

# 2022 MathWorks 中国汽车年会

软件定义汽车 - 重塑汽车软件开发体系

龚小平 迈斯沃克中国

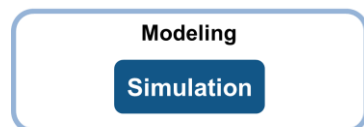
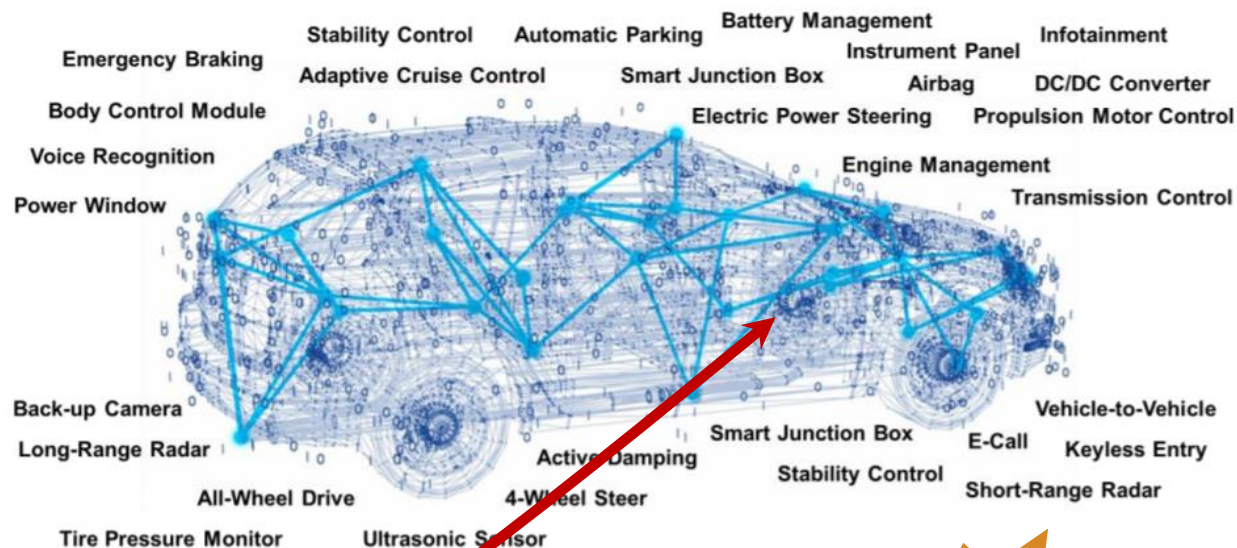


# 内容

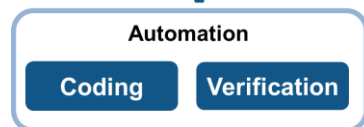
- 软件定义汽车的挑战和机遇
- 重塑汽车软件开发体系
- 成为行业的可靠合作伙伴



# 汽车行业的第一次数字转型 – 软件提升硬件



Fast repeatable tests



Fast agile development loops

基于模型设计



- 设计：利益相关方无法有效进行评审
- 实现：错误发现过晚导致修复成本高
- 人员：汽车工程师不熟悉软件开发

1980

1990

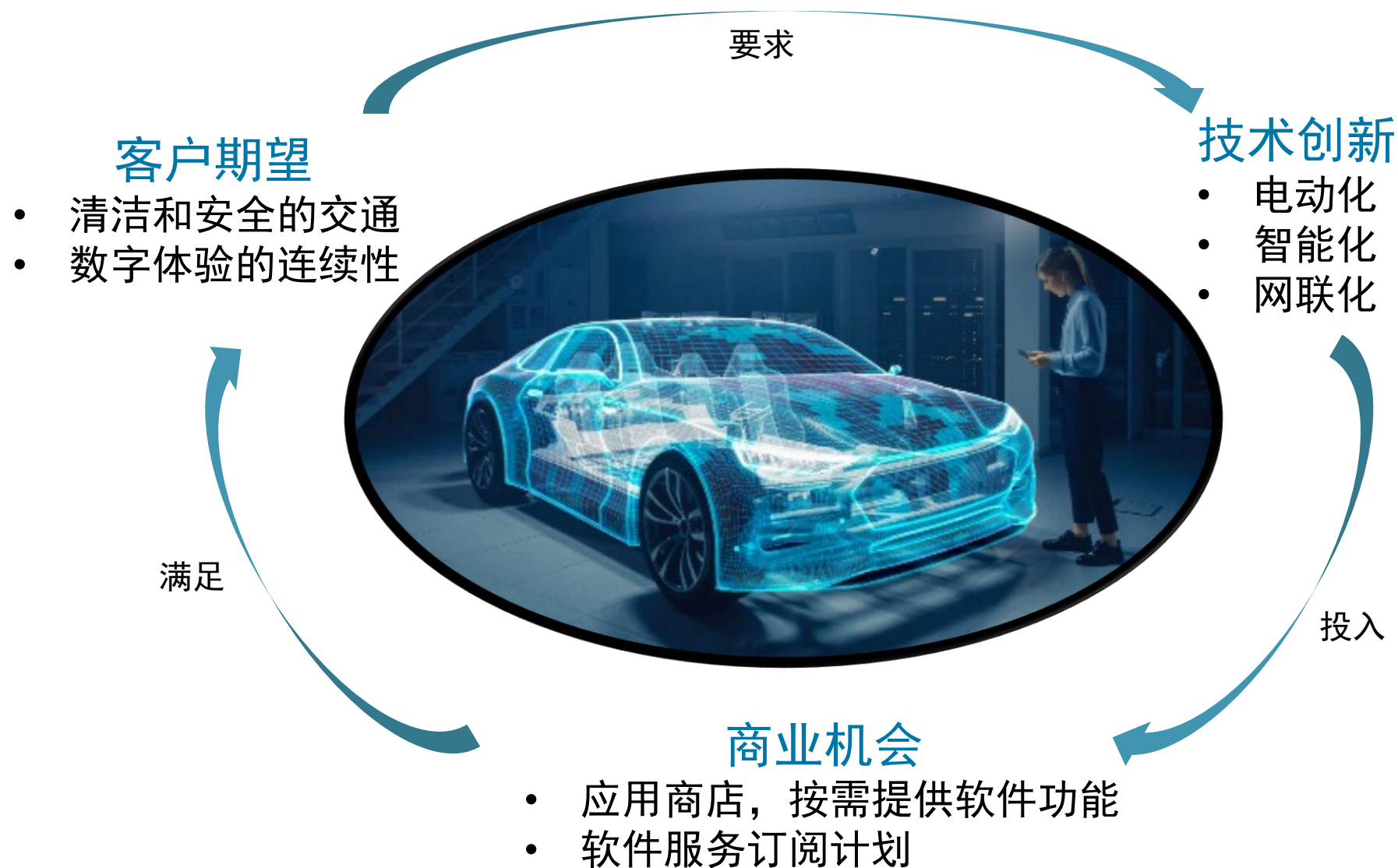
2000

2010

2020

2030

# 汽车行业的第二次数字转型 – 软件定义汽车



# 软件定义汽车的机遇 – 重塑汽车软件开发体系

## 关键挑战

- 缺少跨学科专业知识
- 硬件原型不足且昂贵
- 缺陷发现时间太晚
- 如何符合行业标准

## 颠覆性创新

- 复杂软件
- 数据管理
- AI专业知识
- 数字&工程

系统工程

软件工厂

数据驱动

虚拟车辆

# 软件定义汽车的关键 – 整合系统、软件和数据

- 如何有效评估架构
- 如何开展权衡分析
- 如何进行组件选型
- 如何构建方案原型
- 如何前置集成测试

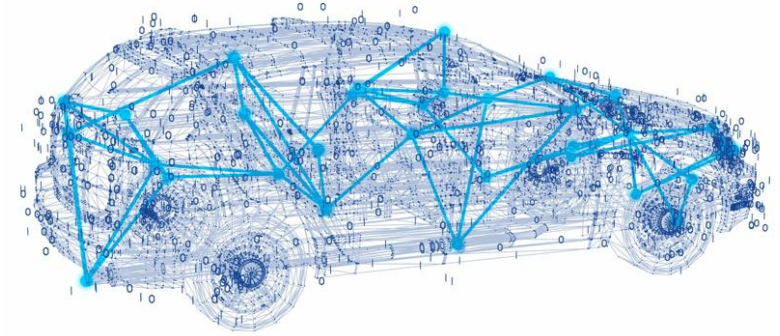


- 如何确保实现所有需求
- 如何提高软件重用率
- 如何提升交付质量
- 如何提高设计迭代效率
- 如何提升自动化流程

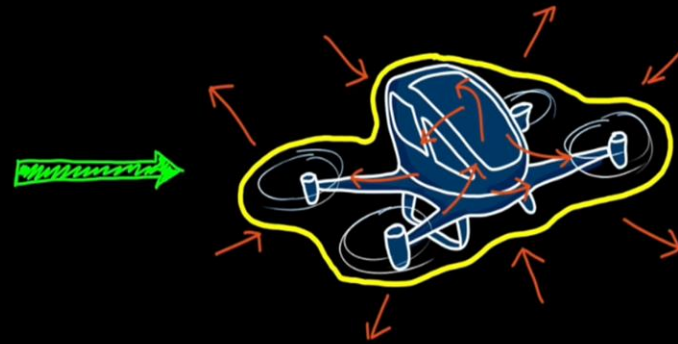
- 如何进行预处理和筛选
- 如何提取特征和业务价值
- 如何实现AI算法的训练
- 如何部署云端解决方案



# 系统工程 – 应对复杂多学科系统设计

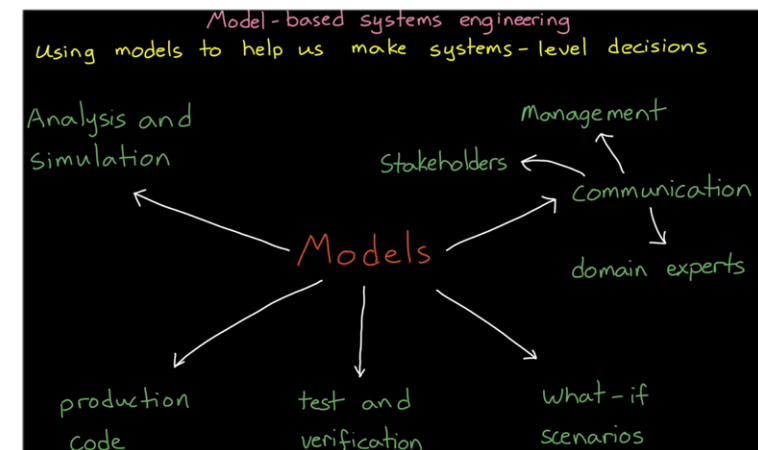
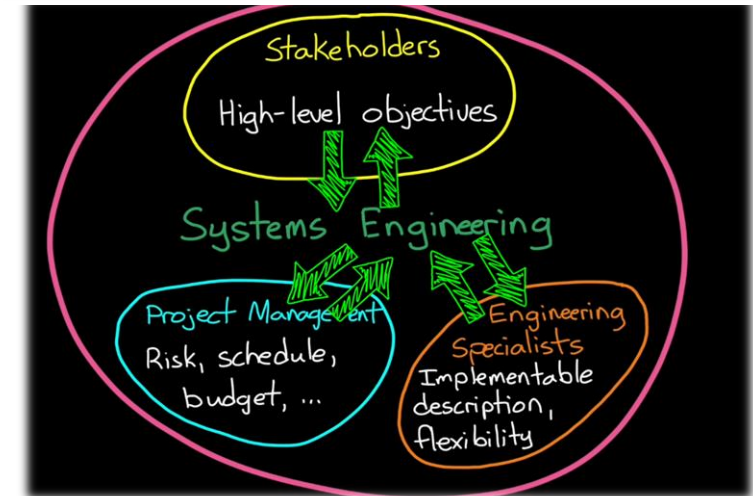


Systems engineering is a process that we can use to develop something that is too complex to design and build as a single monolithic entity.



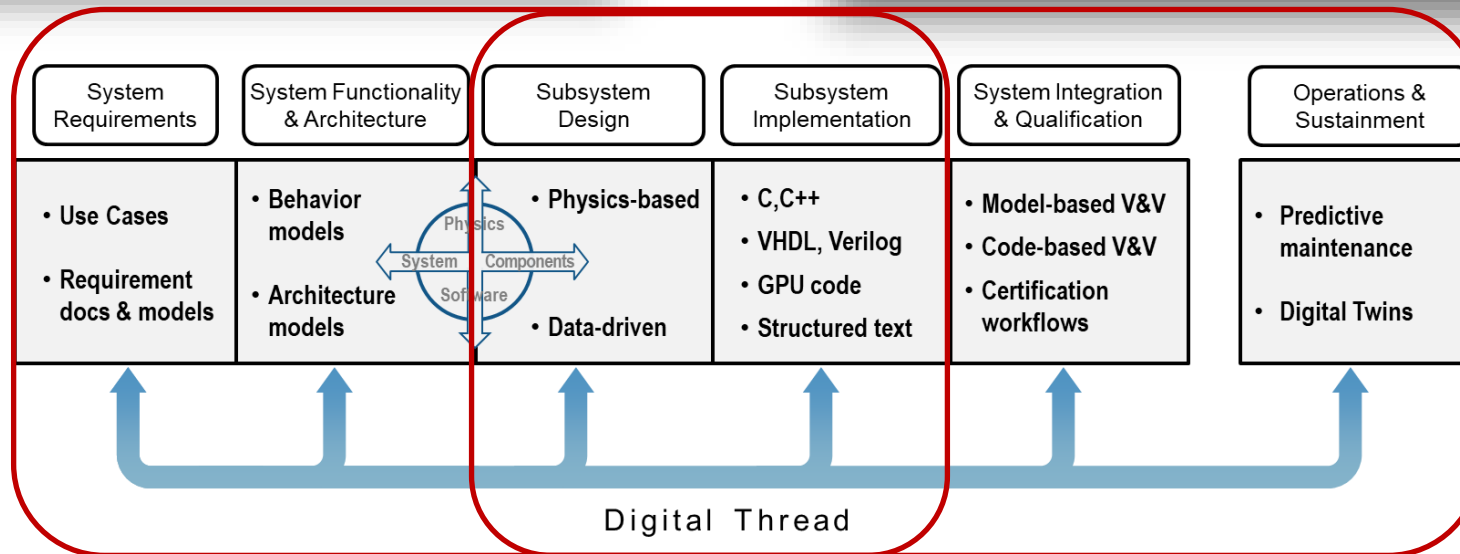
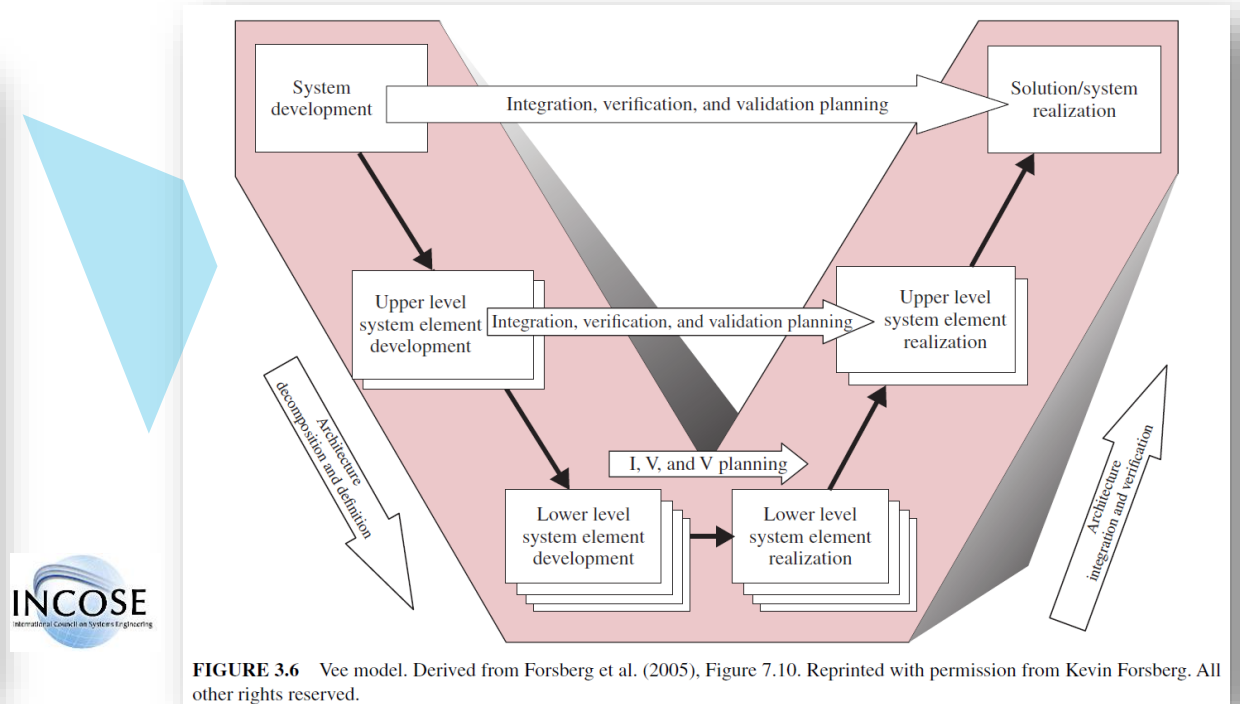
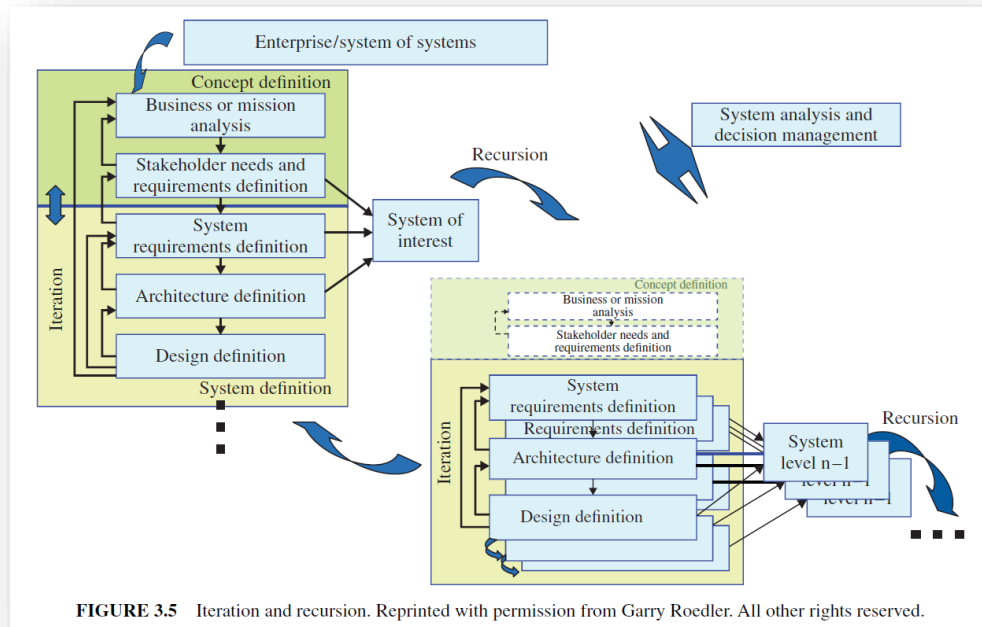
Idea

It's guiding the engineering process so that the system can be implemented and meets the needs of the project



[MATLAB TECH TALKS: What Is Systems Engineering?](#)

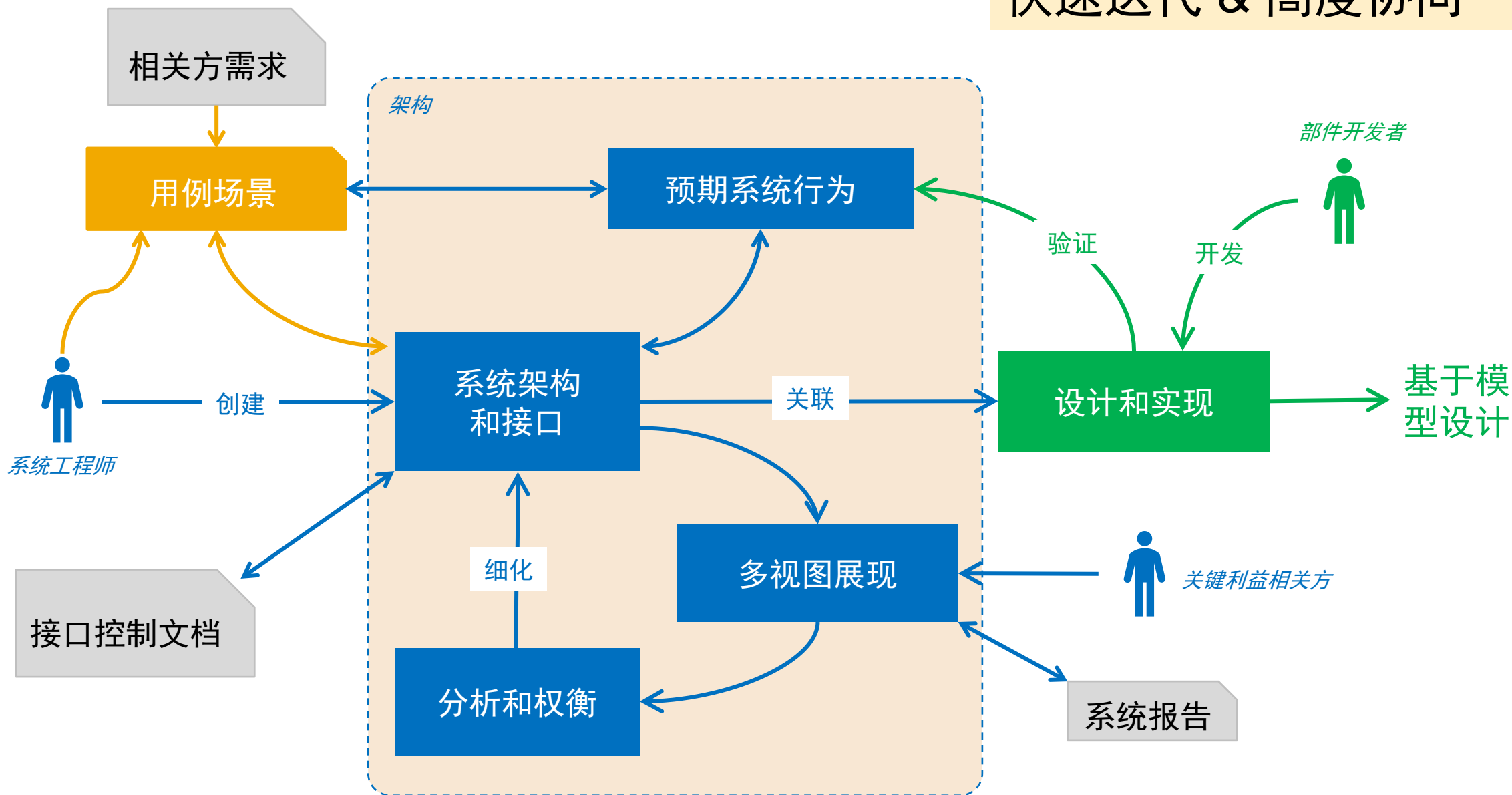
# 系统工程 – 从MBD到MBSE





# 系统工程 – MBSE参考流程

快速迭代 & 高度协同



# 系统工程 – MBSE的工作框架

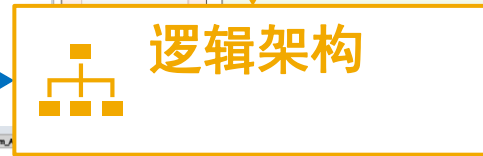


为什么



是什么

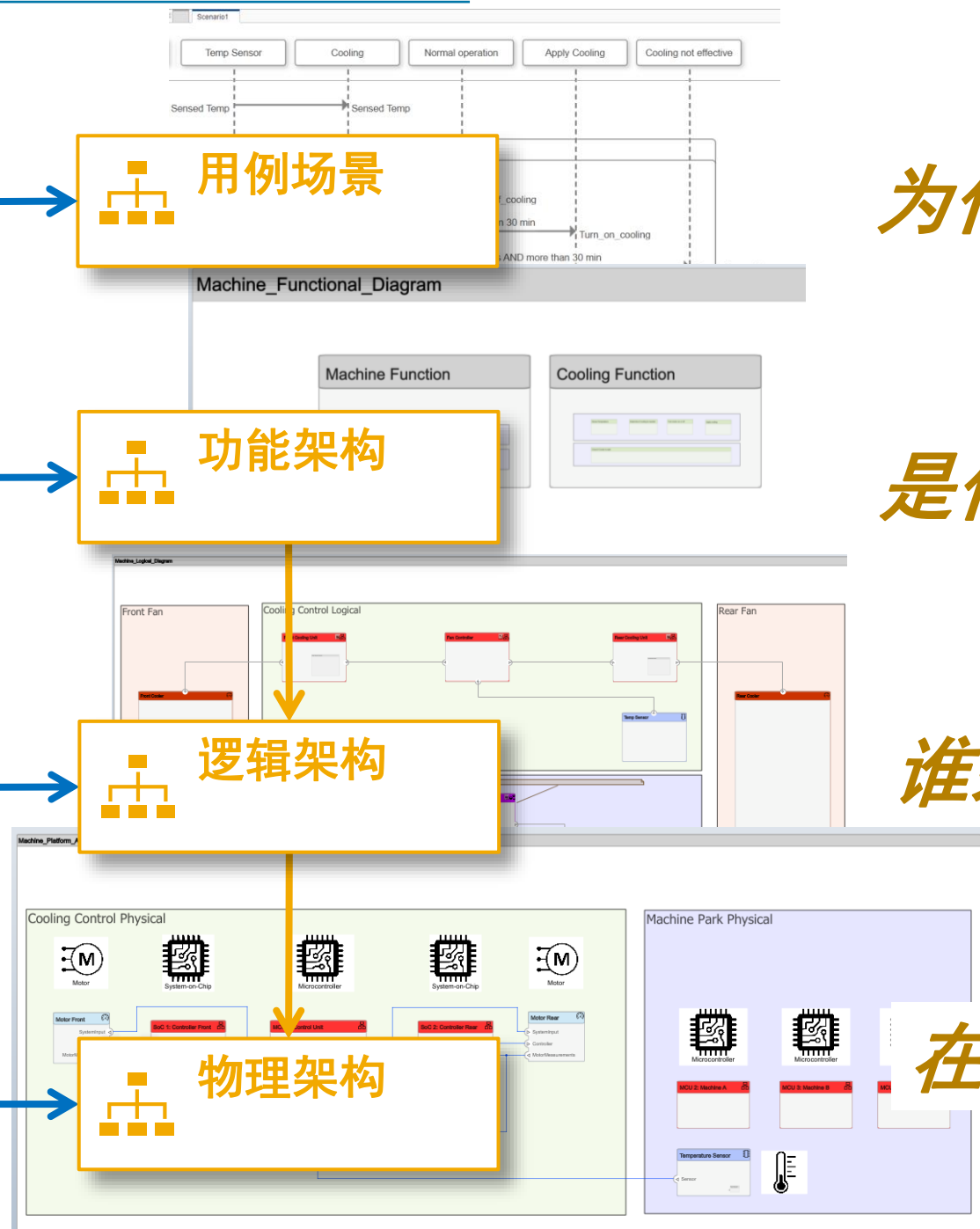
- (R)equirement
- (F)unctional
- (L)ogical
- (P)hysical



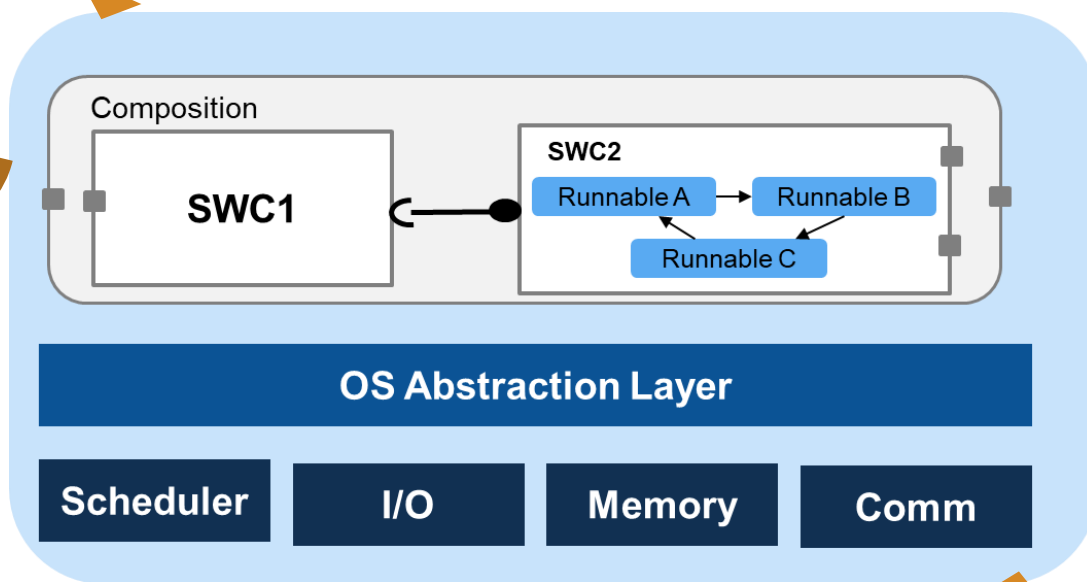
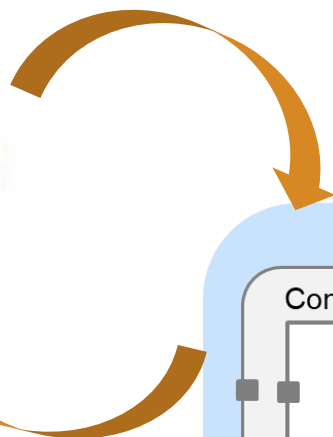
谁来做



在哪做

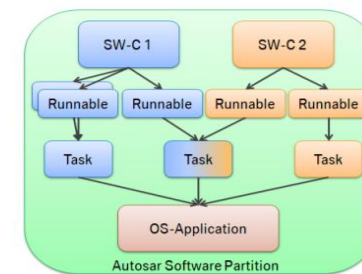


# 系统工程 – 从系统架构到软件架构

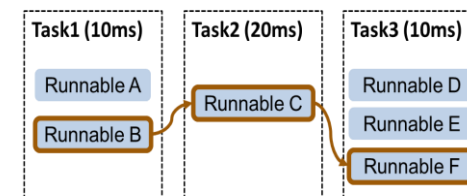


Service Interface
Property 1
Method1()
Method2()
Event1
Event2

服务接口

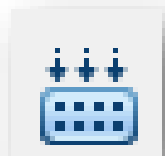
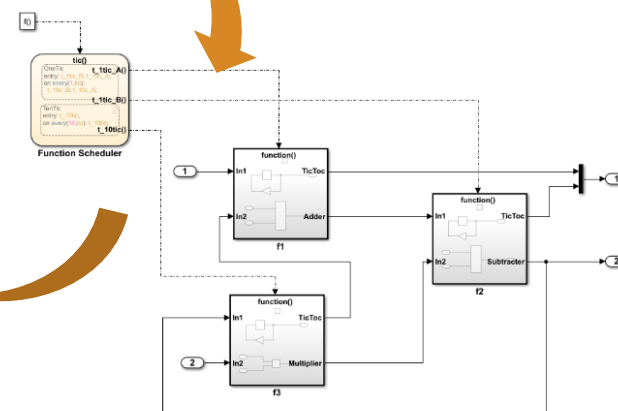


函数/任务/进程



Effect Chain: B -> C -> F < 20 ms

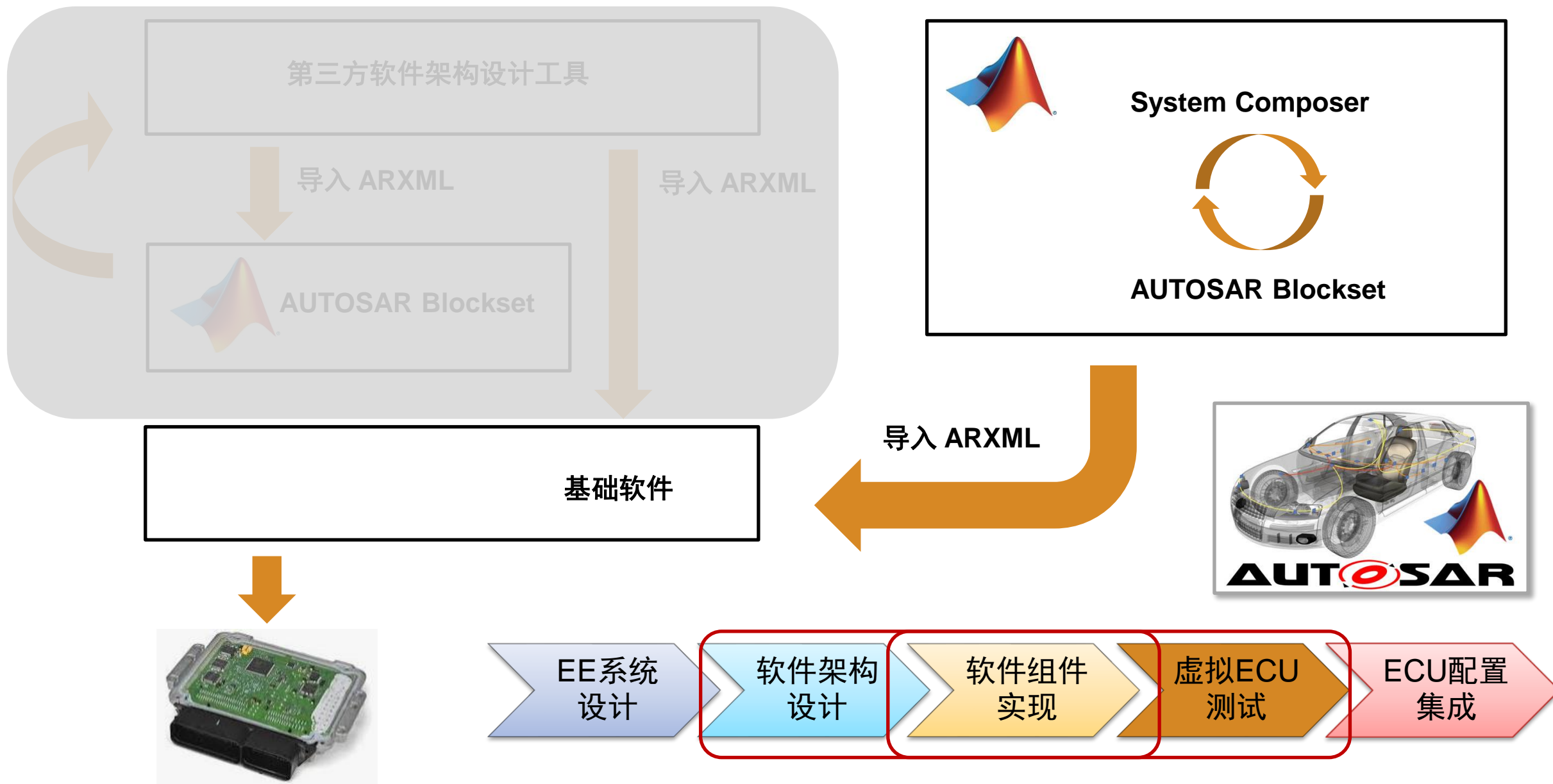
时序/调度



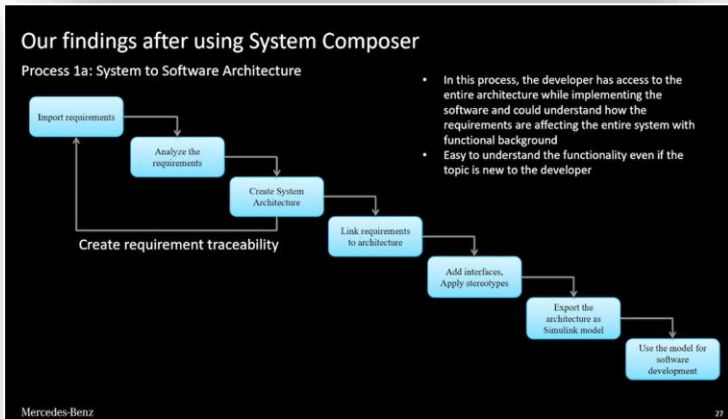
平台部署



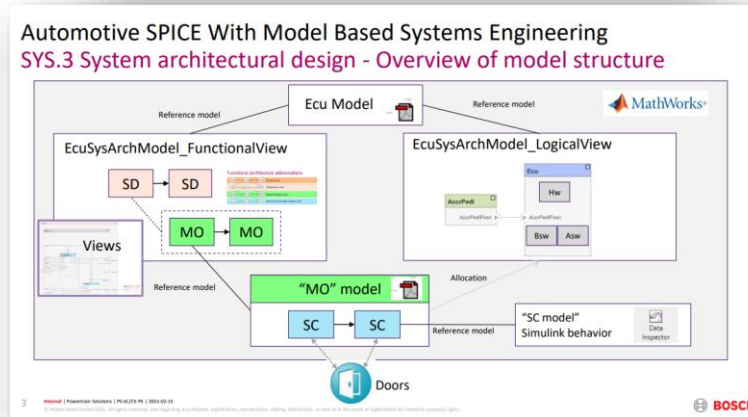
# 系统工程 – 统一数据源，简化 workflows



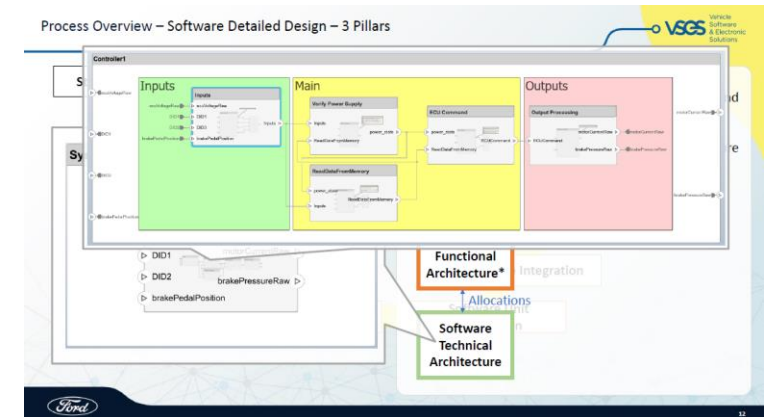
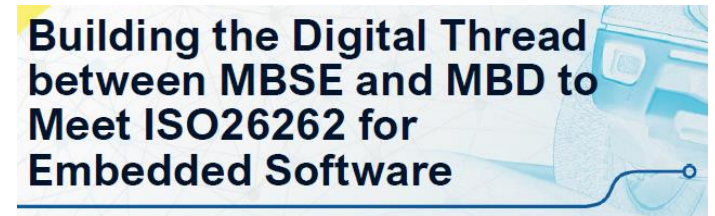
# 系统工程应用参考



奔驰 - 采用System Composer 创建系统架构



博世 - 按照ASPIICE要求开展系统架构设计



福特 - 建立满足ISO26262的 MBSE与MBD之间的数字线索

# 软件工厂 – 持续快速交付高质量软件产品

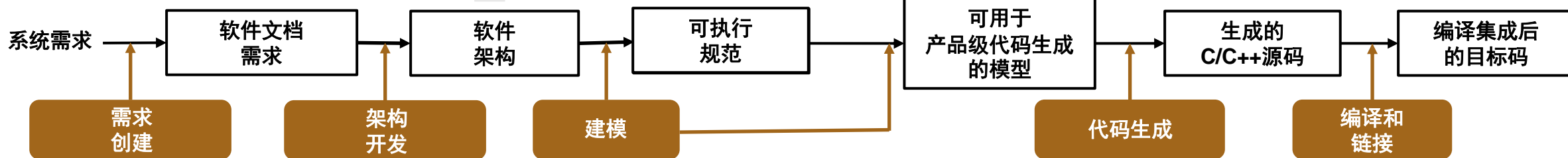
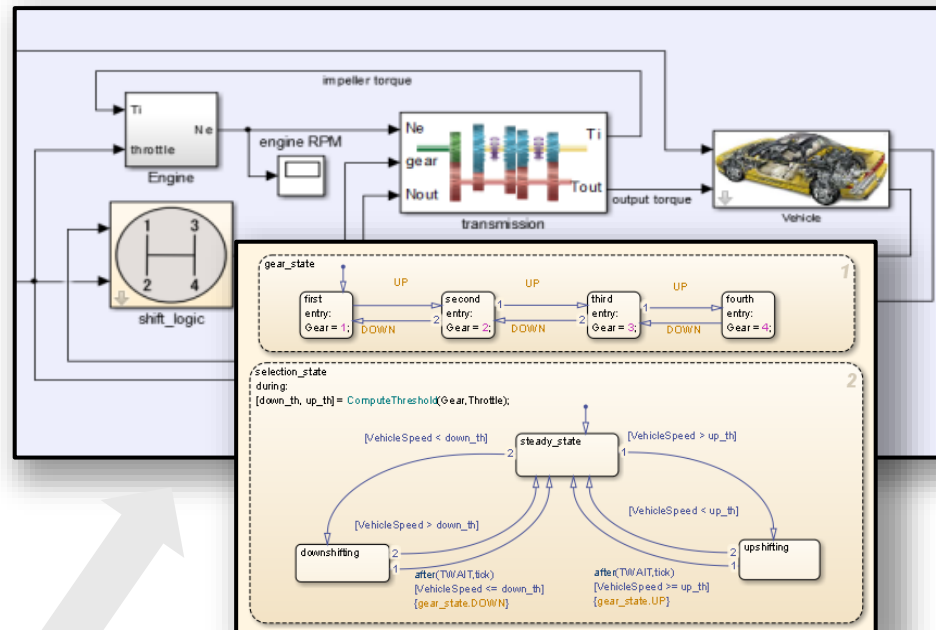
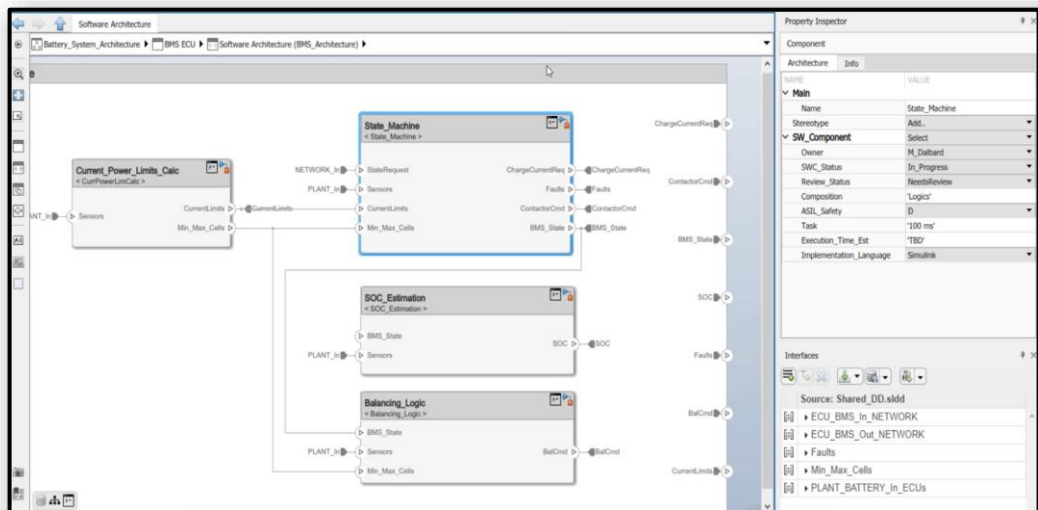
- 软件工厂应该持续交付高质量的软件产品
- 软件产品除软件本身还应包括文档、报告、工件等产出
- 高频任务（回归测试、度量统计、报告生成等）应该实现自动化
- 应该利用尚未自动化的流程探索阶段尽量查找并填补标准规范中的空白



问题：什么时候在软件工厂引入高频任务并自动化运行这些任务？

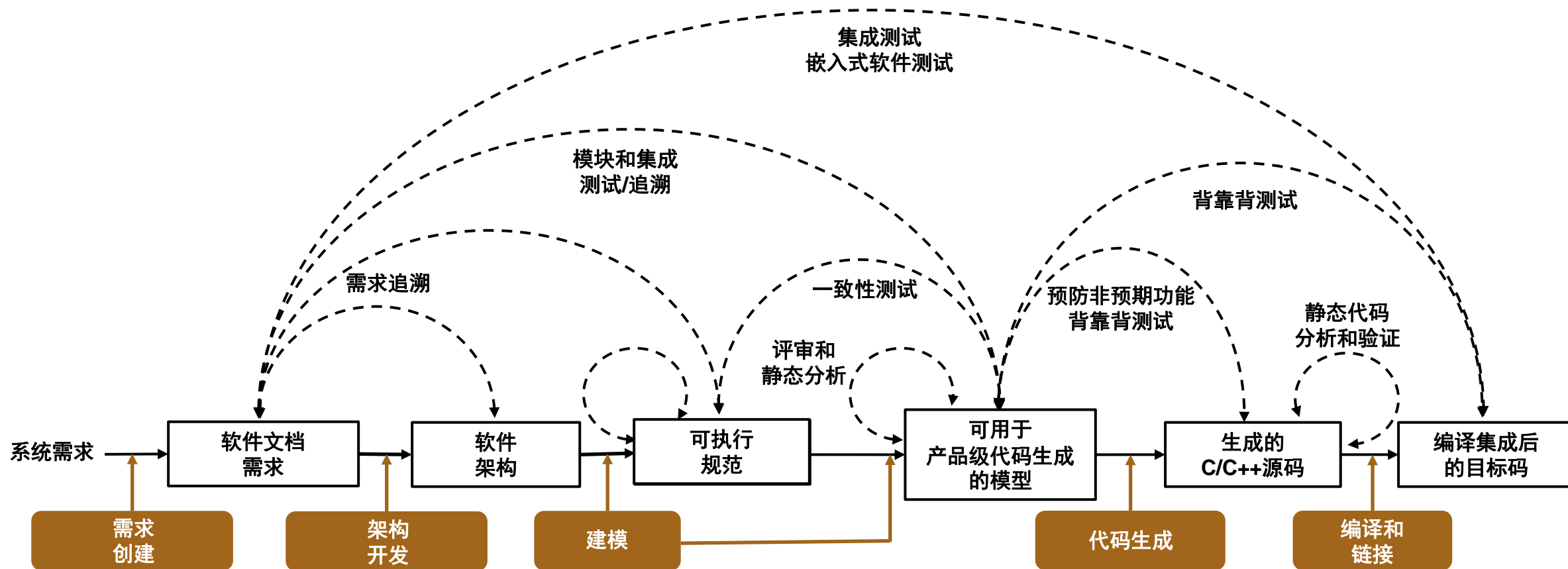


# 软件工厂 – 自动化的时机和方法考量

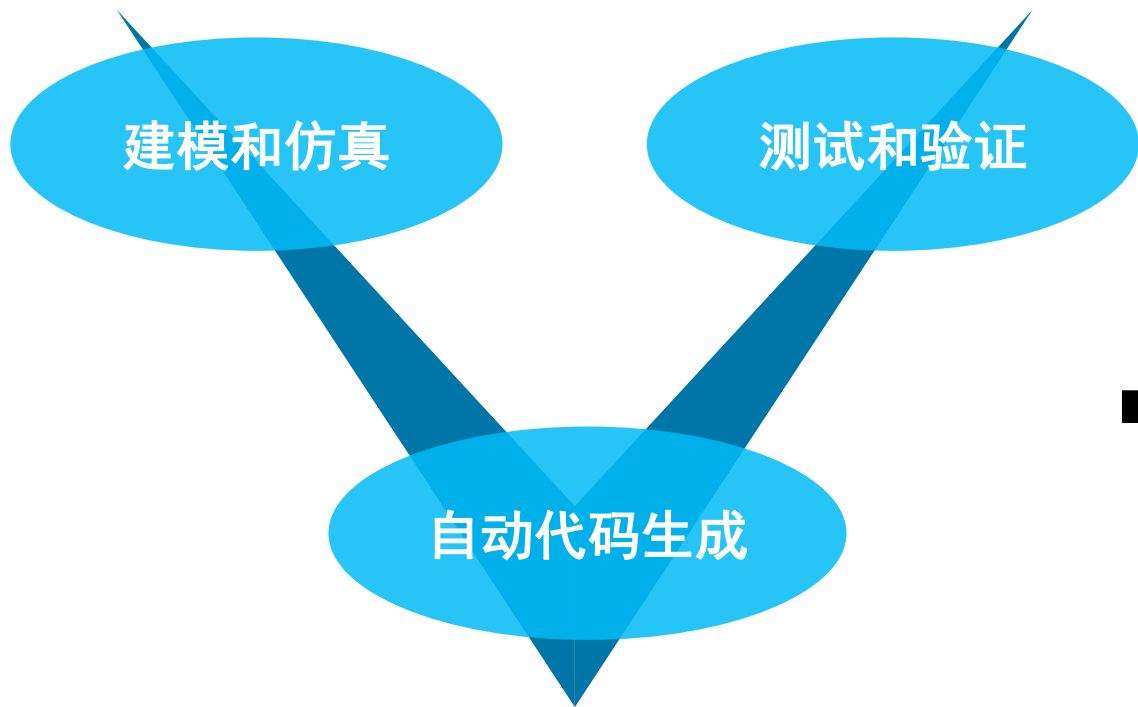


# 软件工厂 – 符合行业标准的流程选择

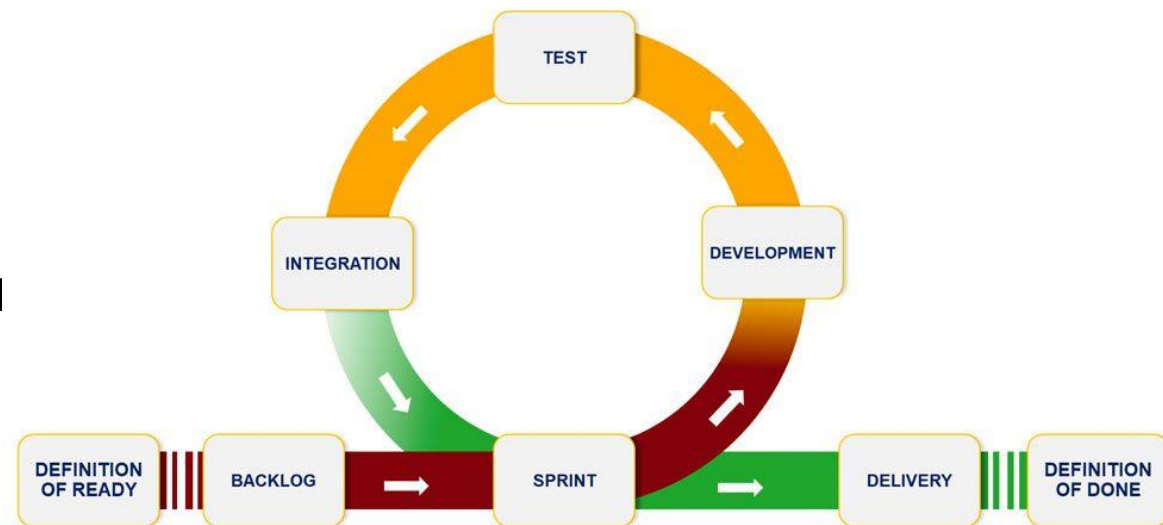
充分利用流程自动化手段 – 加速开发迭代，提升设计信心



# 软件工厂 – 融合敏捷开发理念和流程



+



复杂度提升



上市时间缩短



有限预算



软硬生命周期分离

# 软件工厂 – 搭建MBD持续集成平台



版本控制



- Git
- GitHub
- Subversion
- GitLab



流程触发



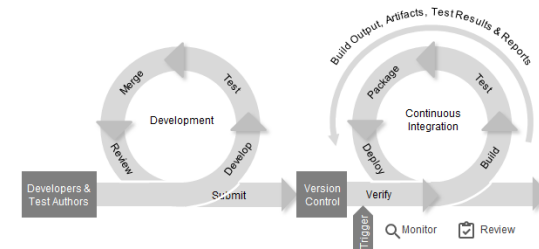
- 推送
- 合并请求
- 拉取请求
- 检入
- 周期
- 手动



构建生成



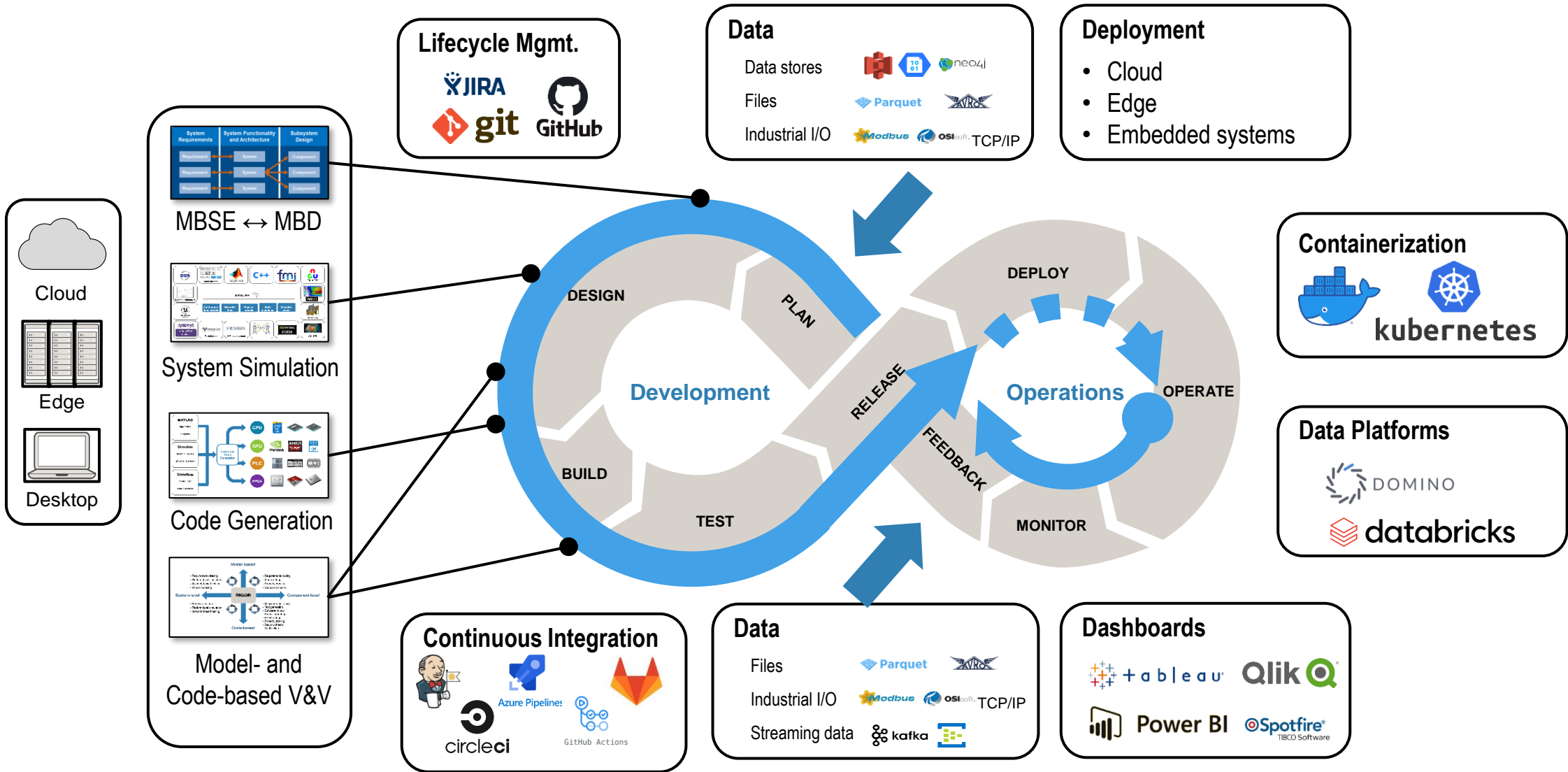
- 运行 MATLAB / Simulink 测试
- 运行性能测试
- 编译MEX文件
- 生成代码<sup>†</sup>
- 打包工具箱
- MATLAB Compiler编译<sup>†</sup>
- 集成其他软件技术



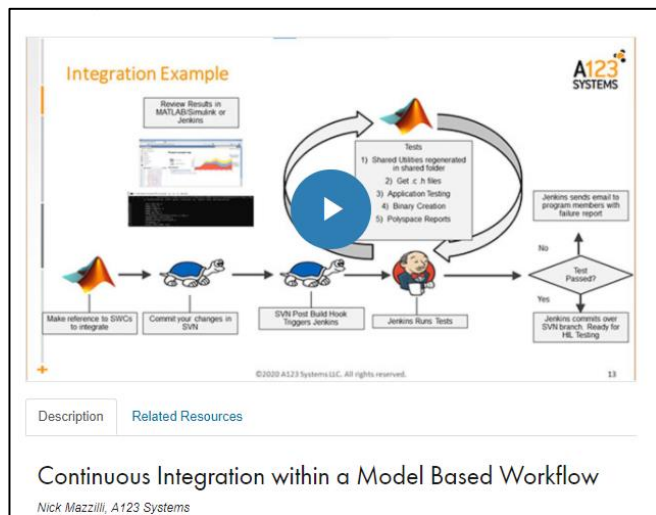
- 发布:
  - 测试结果
  - 覆盖结果
  - 性能结果
- 接受合并请求
- 邮件通知
- 构建其他项目

<sup>†</sup> 涉及产品形态转换时需要客户端访问许可证 (CAL)

# 软件工厂 – 从持续集成到开发运营

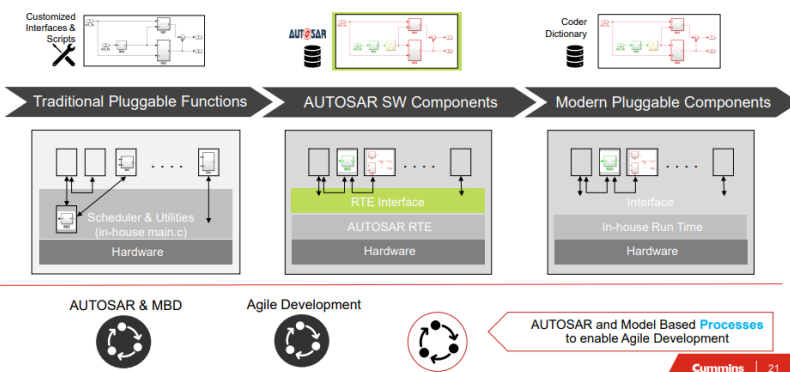


# 软件工厂应用参考



## A123 - 基于模型的持续集成工作流程

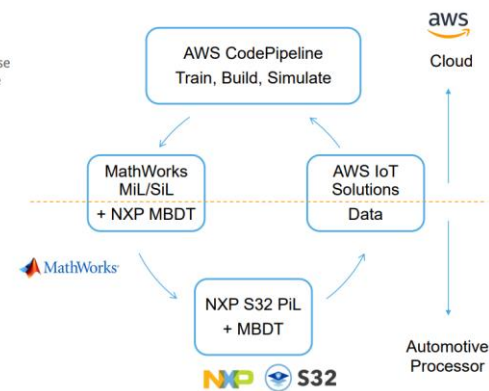
### Enable Agile Development with AUTOSAR and Model Based Processes



## 康明斯 - 集成AUTOSAR、ISO26262和基于模型设计实现更快地产品开发和系统仿真

### AUTOMOTIVE DEVOPS MODEL-BASED DESIGN

- Design, build and simulate in the cloud. Engineers use model-based systems engineering (MBSE) to manage system complexity, improve communication, and produce optimized systems
- Deploy to the Automotive Edge. NXP's S32G Vehicle Network Processors interface with all the vehicle functional domains and provide secure processing (AI/ML) and network acceleration for vehicle edge services.
- Integration with AWS CodePipeline and AWS IoT Greengrass enables a DevOps workflow built on AWS.



## NXP - 基于模型设计的车辆开发运营平台



# 数据驱动 – 集成人工智能算法赋能系统

- AI算法的集成是当今企业的首要任务
- AI算法应用的主要障碍:  
1) 与已有技术的集成 2) 数据的复杂度和质量 3) 专业技能
- AI应用成功 = 能交付给市场基于AI技术的产品和服务



“ We struggle to have correlation with the reality.  
**Linking the real system and AI** is the key challenge.”

– Innovation Manager - Automotive OEM



问题：如何将真实系统和AI连接起来？如何让数据科学家和工程师有效协同工作？

# 数据驱动 – 端到端的工作流

## 1 数据预处理

 数据清洗和预处理

 人工洞察

 仿真生成数据

## 2 AI建模

 模型设计和调试

 硬件加速训练

 互联互通

## 3 仿真测试

 复杂系统集成

 系统仿真

 系统验证和确认

## 4 算法部署

 嵌入式设备

 企业系统

 边缘、云端、桌面

# 数据驱动 – 预处理

1 数据预处理

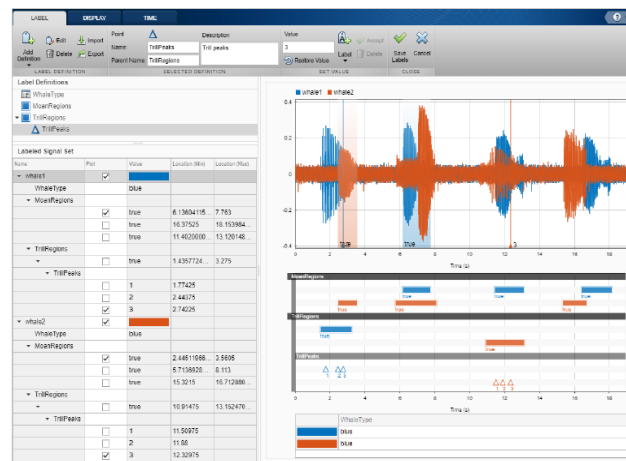
2 AI 建模

3 仿真测试

4 算法部署



简化处理数据的过程

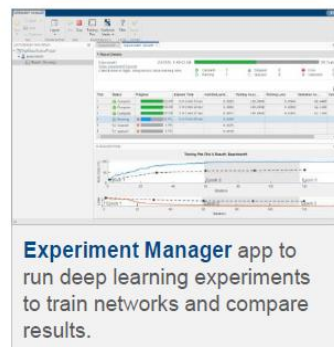
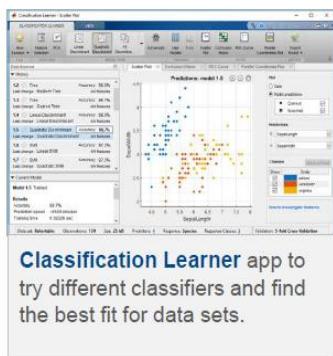
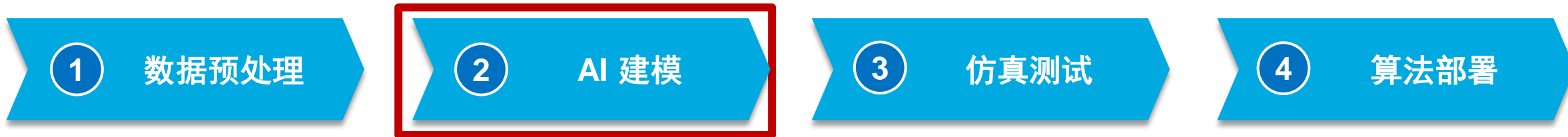


采用自动标注提高效率



基于仿真生成数据

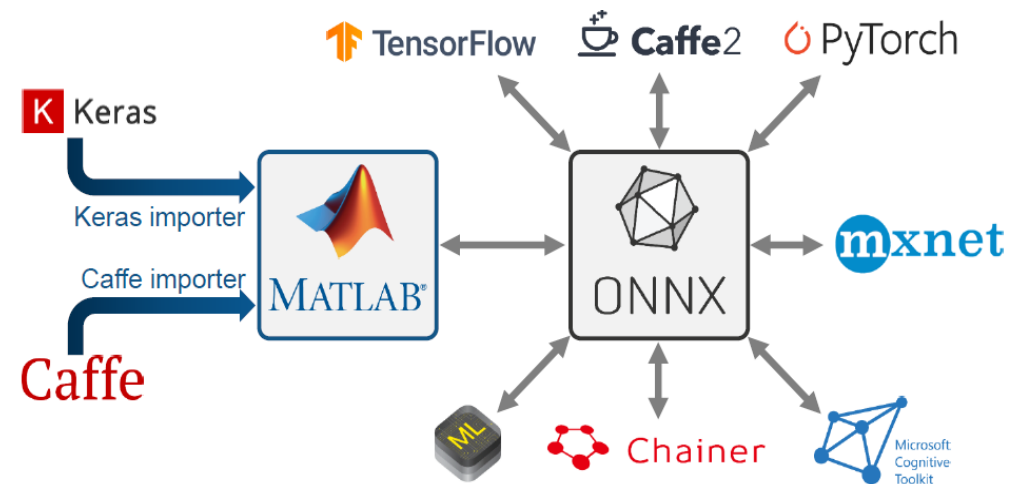
# 数据驱动 – AI建模



自动化的训练、调试和可视化



基于特定平台加速训练



利用标准实现AI模型的互联互通

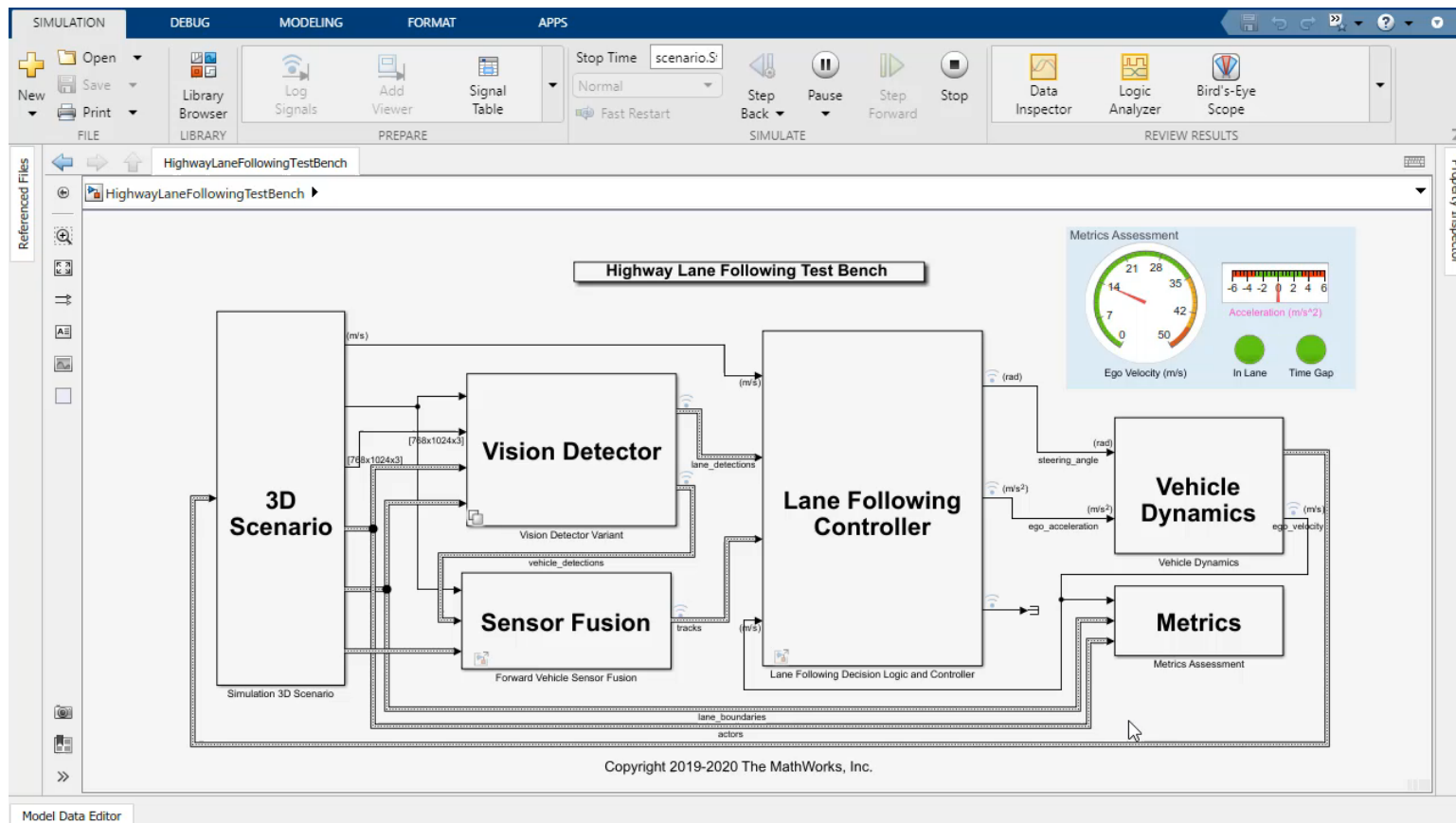
# 数据驱动 – 仿真测试

1 数据预处理

2 AI 建模

3 仿真测试

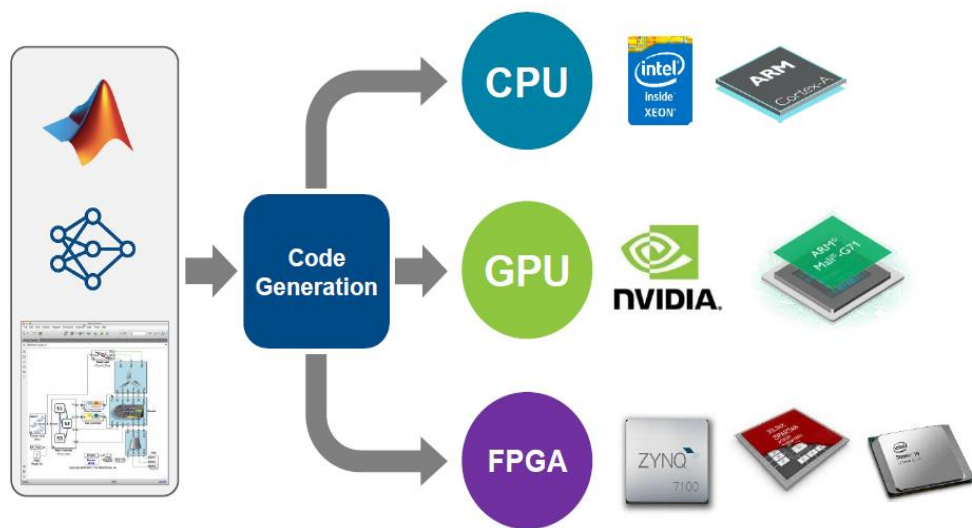
4 算法部署



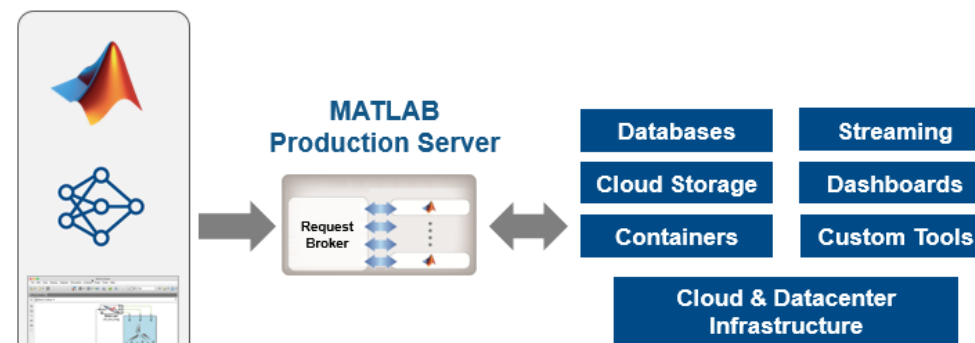
在Simulink平台结合传感器融合、控制逻辑、车辆动力学来执行仿真

集成AI算法模型到其运行的仿真场景

# 数据驱动 – 算法部署



部署到嵌入式设备



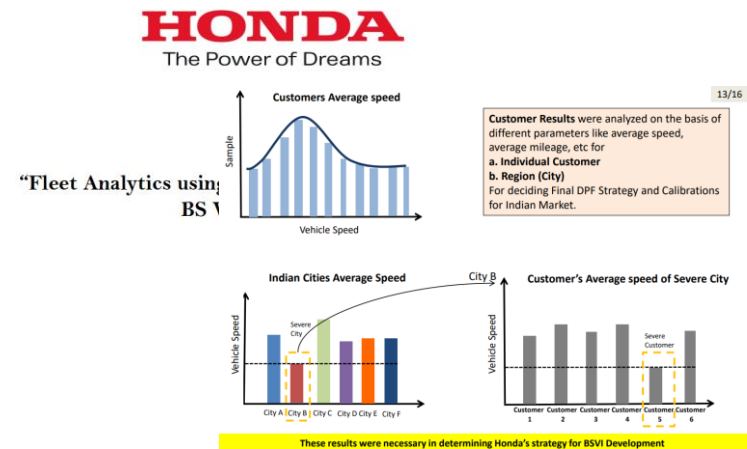
部署到服务器和云端



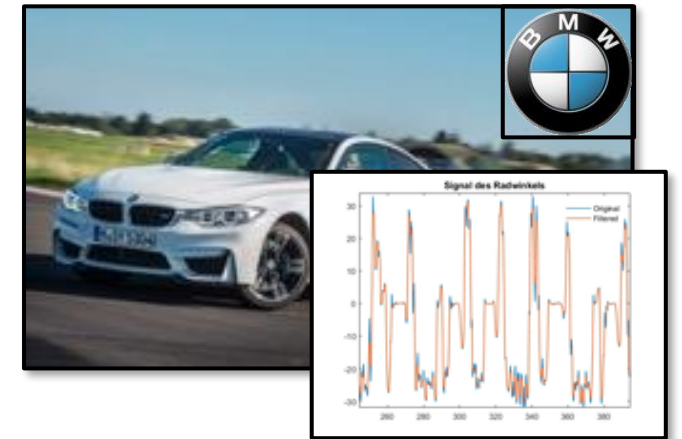
# 数据驱动应用参考



卡特彼勒 – 搭建人工智能和大数据基础架构



本田 – 使用驾驶数据分析开发排放控制策略



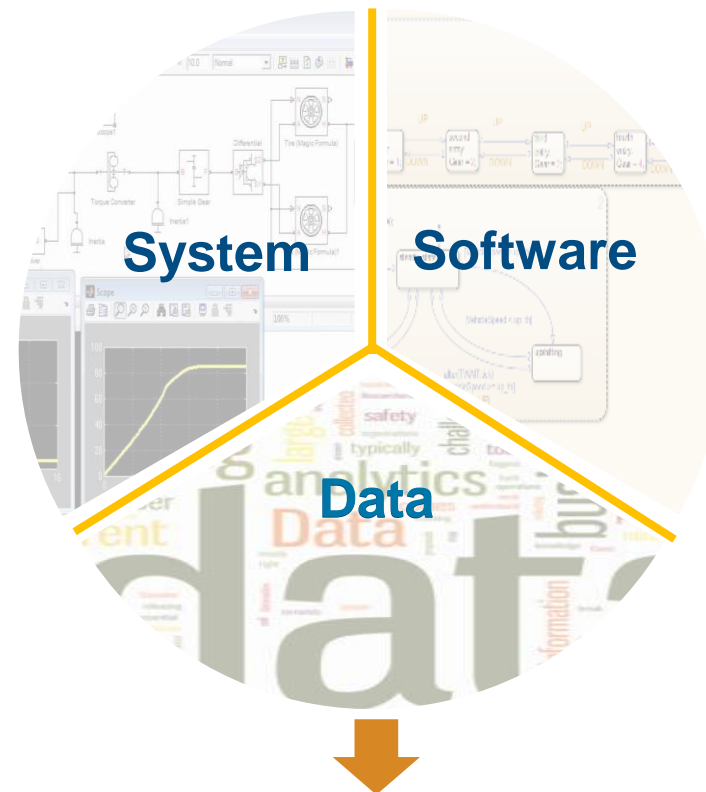
BMW - 利用机器学习检测汽车的过度转向

# 虚拟车辆 – 建模仿真车辆完整功能行为

虚拟车辆 – 让功能设计者可以在几分钟内以适当的详细程度集成**系统、软件和数据**创建出一辆虚拟车辆，用于对功能进行仿真，从而开展原型设计、虚拟标定和虚拟验证。

## 虚拟车辆的核心 – 仿真集成平台

- 丰富的即用功能实现快速创建
- 保持接口开放性进行定制扩展
- 仿真实现最大程度的流程前置



# 虚拟车辆仿真

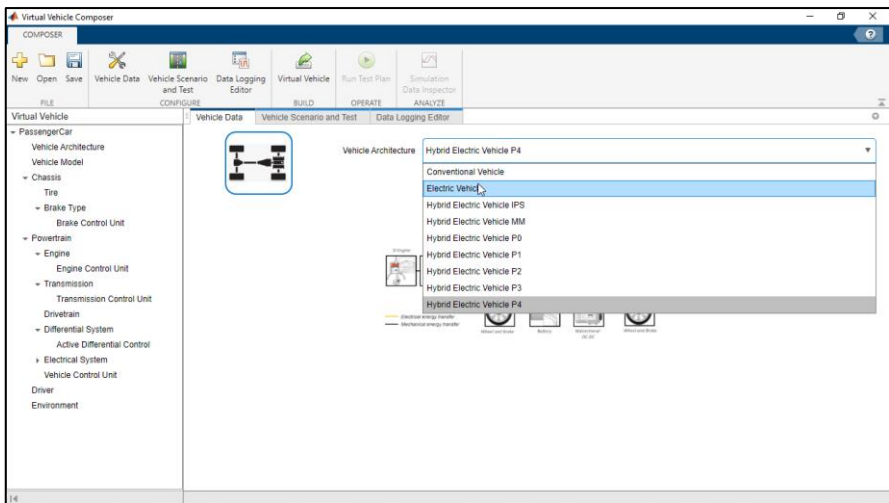
创建车辆

集成软件

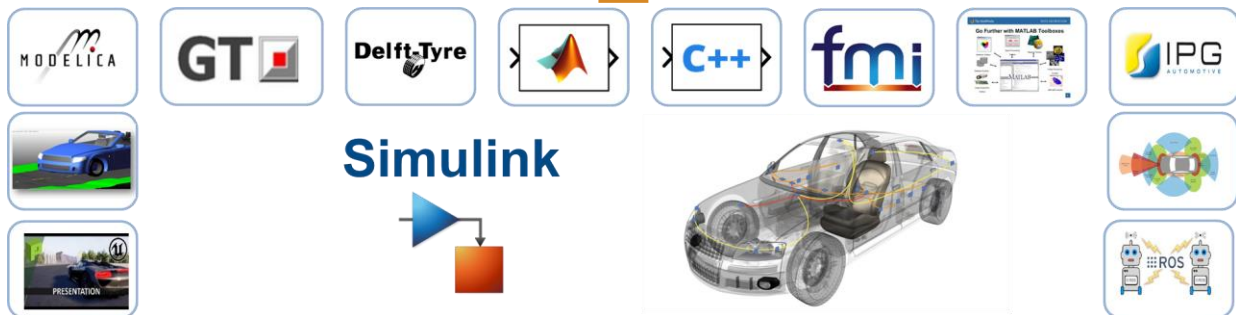
创建场景

仿真分析

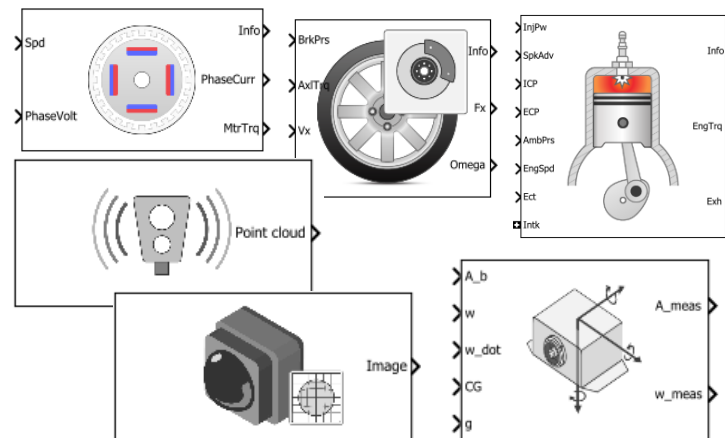
仿真部署



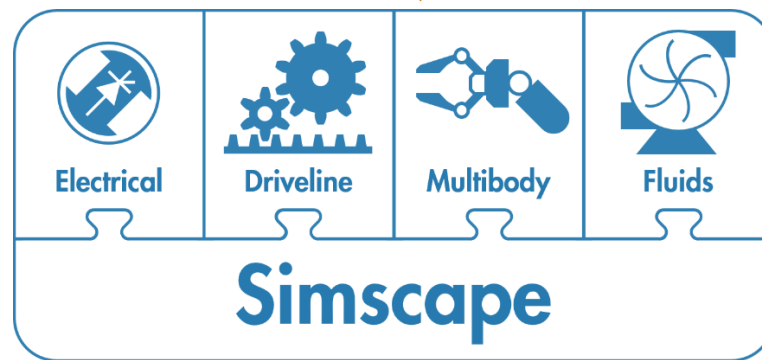
车型模板



外部模型集成



专业部件库



多学科模型库



# 虚拟车辆仿真

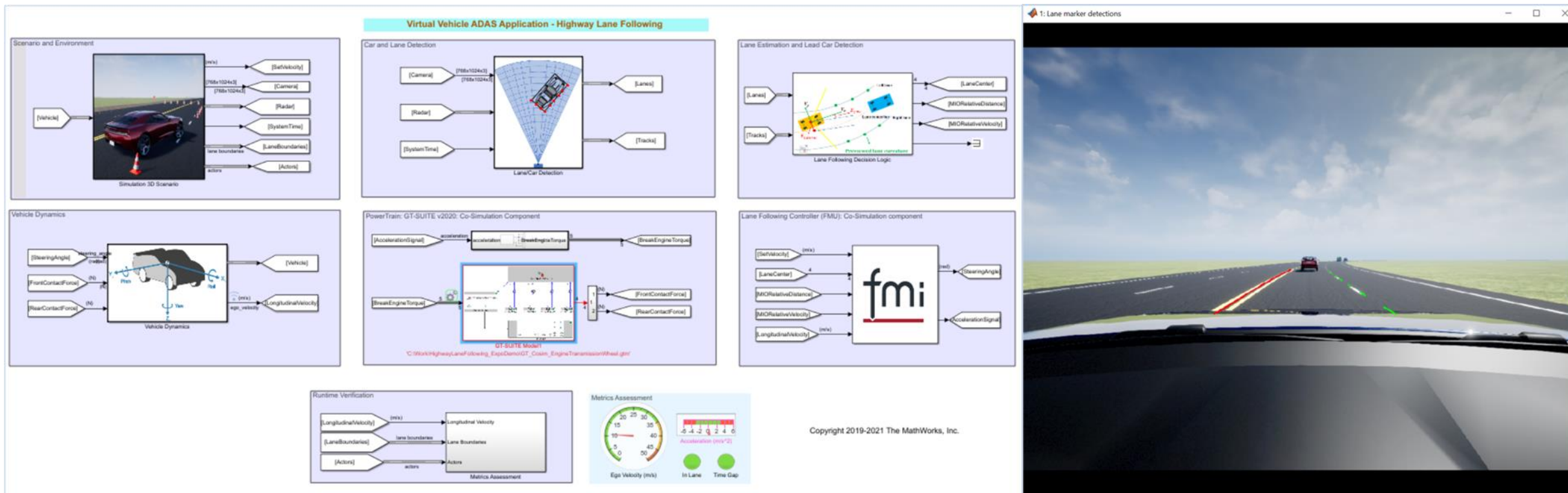
创建车辆

集成软件

创建场景

仿真分析

仿真部署



算法集成 – Simulink模块、C/C++源码/库、第三方软件（FMU、S函数）、.....



# 虚拟车辆仿真

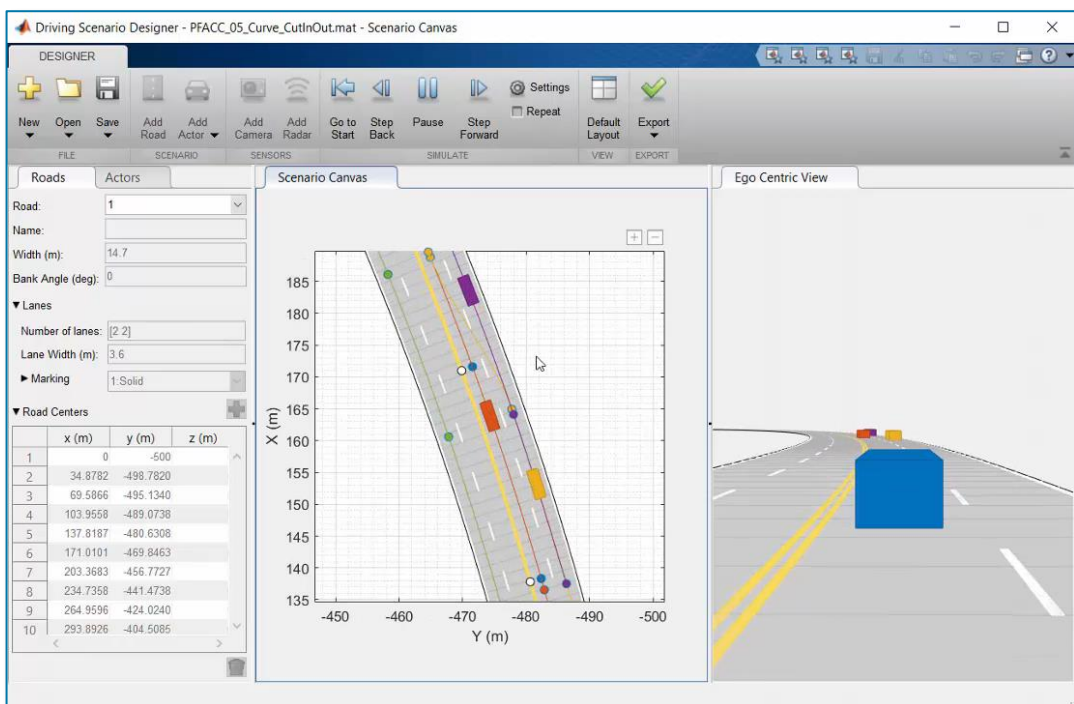
创建车辆

集成软件

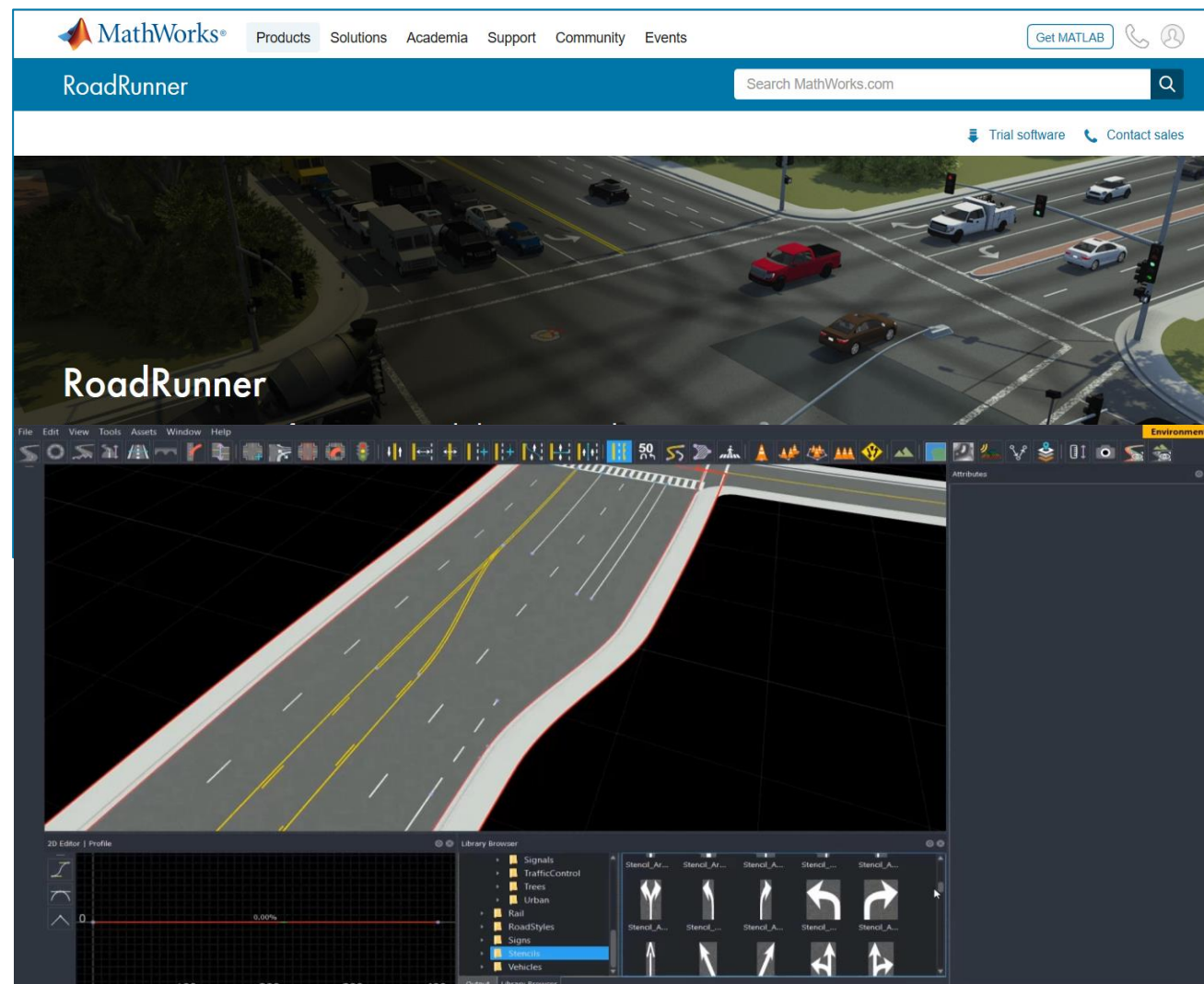
创建场景

仿真分析

仿真部署



简易场景



逼真场景

# 虚拟车辆仿真

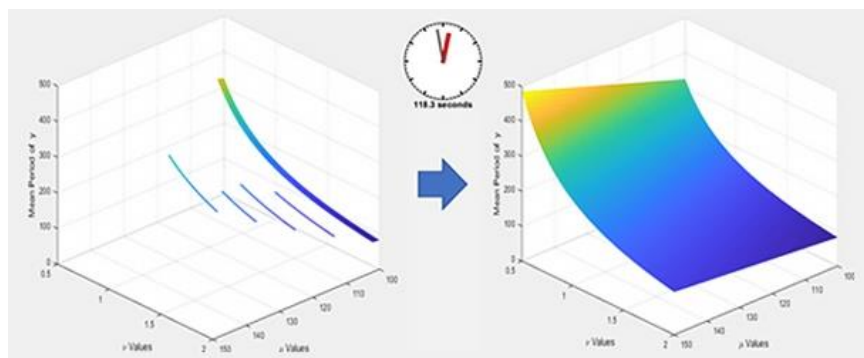
创建车辆

集成软件

创建场景

仿真分析

仿真部署



并行加速



```
164
165 finder = ChartDiagramFinder(hModel);
166 charts = find(finder);
167 ch = Chapter("Title", "Charts");
168 for chart = charts
169     section = Section("Title", chart.Name);
170     diag = getReporter(chart);
171     diag.SnapshotFormat = getSnapshotFormat(rpt);
172     add(section, diag);
173
174 % Report the objects in this chart
175 objFinder = StateflowDiagramElementFinder(chart);
176 sfObjects = find(objFinder);
177 for sfObj = sfObjects
178     objSection = Section("Title", sfObj.Name);
179     add(objSection, sfObj);
180 end
181 add(ch, section);
182 end
183 add(rpt, ch);
184
185
```

Chapter 2. Charts

2.1.1. shift\_logic

Figure 2.1.1. shift\_logic

Table 2.1. sf\_car/shift\_logic/gear\_state Properties

Property	Value
Type	AND State
Label	gear_state
Events	DOWN UP

2.1.2. selection\_state

Table 2.2. sf\_car/shift\_logic/selection\_state Properties

Property	Value
Type	AND State
Label	selection_state
during	[down_th_up_th] = calc_th(gear.throttle);

仿真可视化和后处理

# 虚拟车辆仿真

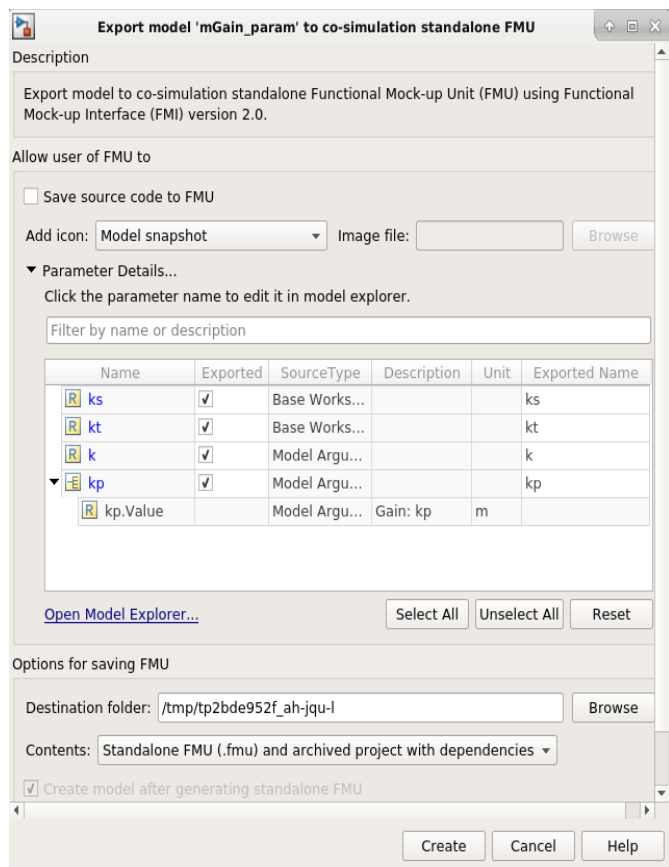
创建车辆

集成软件

创建场景

仿真分析

仿真部署



操作系统



实时仿真机

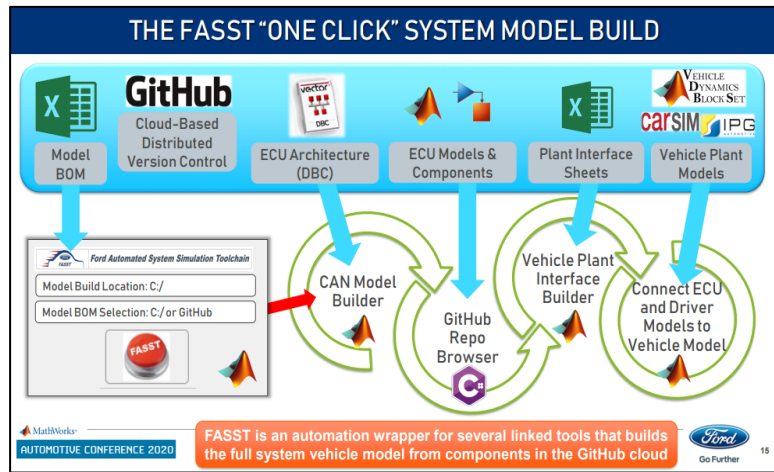


功能模型单元 (FMU)

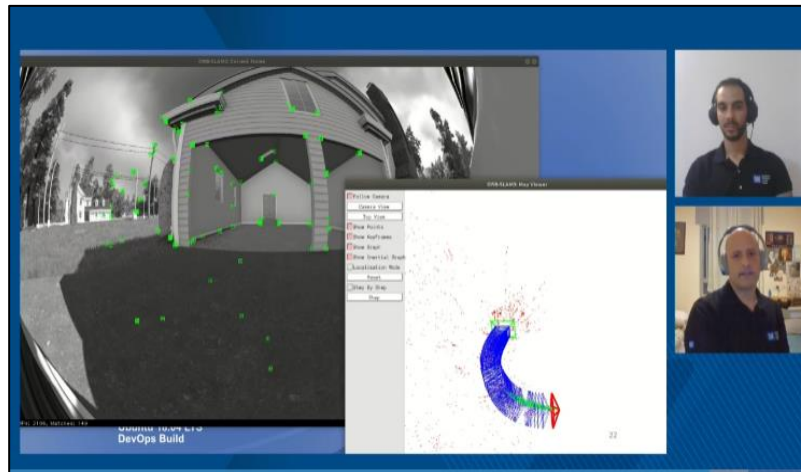
网页端APP应用



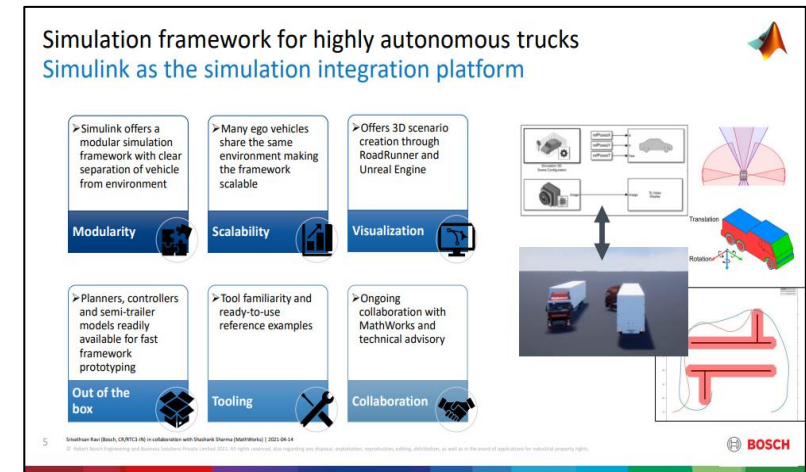
# 虚拟车辆仿真应用参考



福特 - 基于模型的敏捷化自动系统集成仿真平台



通用 - 基于仿真开发验证自动泊车系统



博世 - 高度自主物流卡车仿真框架

# 成为行业可靠的合作伙伴 – 对软件开发体系的建议

## 集成系统工程与软件开发

对系统级需求进行建模并关联到测试用例和可执行规范。具有组件边界和接口的架构为早期集成测试提供了基础。

## 融合敏捷开发到汽车工程

敏捷在被采用到汽车级车辆软件开发之前需要进行调整。

## 提前考虑云解决方案

软件构建/测试和仿真正在向云端转移。随着未来几年新项目的启动，这一趋势可能会加速。工具链选择应该提前考虑云解决方案，以使团队能够更好地协作。

## 标准化软件工厂中的工具链

标准工具链使工程团队能够专注于价值创造，并帮助高效应对标准的合规性。

## 选择流程中的自动化步骤

高频任务应该优先实现自动化，但更重要的是要找到并填补标准规范中的空白，并且不要过早地自动化以避免返工。

## 培养内部AI专家和开发能力

评估团队的现有成熟度，并确定领先和滞后指标。建立卓越中心以改善滞后指标。

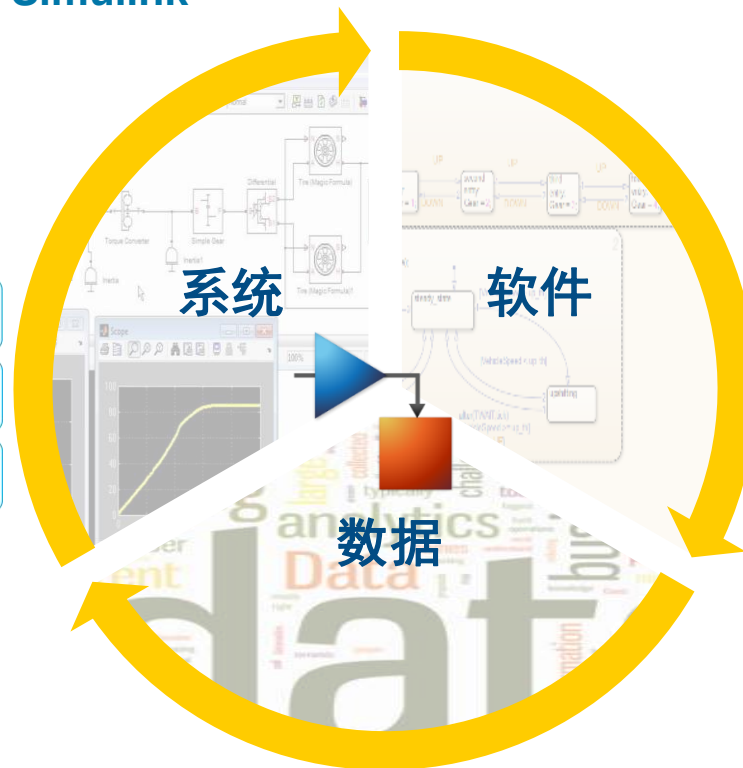
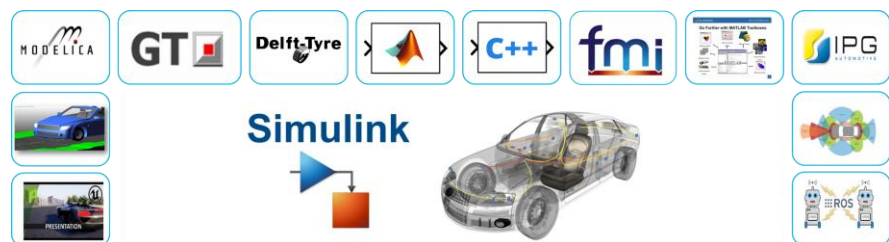
## 集成模型驱动和数据驱动流程

除了定制解决方案之外，数据驱动开发还需要与系统的其余部分和软件开发集成，以便在ADAS以外的电气化整车环境中测试数据算法。

# 成为行业可靠的合作伙伴 – 产品与平台

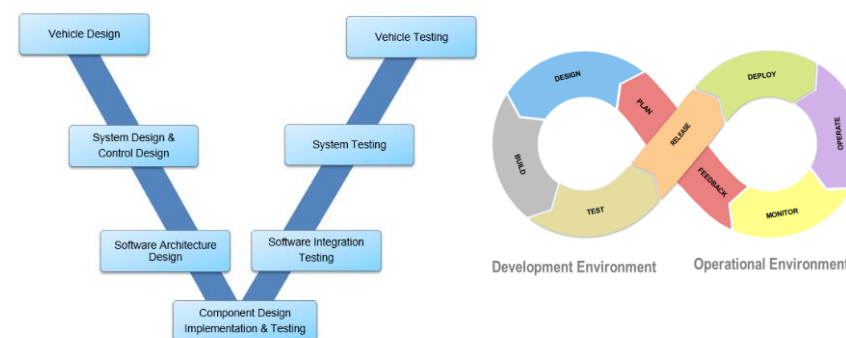
## 领先的仿真集成平台 – Simulink

- 架构权衡
- 部件选型
- 设计优化
- 原型仿真



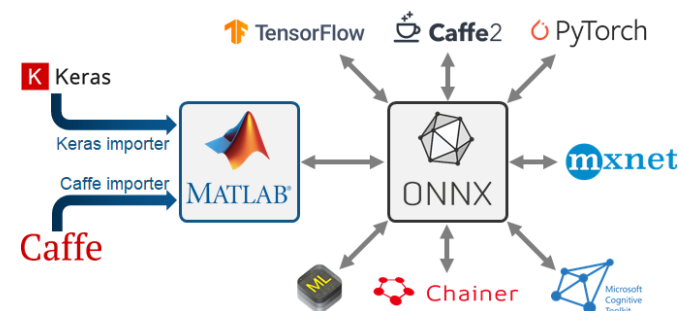
## 汽车行业软件开发标准方法 – 基于模型设计

- 可重用的组件
- 优化生成代码
- 集成测试框架
- 验证自动化



## 面向工程的AI 开发领导者 – MATLAB

- 从数据中挖掘要求和功能
- 提炼数据用于 AI 训练
- 根据测试结果验证仿真



# 成为行业可靠的合作伙伴 – 服务与支持



## 技术支持

- 产品使用
- 通用支持
- 508-647-7000



## 应用支持

- 流程、方法和工具
- 专题、实训和讲座
- 评估支持



## 项目支持

- 流程应用
- 工具导入
- 概念验证



## 培训服务

- 工具和/或流程的付费培训
- 现场或线上由讲师带领
- 在线自定进度培训



## 咨询服务

- 面向工程服务的付费咨询
- 流程评估、工具定制等
- 根据客户需求定义目标



## 总结

- 汽车行业清洁安全和数字移动的期望引导着技术创新和投资方向
- 需要重新审视和塑造现有的软件开发体系，以应对软件定义汽车趋势
  - 系统工程
  - 软件工厂
  - 数据驱动
  - 虚拟车辆
- MathWorks继续成为行业可靠的合作伙伴，共同面对挑战 and 机遇

感谢参会！



© 2022 The MathWorks, Inc. MATLAB and Simulink are registered trademarks of The MathWorks, Inc. See [mathworks.com/trademarks](https://www.mathworks.com/trademarks) for a list of additional trademarks. Other product or brand names may be trademarks or registered trademarks of their respective holders.