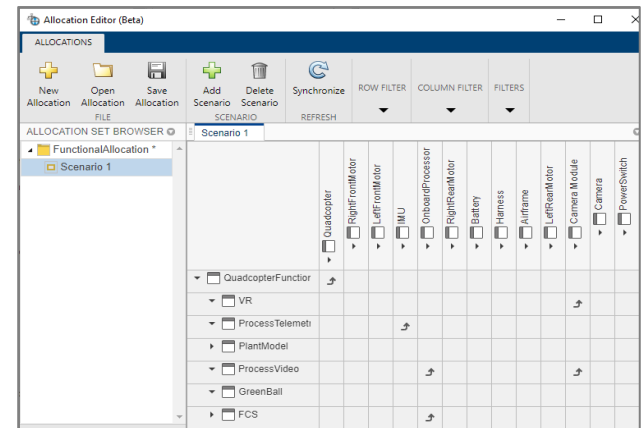
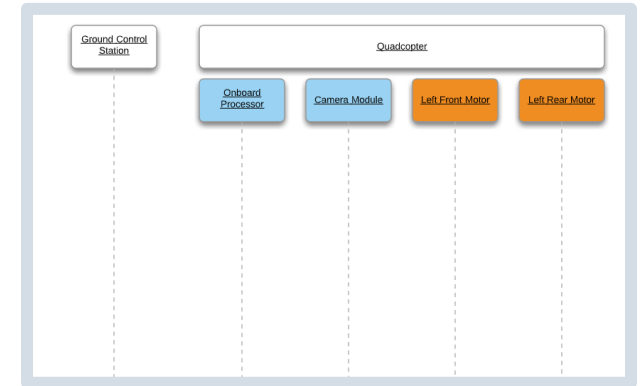
# Simplify the complex with Filters and autogenerated Views

## R2019b



# And we are only getting started. Coming soon:

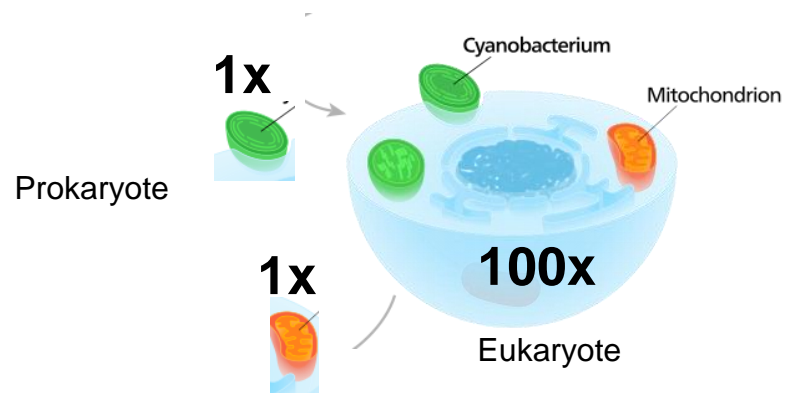
- Behavior modeling using Sequence Diagrams
- Architecture Allocations through Analysis (e.g. Logical to Physical)
- Software Architecture Modeling
  - Link to AUTOSAR (R2019b)
  - Other middlewares such as DDS
- And much more!



```

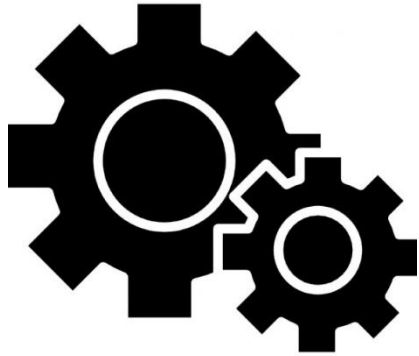
187 /* S-Function (sfunXmlAppWrite): '<Root>/XmlAppWrite' */
188
189 /* bracket all code for block in (...) to avoid namespace issues */
190 {
191     DDS_InstanceHandle_t instance_handle = DDS_HANDLE_NIL;
192
193     /* write DDS sample */
194     if (localDW->XmlAppWrite_DataWriterPtr != NULL) {
195         localB->XmlAppWrite = ModuleA_BasicTypeDataWriter_write(
196             (ModuleA_BasicTypeDataWriter *)localDW->XmlAppWrite_DataWriterPtr,
197             &rtb_BusCreator, &instance_handle);
198     } else {
199         /*
200          * most likely cause of error is either not able to locate participant
201          * configured for Dynamic Data while code was generated for Compiled/
202          */
203         localB->XmlAppWrite = DDS_RETCODE_PRECONDITION_NOT_MET;
204     }
205 }
    
```

# Evolving for Collaborative Engineering



<https://en.wikipedia.org/wiki/Symbiogenesis>

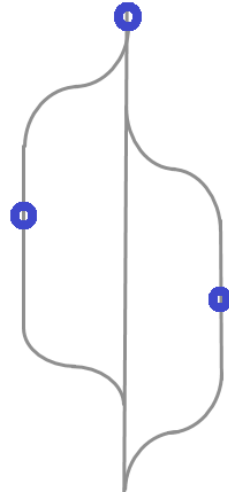
# Trend: An increased demand for Agile team-based workflows



Shared team environment



Collaboration



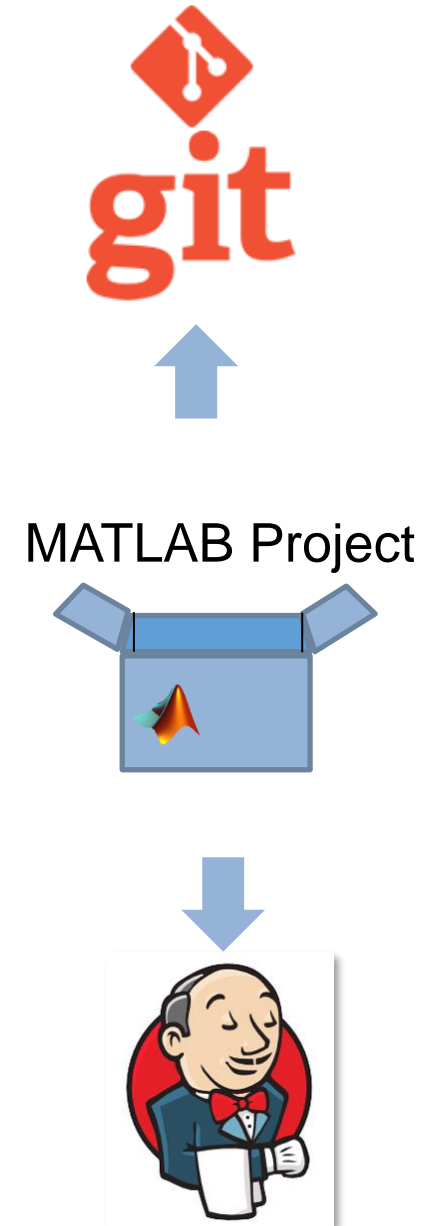
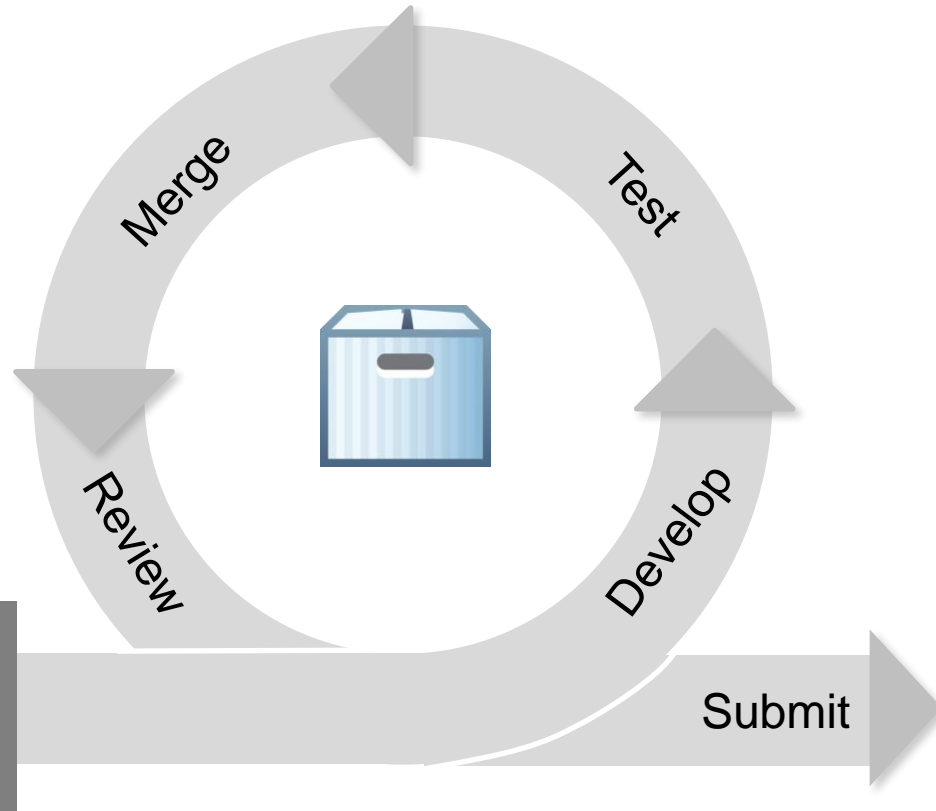
# Jenkins

Continuous Integration & Test





# Strategy: Continued investments to facilitate Continuous Integration as a critical lynch-pin in Agile workflows

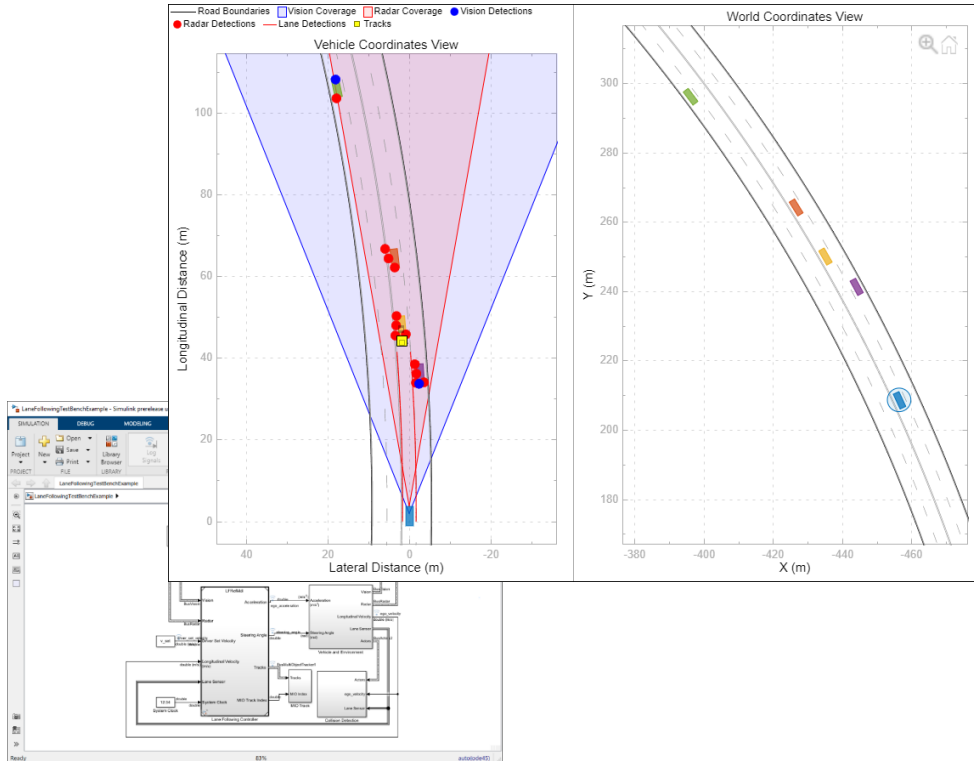


# Can I do CI today in Simulink?

Yes, lets consider an example from

**R2019b**

# Lane Following Assist Example



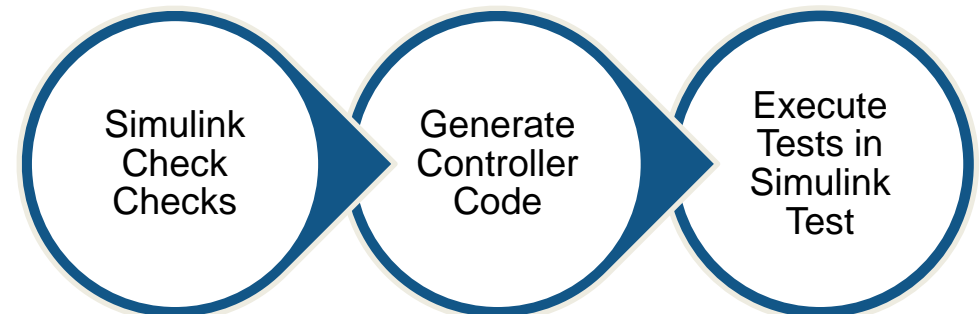
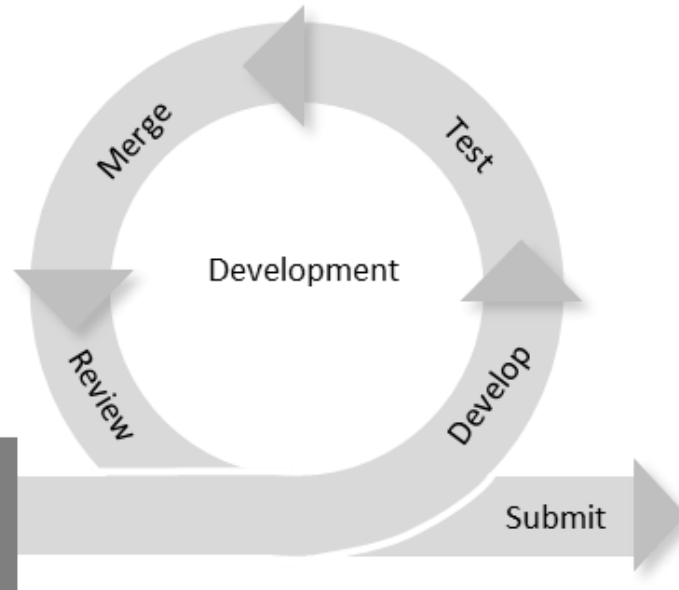
Simulink Check Checks

SIL Code Generation

SIL Testing Simulink Test

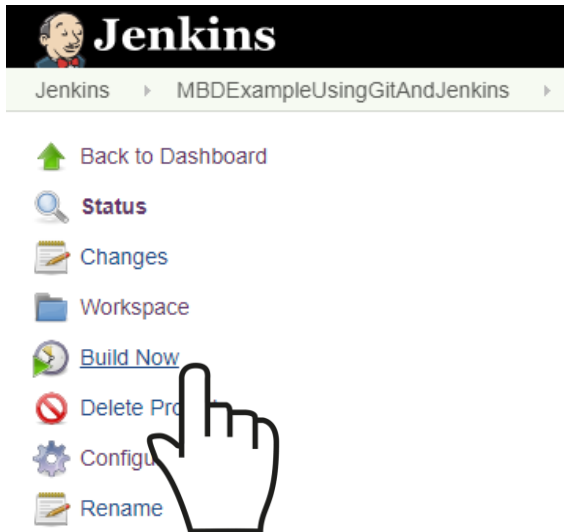


# How Does It All Fit Together?



# 1. Trigger

Trigger

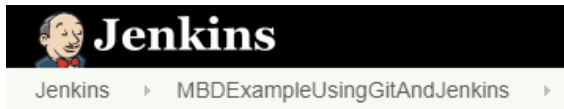


The screenshot shows the Jenkins web interface. At the top, there is a black header with the Jenkins logo (a cartoon character) and the word "Jenkins" in white. Below the header, there is a breadcrumb trail: "Jenkins > MBDEExampleUsingGitAndJenkins >". A list of menu items is displayed on the left side, each with a small icon:

- Back to Dashboard (green arrow icon)
- Status (magnifying glass icon)
- Changes (notepad icon)
- Workspace (blue folder icon)
- Build Now (blue play button icon) - This item is highlighted by a hand cursor.
- Delete Pro (red prohibition sign icon)
- Configu (gear icon)
- Rename (notepad icon)

# 1. Trigger

Continuous  
Integration



Back to Dashboard

Status

Changes

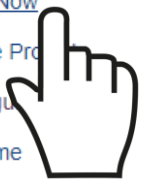
Workspace

Build Now

Delete Pro

Configu

Rename

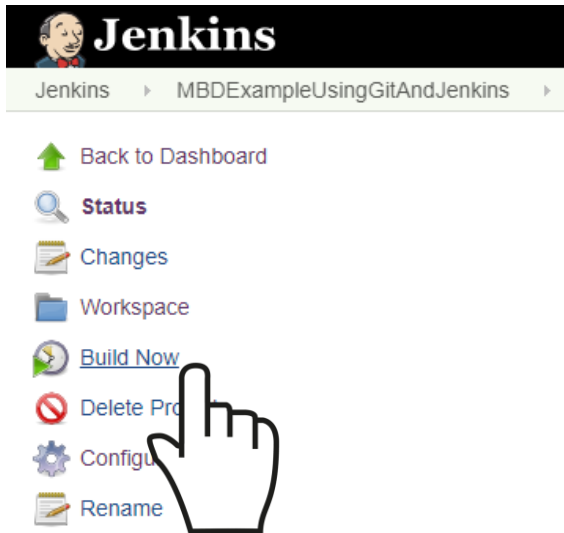
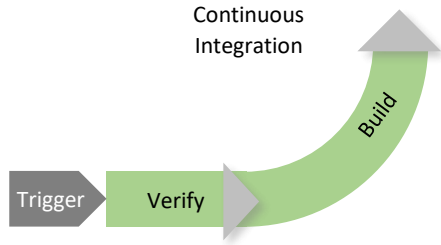


```
Running LaneFollowingModelAdvisorChecks
.
Done LaneFollowingModelAdvisorCheck
```

**Simulink Check**



# 1. Trigger



```

Running LaneFollowingModelAdvisorChecks
.
Done LaneFollowingModelAdvisorCheck
  
```

**Simulink Check**



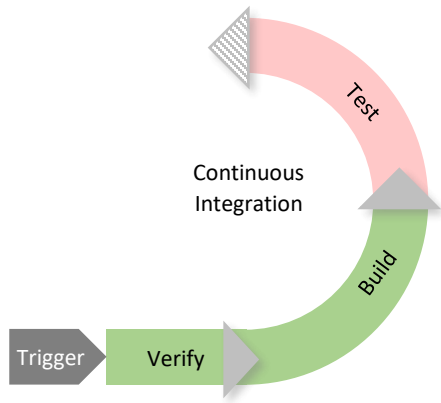
```

## Caching model source code
.....
.....
### Writing header file rtGetNaN.h
### Writing source file rtGetNaN.cpp
### Writing header file rt_defines.h
### Writing header file rt_nonfinite.h
### Writing source file rt_nonfinite.cpp
  
```

**Code Generation**



## 2. Detect



### Failure Summary:

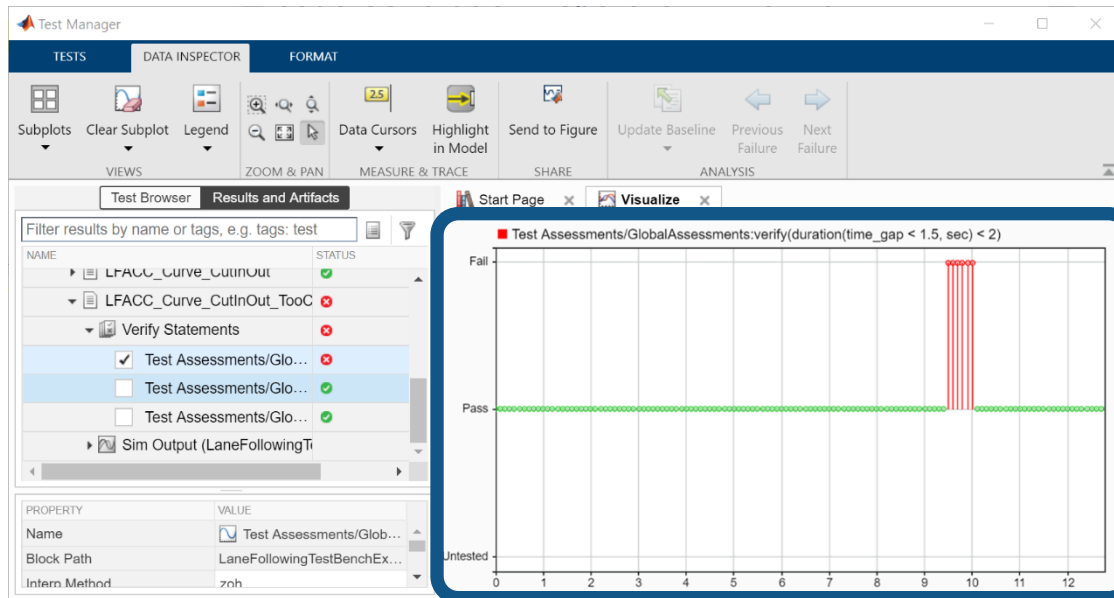
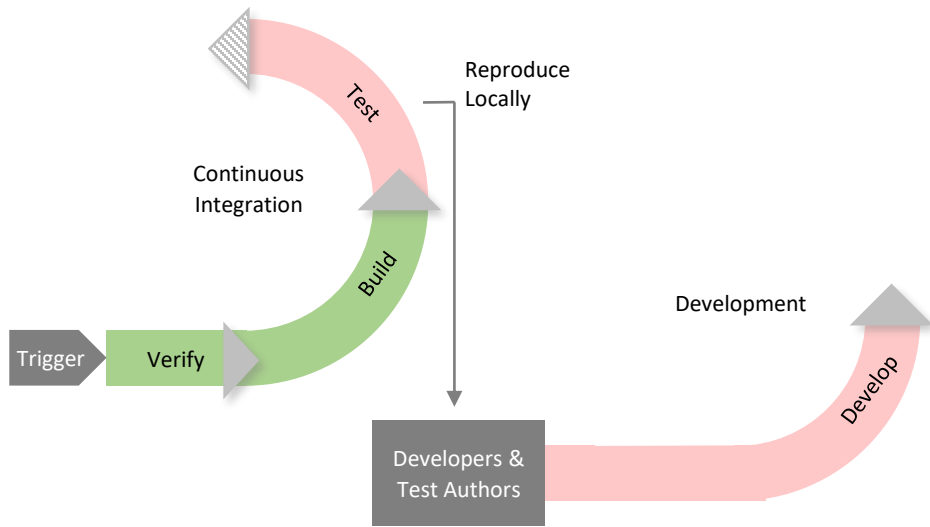
Name	Failed	Incomplete	Reason(s)
===== LaneFollowingTestScenarios > Scenarios/LFACC_Curve_CutInOut_TooClose	X		Failed by verification.
ERROR: MATLAB error Exit Status: 0x00000001			
Build step 'Run MATLAB Tests' changed build result to FAILURE			
Finished: FAILURE			

Simulink Test

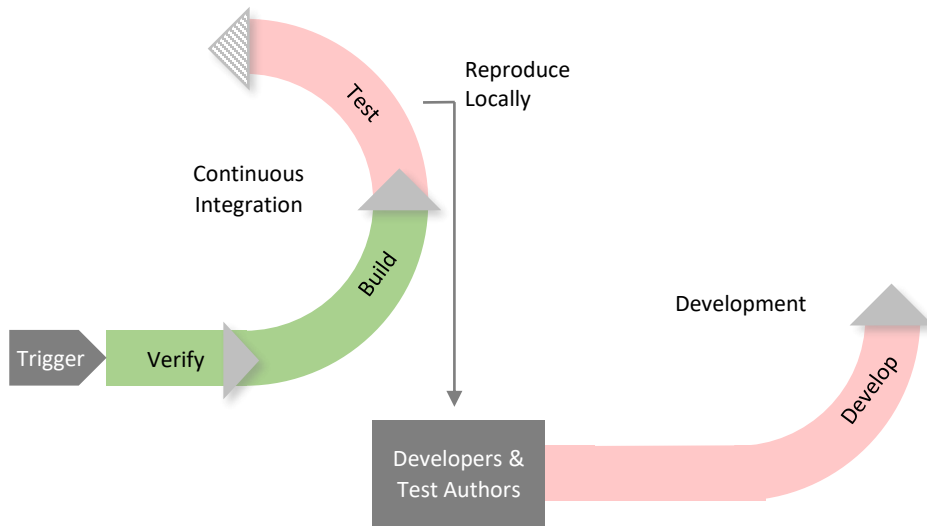




# 3. Reproduce



# 4. Fix Locally

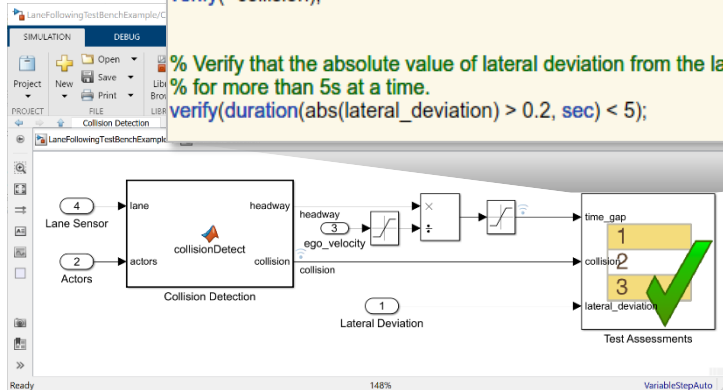


## GlobalAssessments

% Ensure that the time gap between the ego vehicle and lead vehicle does not dip below 1.5s for more than 2s at a time.  
`verify(duration(time_gap < 1.5, sec) < 2);`

% Verify that no collision was detected  
`verify(~collision);`

% Verify that the absolute value of lateral deviation from the lane centerline does not exceed 0.2m for more than 5s at a time.  
`verify(duration(abs(lateral_deviation) > 0.2, sec) < 5);`



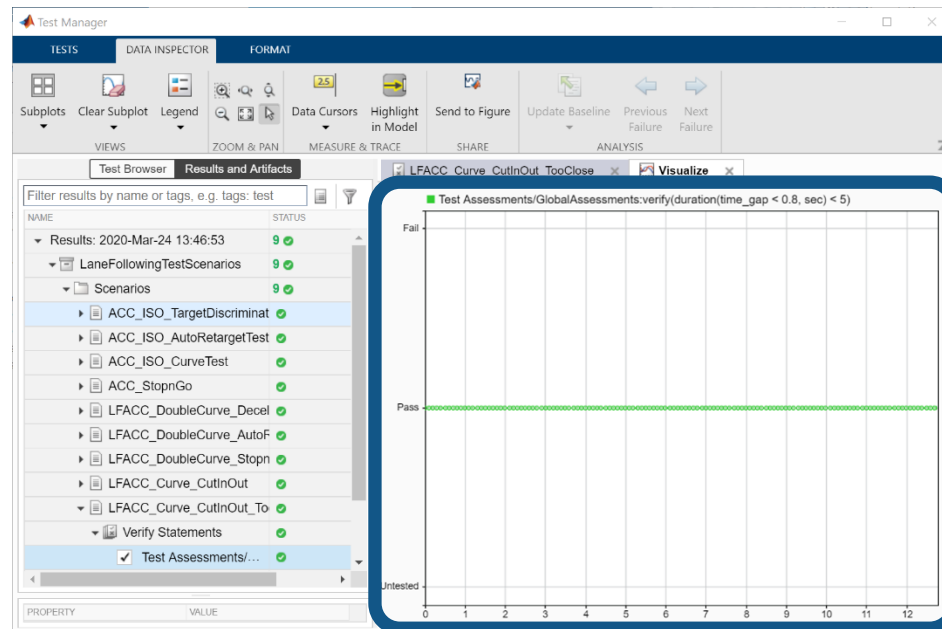
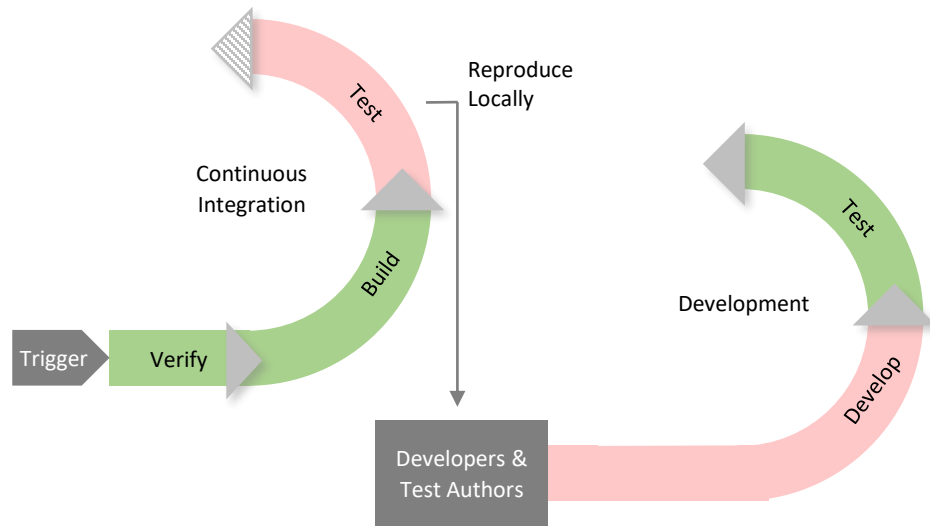
## GlobalAssessments

% Ensure that the time gap between the ego vehicle and lead vehicle does not dip below 0.8s for more than 5s at a time.  
`verify(duration(time_gap < 0.8, sec) < 5);`

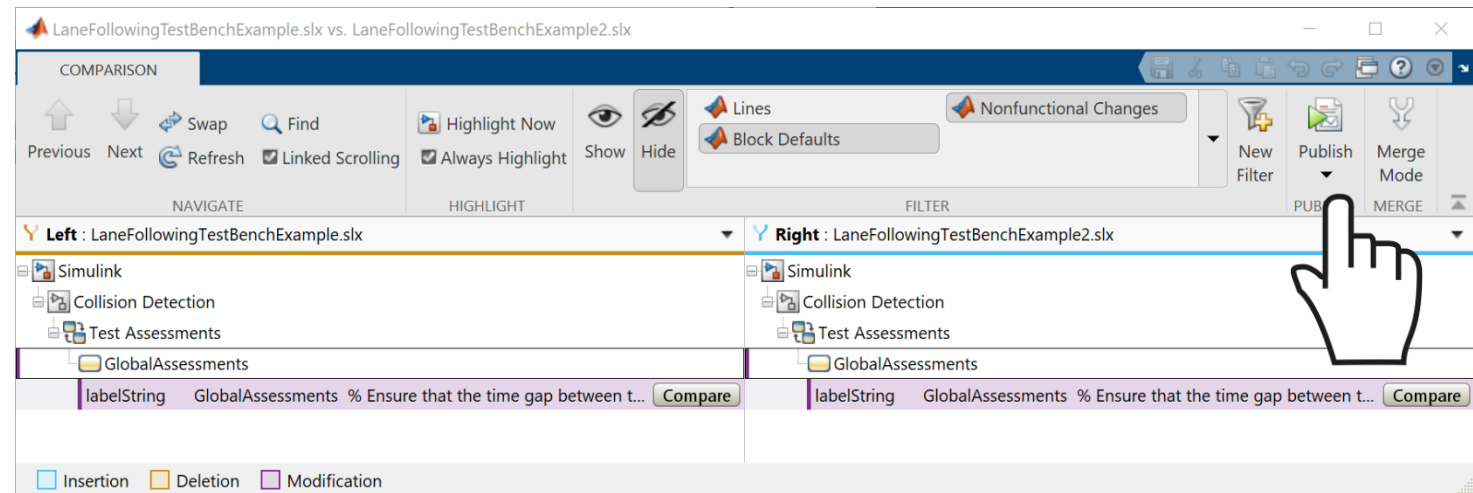
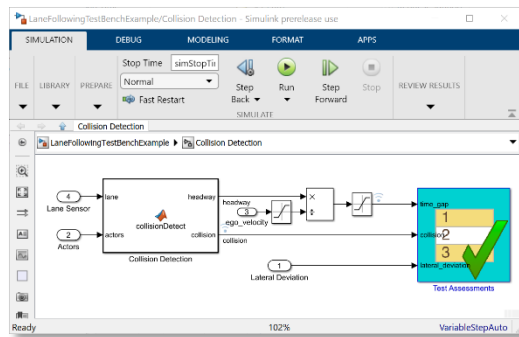
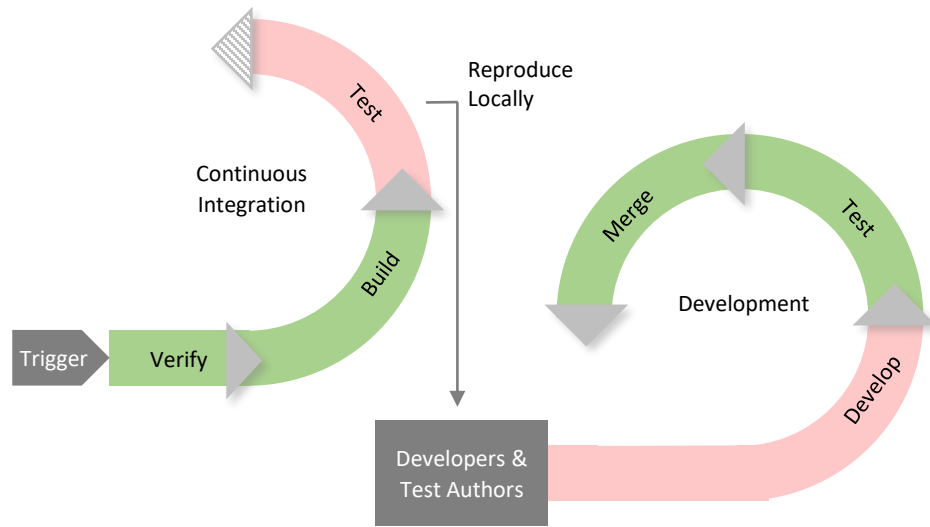
% Verify that no collision was detected  
`verify(~collision);`

% Verify that the absolute value of lateral deviation from the lane centerline does not exceed 0.2m for more than 5s at a time.  
`verify(duration(abs(lateral_deviation) > 0.2, sec) < 5);`

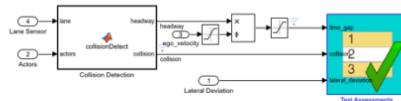
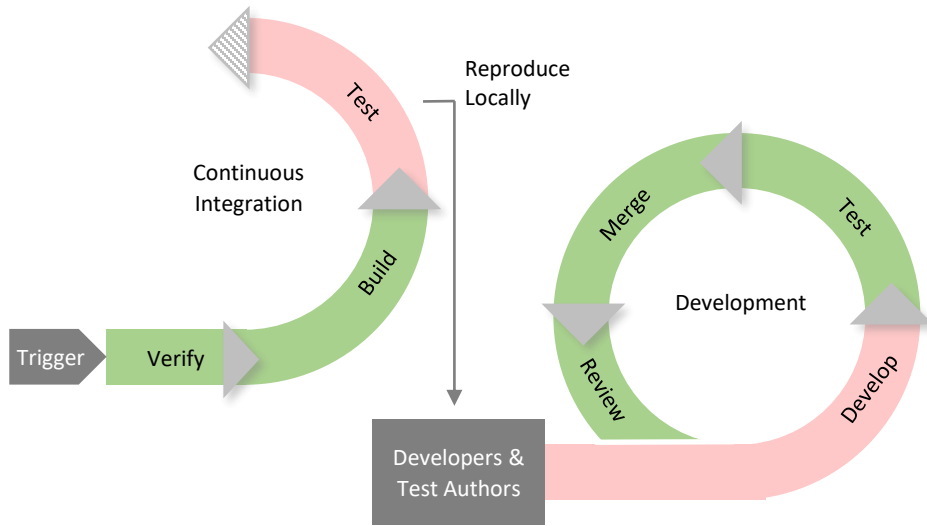
# 5. Test Locally



# 6. Merge



# 6. Review



LaneFollowingTestBenchExample/Collision Detection/Test Assessments

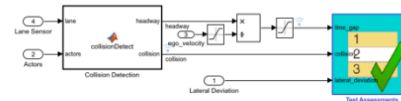
### GlobalAssessments

labelString : GlobalAssessments

% Ensure that the time gap between the ego vehicle and lead vehicle does not dip below % 1.5s for more than 2s at a time.

`verify(duration(time_gap < 0.8, sec) < 2);`

% Verify that no collision was detected  
`verify(~collision);`



LaneFollowingTestBenchExample/Collision Detection/Test Assessments

### GlobalAssessments

labelString : GlobalAssessments

% Ensure that the time gap between the ego vehicle and lead vehicle does not dip below % 1.5s for more than 5s at a time.

`verify(duration(time_gap < 0.8, sec) < 5);`

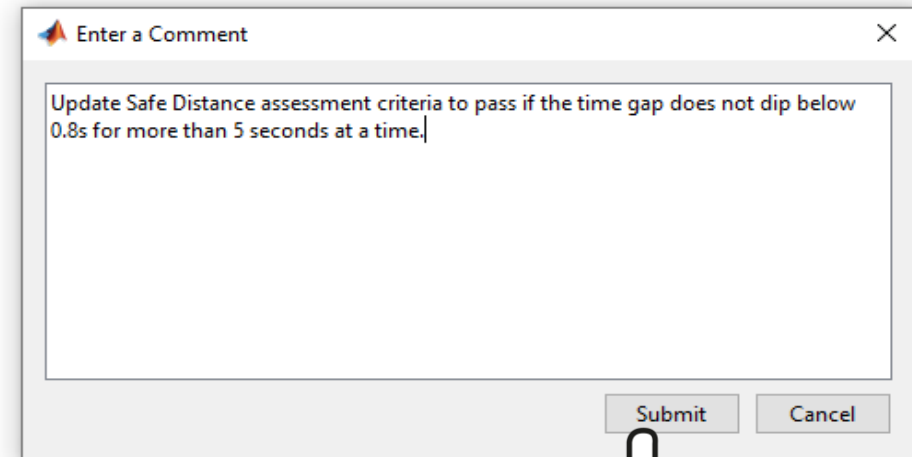
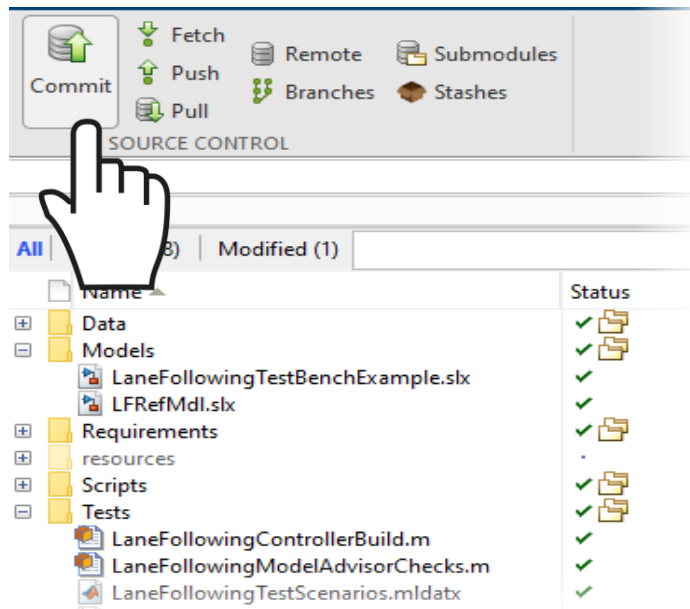
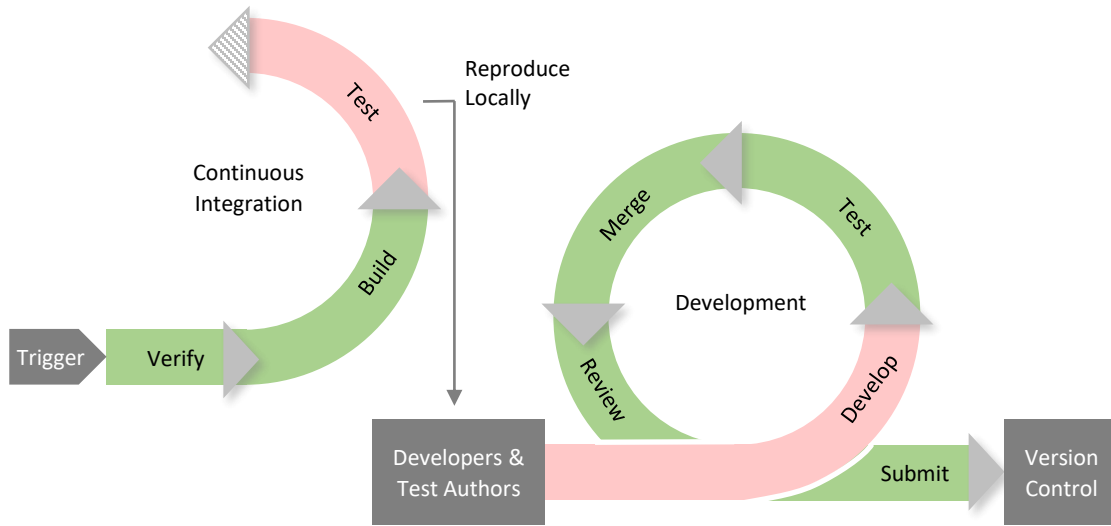
% Verify that no collision was detected  
`verify(~collision);`



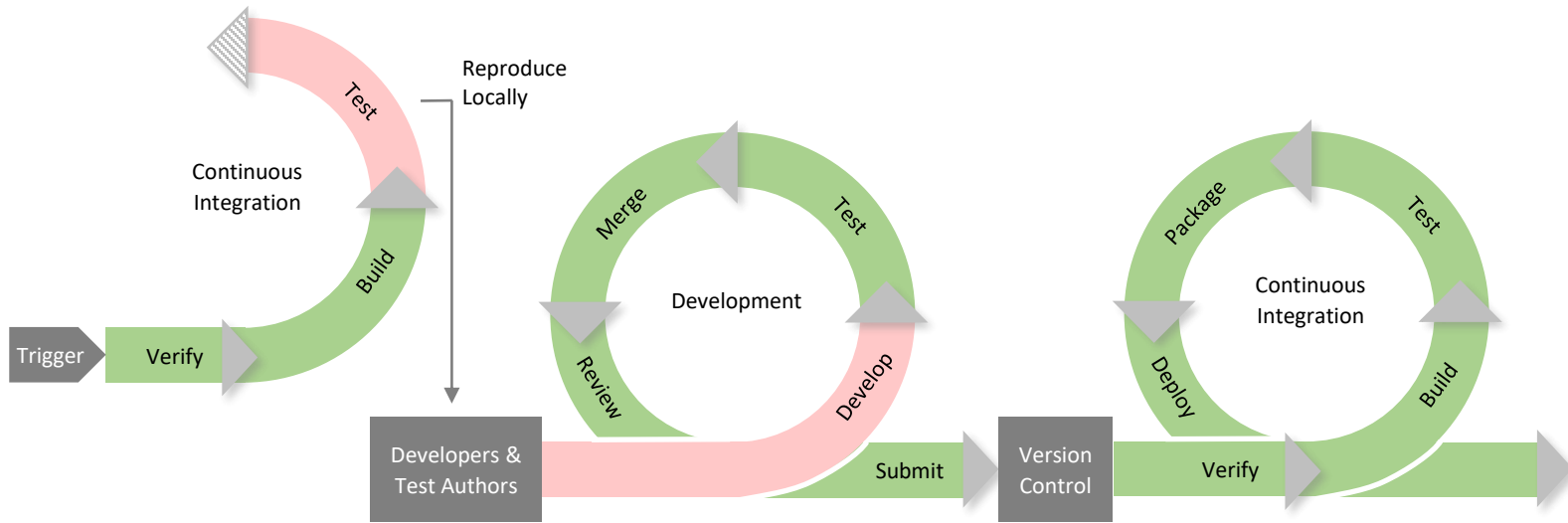
David Boissy 12 minutes ago  
Ship It!

[Reply](#) [Resolve](#)

# 7. Commit



# 8. Verify, Build, Test




**Finished: SUCCESS**




# Continuous Integration Success is within your reach

1

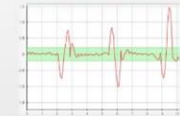
## Tooling



Jenkins  
Plugin




MATLAB  
Unit



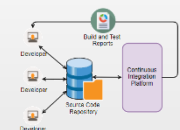
Simulink  
Test

2


## Documentation



Technical  
Article




Documentation  
Hub




Solutions  
Page

3

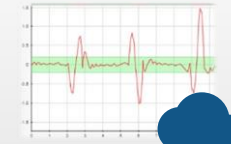
## Future



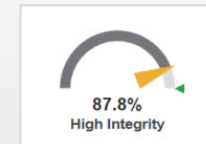
Pipeline



Server  
Workflows



Test Results  
Online



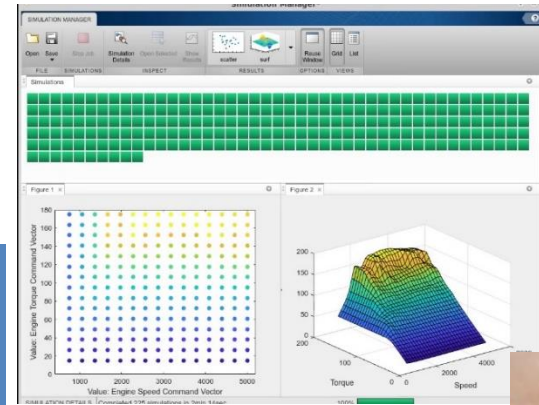
Dashboards



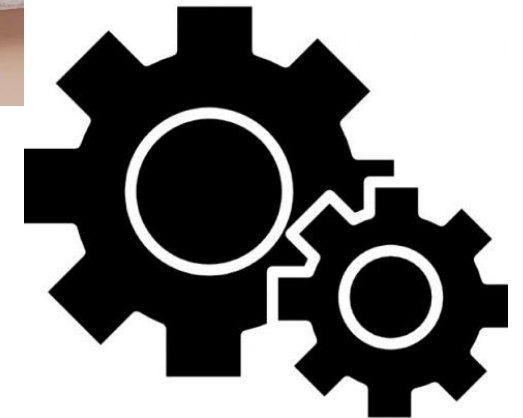
# Lets go back to the broad forces that shape our platform evolution

## 1. Simulation Scale

**YOU!**



## 3. Collaborative Engineering



# Q&A

Please contact us with questions



[mani@mathworks.com](mailto:mani@mathworks.com)