MATLAB EXPO

2024.06.11 | 그랜드 인터컨티넨탈 서울 파르나스

자율주행 소프트웨어 개발을 위한 모델기반 시스템 엔지니어링에 대한 사례

이동한 책임연구원, HL Klemove

MathWorks

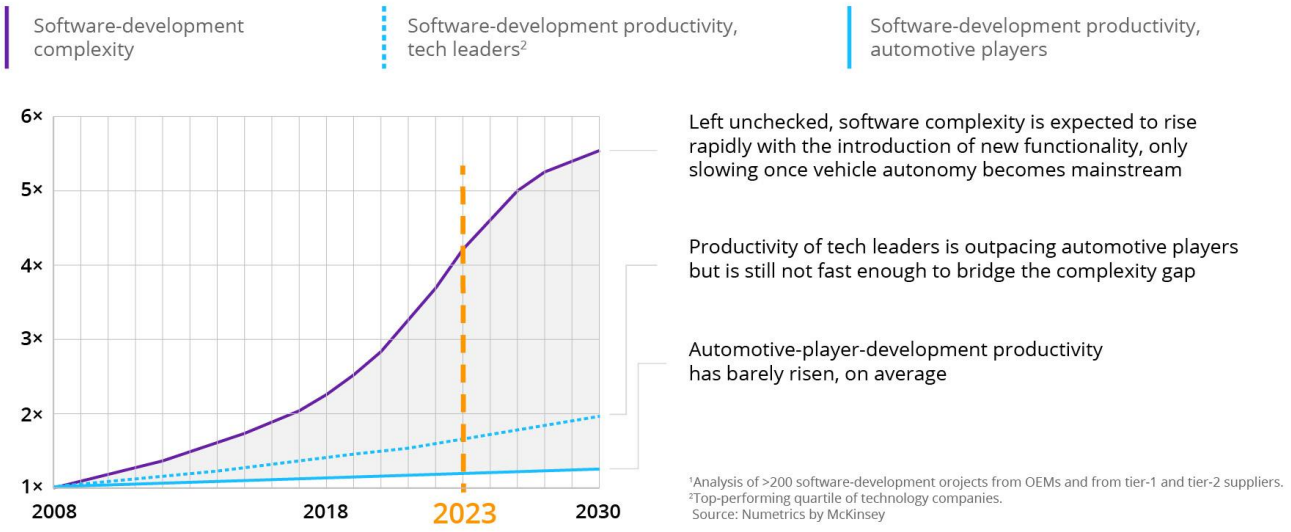
Background

- **Need to Change SW Development Methods**

Growing complexity of automotive software and a demand for **more and more features** in a **shorter timeframe**.

The increase in complexity is **outpacing** our improvements in efficiency

Relative growth over time, for automotive features,[1] indexed, 1 = 2008



[1]Analysis of >200 software-development orojects from OEMs and from tier-1 and tier-2 suppliers.
[2]Top-performing quartile of technology companies.
Source: Numetrics by McKinsey

https://www.sjf.tuke.sk/transferinovacii/pages/archiv/transfer/29-2014/pdf/251-253.pdf

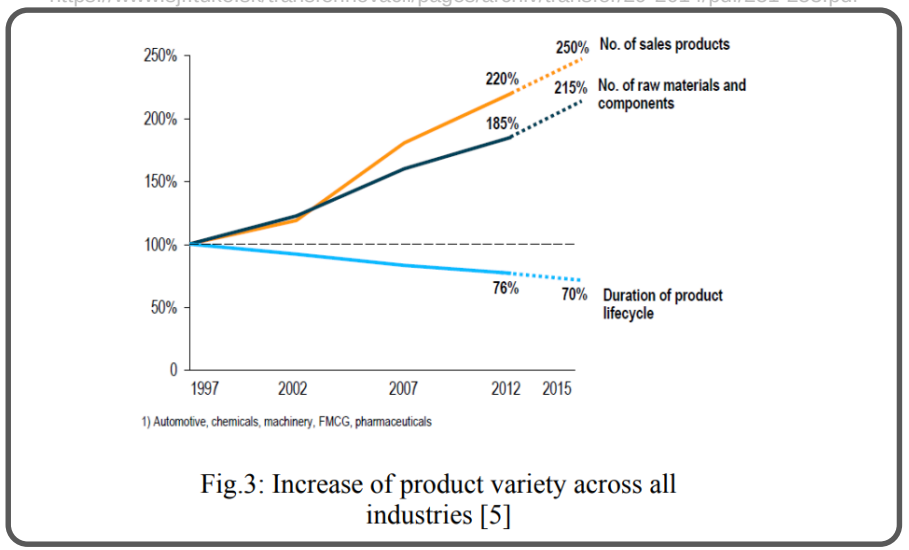
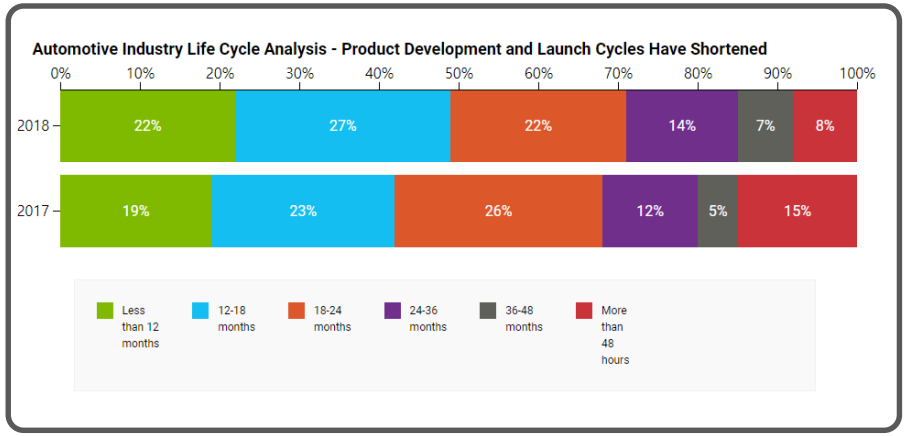


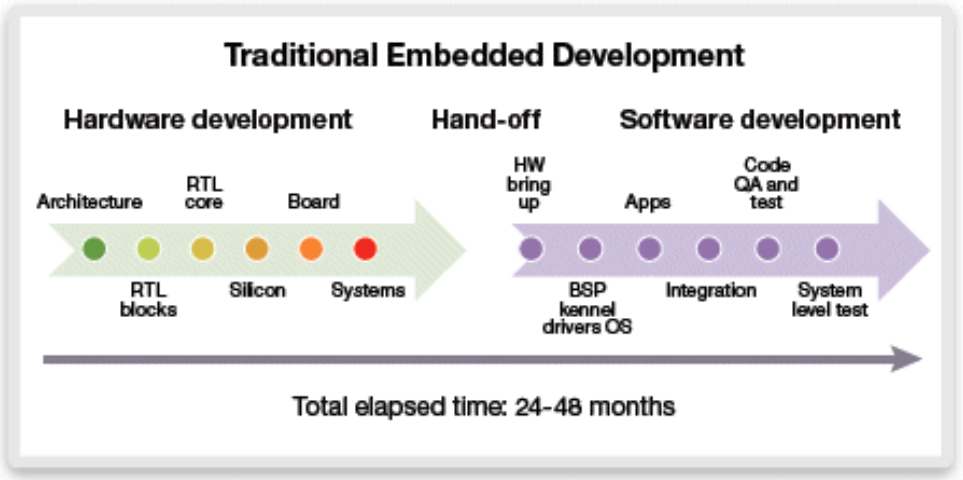
Fig.3: Increase of product variety across all industries [5]

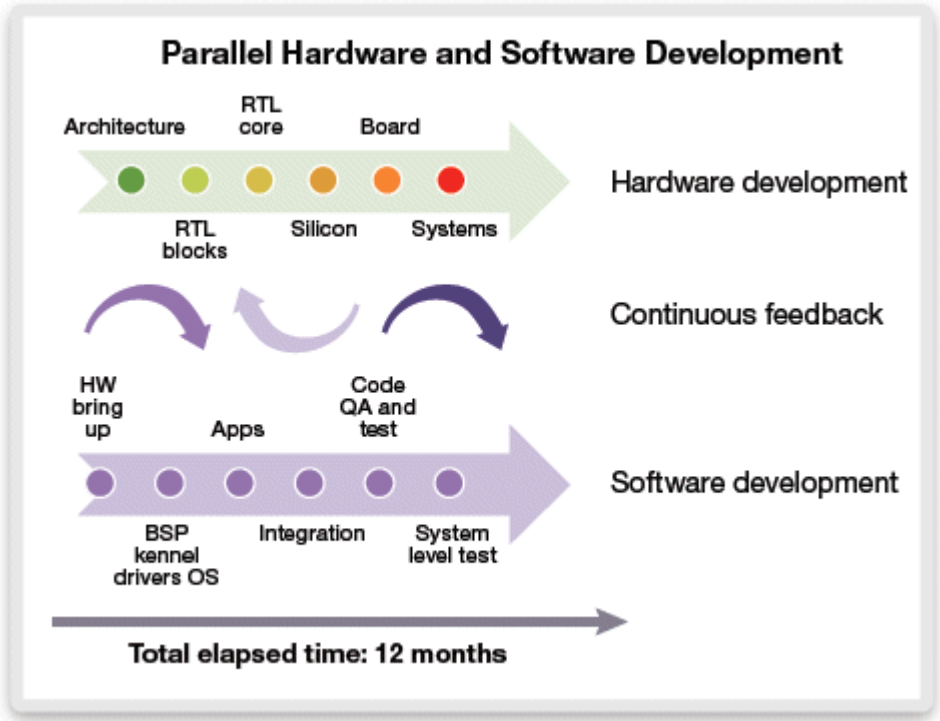


https://www.jabil.com/blog/automotive-industry-trends-point-to-shorter-product-development-cycles.html

Background

- **Parallel Development Process**

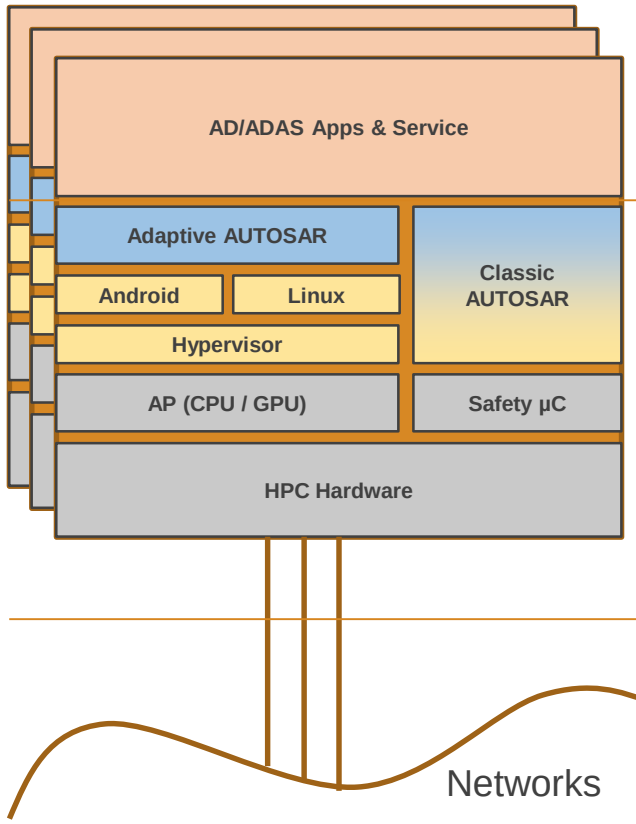




Background

- **Virtualization**

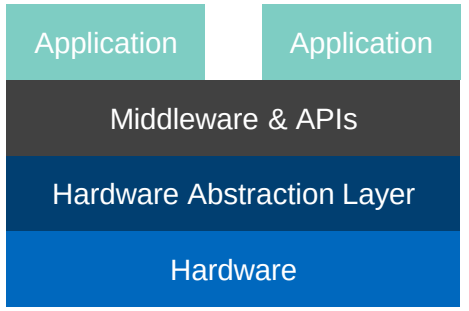
We need to define the API on the software architecture! This build the basis for virtualization



Functions

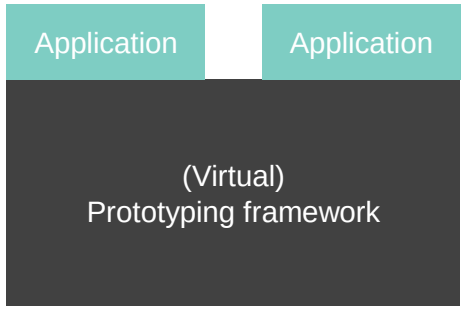
Future : API
on software level

Any (heterogeneous) virtualized middleware, basic software, hypervisor and operating systems



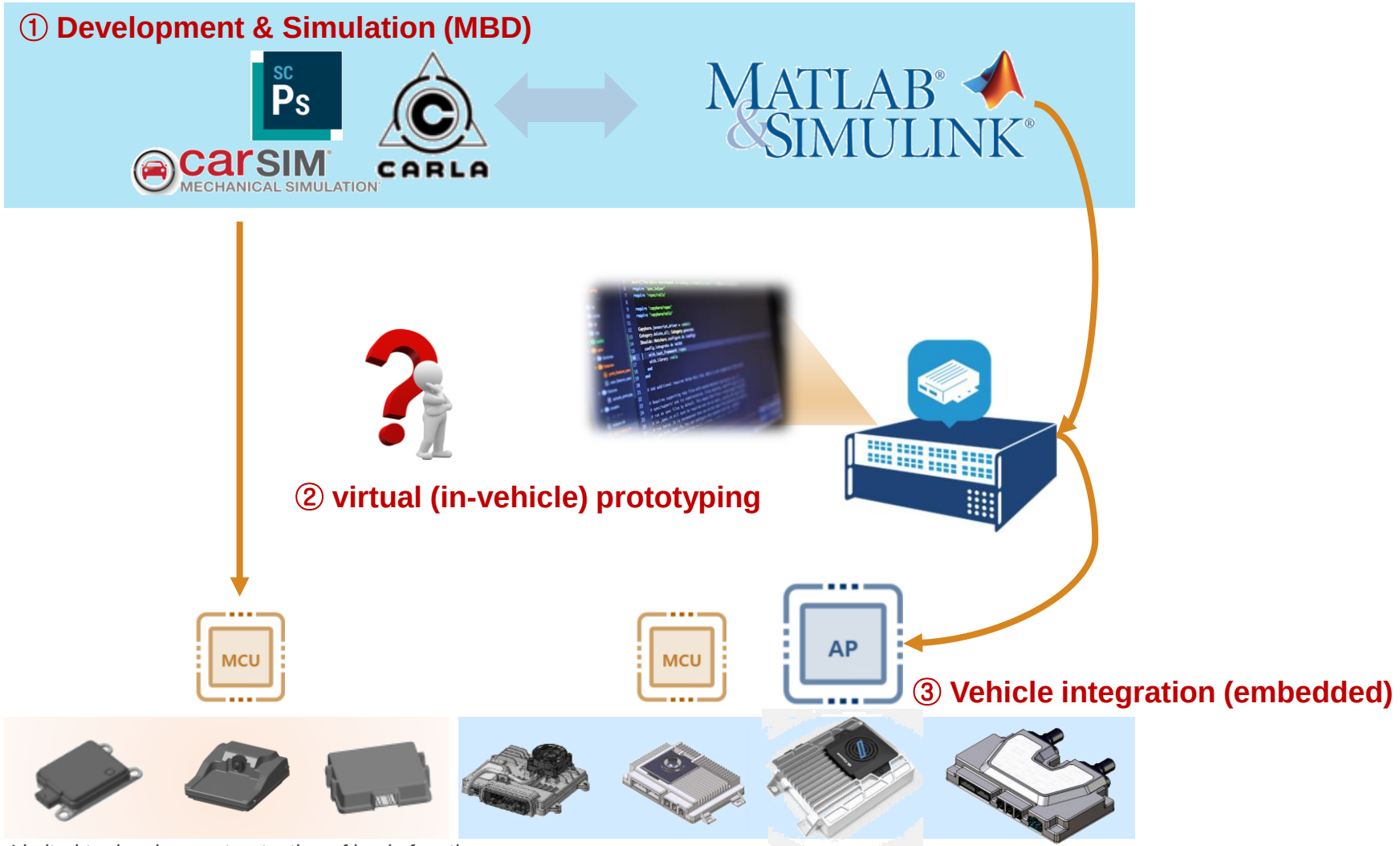
Application can be developed independently of the specific platform

Past : API
on communication level



Background

- **Virtual Prototyping Framework**



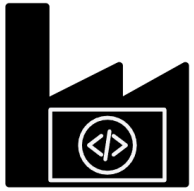
Limited to development or testing of basic functions

Background

- **Development Process with Some Use Cases**

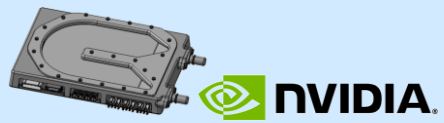
Algorithm Development on cross-platform prototyping framework

- *Purely theoretical and mathematical domain*
- *PC-based development*



Embedded Software Development on embedded target

- *Depends on hardware architecture*
- *Performance and power consumption are essential*

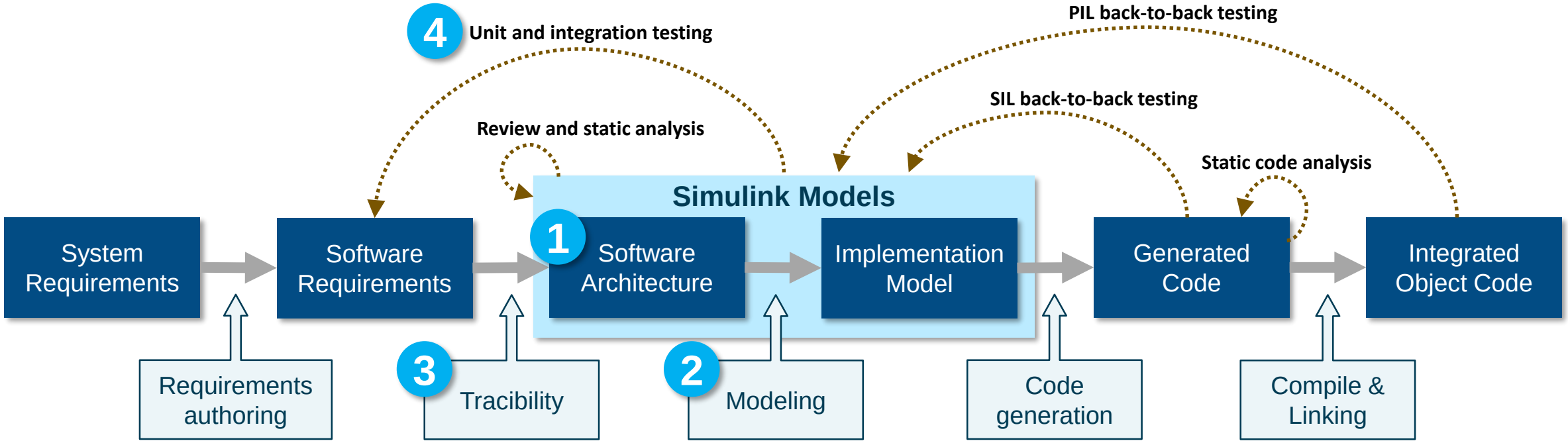


Prototyping framework

ADAS Domain Controller

Challenge

- **Workflow using Model-Based Design**



1 **Software architecture authoring & management**

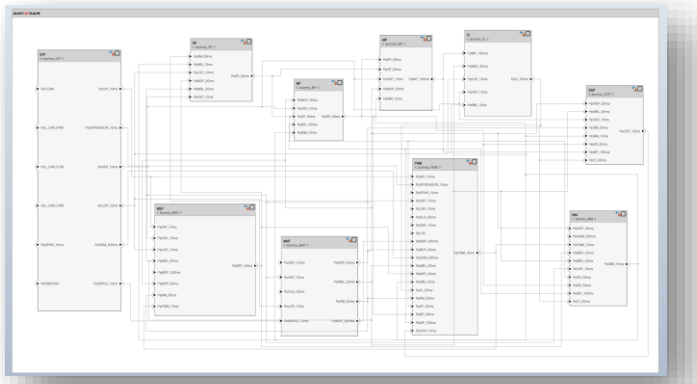
2 **Manually updated ROS wrapper**

3 **Ensuring tracibility between requirements and software**

4 **Simulation-in-the-Loop**

Solution

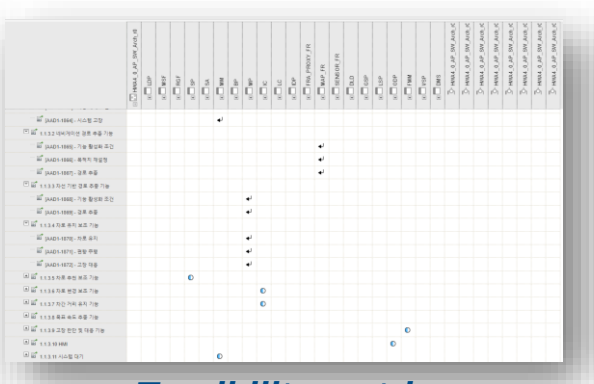
1. **SW Architecture Authoring using System Composer & AUTOSAR Component Designer**
2. **Customized Wrapper Generator**
3. **Tracibility using Requirements Toolbox**
4. **Improve Speed & Flexibility of SILs with Linux Runtime Manager**



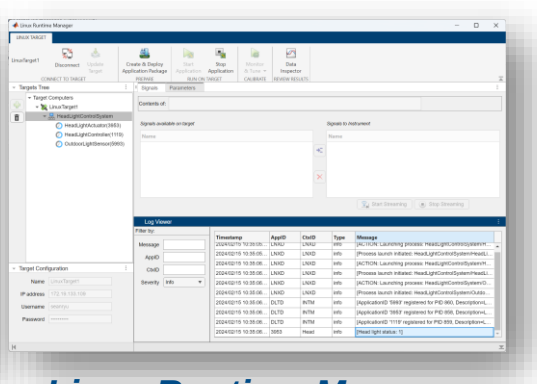
System Composer

```
%% configurations begore code gen
NodeName = "BP"
NodeHz = "20"

modelName = 'HWA_Architecture'; %append(NodeName,'_Wrapper_new_240108');
profileName = 'SW_Arch_Profile';

if NodeName == "MP"
    WrapperCFile = append('MotionPlanner_Wrapper_ROS.cpp');
    WrapperHeader = append('MotionPlanner_Wrapper_ROS.h');
elseif NodeName == "BP"
    WrapperCFile = append('BehaviorPlanner_Wrapper_ROS.cpp');
    WrapperHeader = append('BehaviorPlanner_Wrapper_ROS.h');
else
    WrapperCFile = append(NodeName,'_Wrapper_ROS.cpp');
    WrapperHeader = append(NodeName,'_Wrapper_ROS.h');
end
WrapperFolderName = 'wrappergen';


codeGenDirName = append(modelName,'_ert_rtw');
ProjectInfo = currentProject;  % Simulink Project Information object
CodeGenFolder = ProjectInfo.SimulinkCodeGenFolder; % code generation folder
WorkFolder = fileparts(CodeGenFolder); % working folder
```

Customized Wrapper Generator

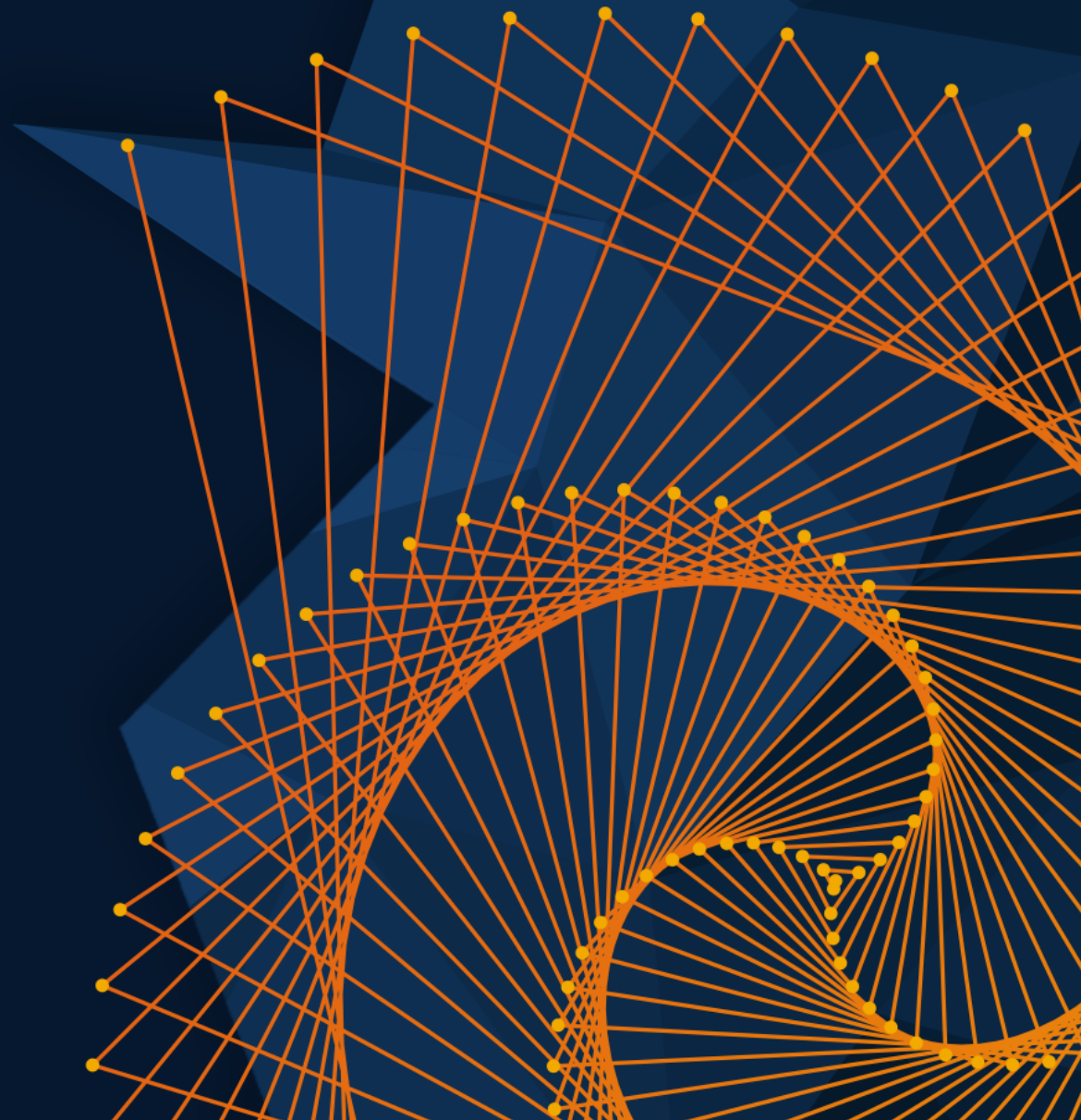


Tracibility matrix



Linux Runtime Manager

MATLAB EXPO



MathWorks®

© 2024 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.