



广汽研究院
GAC R&D CENTER

高质量的MBD汽车软件端到端数字化开发平台

李欢, 广汽研究院



2024 MathWorks
中国汽车年会

关于话题

高质量的MBD汽车软件端到端**数字化**开发平台

端到端
End to End

业务

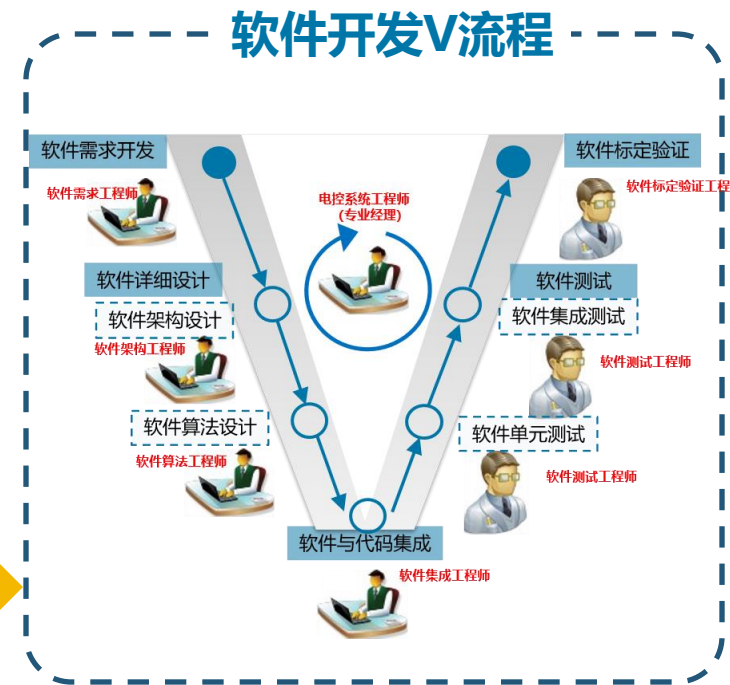
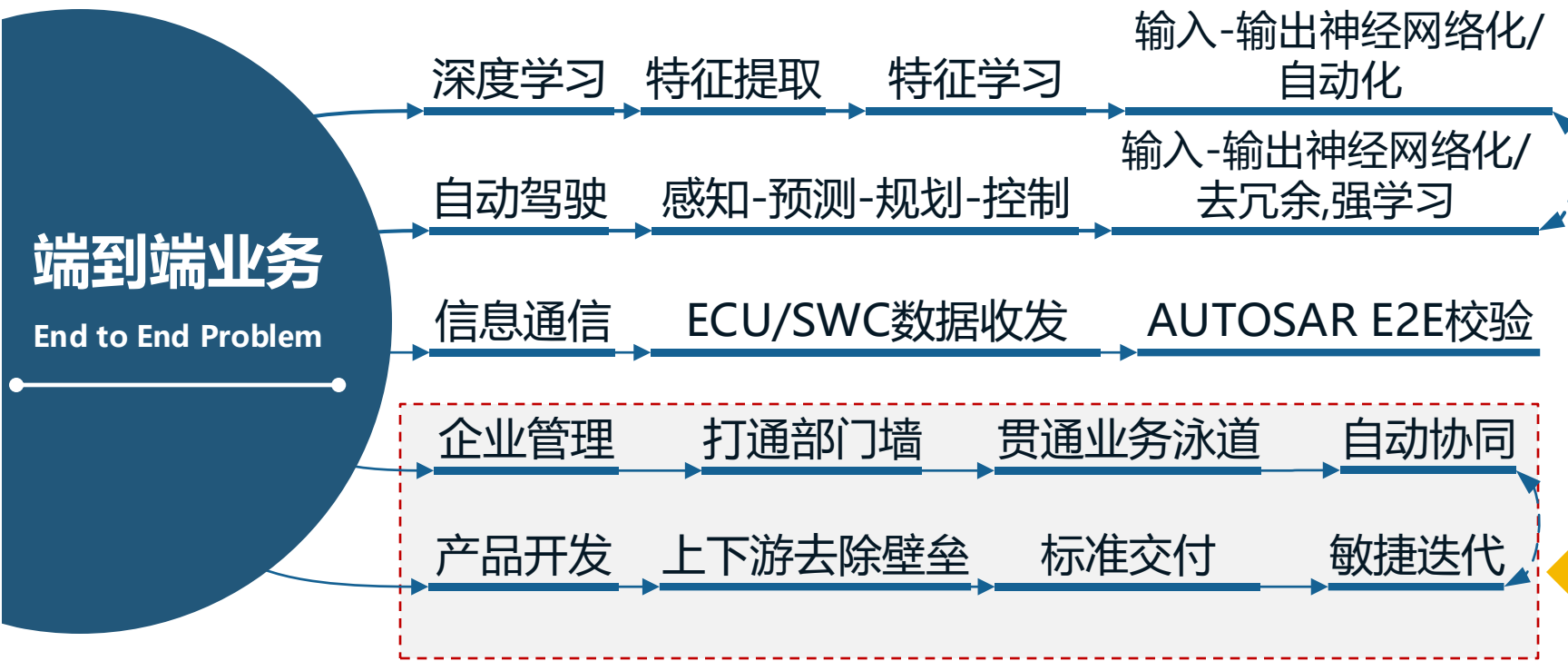


数字化
Digitalization

方法



关于话题 | 什么是端到端



共同特点

- 输入-节点-输出
- 信息加工-传递
- 可重复/迭代

业务目标

- 标准**：优化端到端软件开发业务
- 可靠/自动**：更加标准、自动、敏捷
- 敏捷**

业务实施

- 整体**：构建**一站式**工作平台、打破工具壁垒、提高**自动化**、核心业务**虚拟化**——**端到端工具链优化**
- 局部**：节点上下游紧密协同， **workflow 标准化**、**过程交付标准化**——**端到端人员活动优化**

关于话题 | 端到端软件开发业务如何优化

数字化方法 | Digitalization Methodology

机械化 → 电气化 → 信息化 → 数字化

数字化加速软件定义汽车

升级

软件开发V流程



提出问题

□ 软件本身就是数字化产物，软件开发要如何数字化？

解决问题

✓ MBD端到端开发平台——寻求高效率、高质量新能源整车控制软件开发最佳实践

目录

Outline

01 | 数字化软件开发的内外部需求

02 | 开发**流程**数字化

03 | **业务**活动数字化

04 | 过程**交付**数字化

05 | 平台部署与总结

**MBD端到端
平台解决方案**

数字化软件开发的内外部需求 | 外部需求：汽车软件开发的挑战

数字化转型是对组织核心业务进行系统性重定义，运用IT等相关技术手段，对**组织活动、流程、业务模式和员工能力**的方方面面进行重定义或升级^[1]，是一项系统工程，本质是**业务转型**，目的是**提质增效**，增强企业和产品**竞争力**。



市场竞争加剧、软件迭代加快，**提质增效**（减时减车减费用）势在必行。

◆ 系统规模/法规要求提升

出行工具 智能移动终端

◆ 开发周期/成本压缩

OTA

产品迭代需求倒逼开发周期压缩
行业竞争态势倒逼开发成本压缩

◆ 多样化开发需求

BEV P-HEV REV (含氢动力)

四驱 分布式驱动 两驱(前/后驱)

多样化的驱动形式、能源形式对软件的平台化、模块化提出高要求

◆ 标定验证负荷增加/资源消耗大

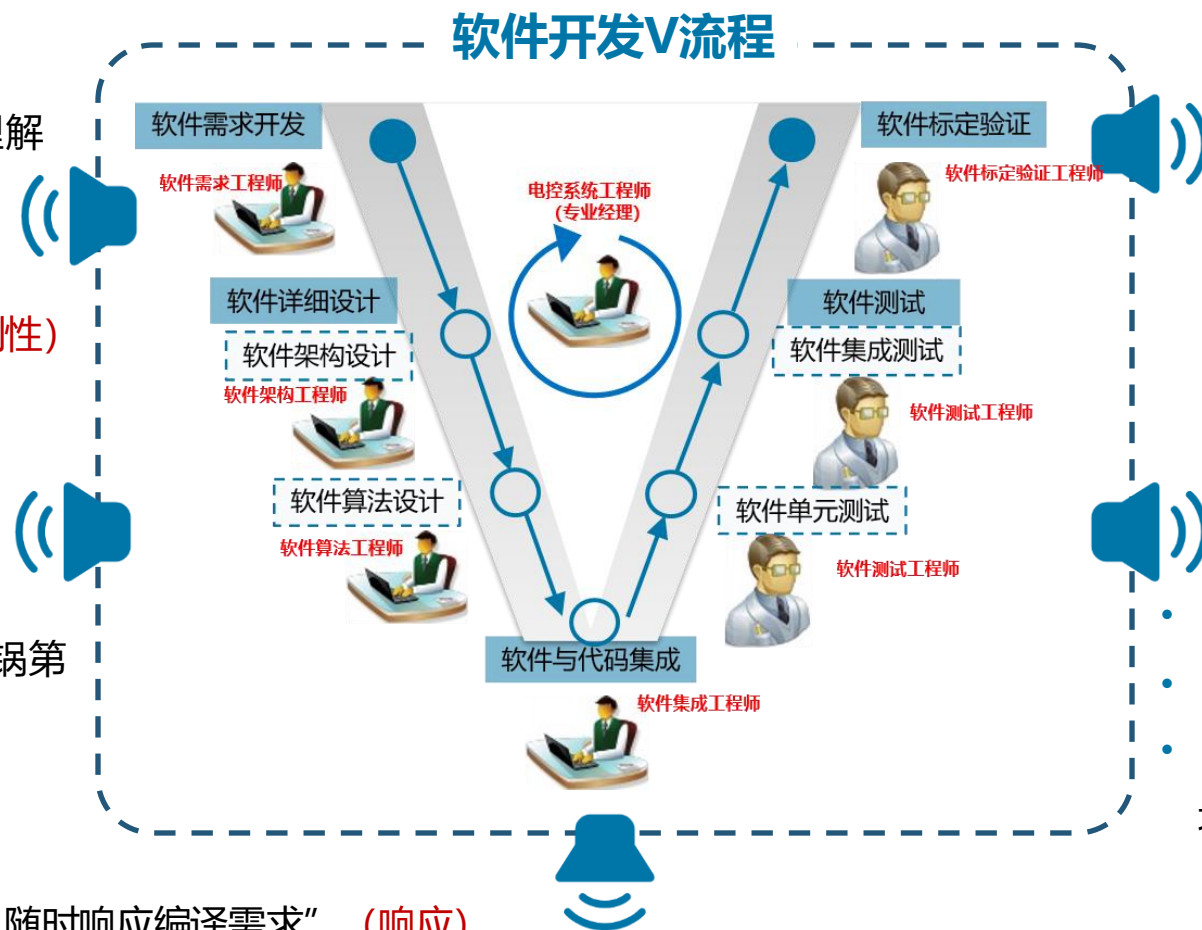
测试工作量成倍增长

数字化软件开发的内外部需求 | 内部需求：软件开发人员的“吐槽大会”

- “软件需求难写，千人千面，理解不一致” (质量)
- “颗粒度难把控” (质量)
- “缺反馈,难维护、吃灰” (便利性)

- “码农、腰间盘突出” (工作量)
- “bug制造者；干的好应该，背锅第一名” (质量)

- “24/7 standby, 随时响应编译需求” (响应)
- “编译耗时，在线等，时效性差” (效率)
- “枯燥重复、工具人、离车端远” (价值)

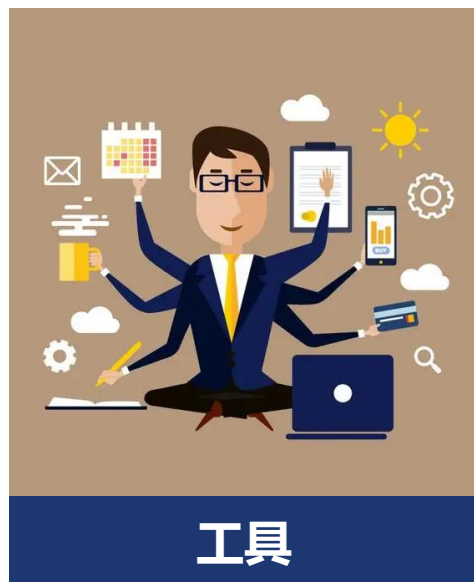


- “功能参数多，标不完，永远标不完” (工作量)
- “前端没搞对，白跑一趟” (效率)
- “样车又被谁借走了” (资源)
- “高速标定你去吧” (难度)
- “用例难编，需求不明确” (质量)
- “海量用例，100%覆盖费时费力” (工作量)
- “需求难冻结，架构和交互一直变，测试环境永远不固定” (效率、工作量)

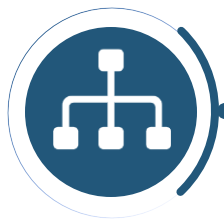
标准化、自动化、规范化、(样车) 虚拟化跟不上业务发展

数字化软件开发的内外部需求 | 需求分解：数字化如何排解软件开发的内忧外患

□ V流程的数字化解决方案：定制化的基于模型的设计(MBD)平台，打通 workflow、工具链、交付物，并支撑虚拟样车业务。



高效高质量控制软件开发要素



workflow

流程清晰、步骤明确



业务工具

工具拉通、高度自动



过程交付

标准输入、标准输出

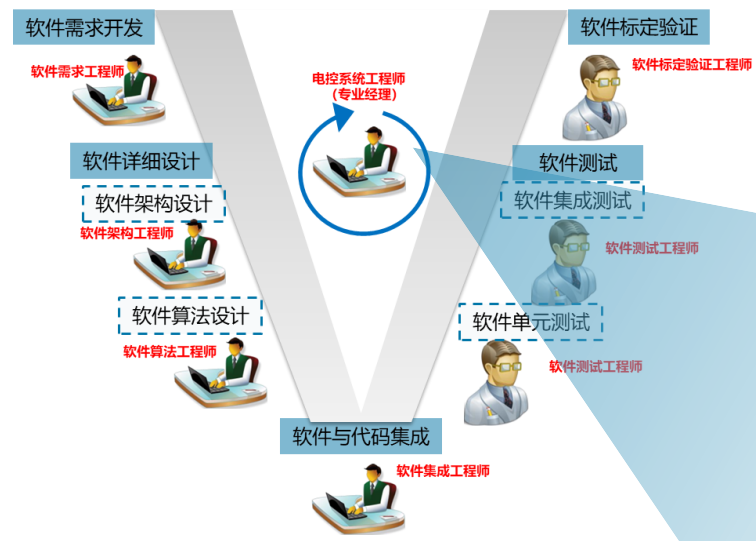


虚拟样车

标定验证、降本增效

数字化软件开发的内外部需求 | 需求分解：数字化如何排解软件开发的内忧外患

□ V流程的数字化解决方案：开发一套贯穿“软件需求”到“标定验证”全流程的高效率、高质量开发平台



有哪些任务、执行顺序、如何固化

- 设计追溯到架构和需求
- 模型静态分析
- 软件单元报告
- 代码生成、代码静态分析
- 编写测试用例，执行动态测试
- 参数标定
- 节点评审

如何自动化执行这些任务操作

- 开发环境
- 工具链集成
- 桌面操作
- 脚本自动化
- CI/CD



有哪些输入输出、如何标准传递自动留存

- 设计交付物：需求/模型/用例
- 双向追溯性
- 设计说明文档
- 评审/仿真/测试报告
- 源代码管理

目录

Outline

01 | 数字化软件开发的内外部需求

02 | 开发**流程**数字化

03 | 业务活动数字化

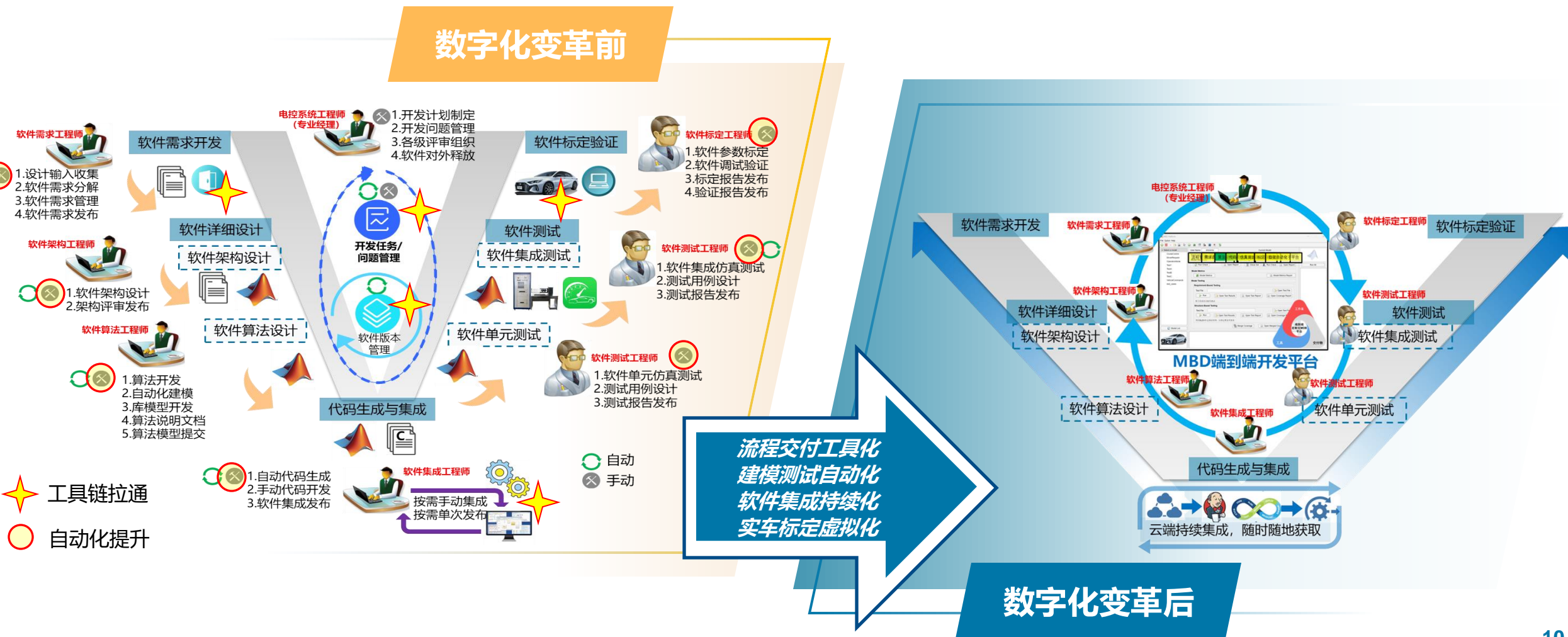
04 | 过程交付数字化

05 | 平台部署与总结

**MBD端到端
平台解决方案**

软件开发流程数字化 | workflow定制方案：端到端流程数字化提升

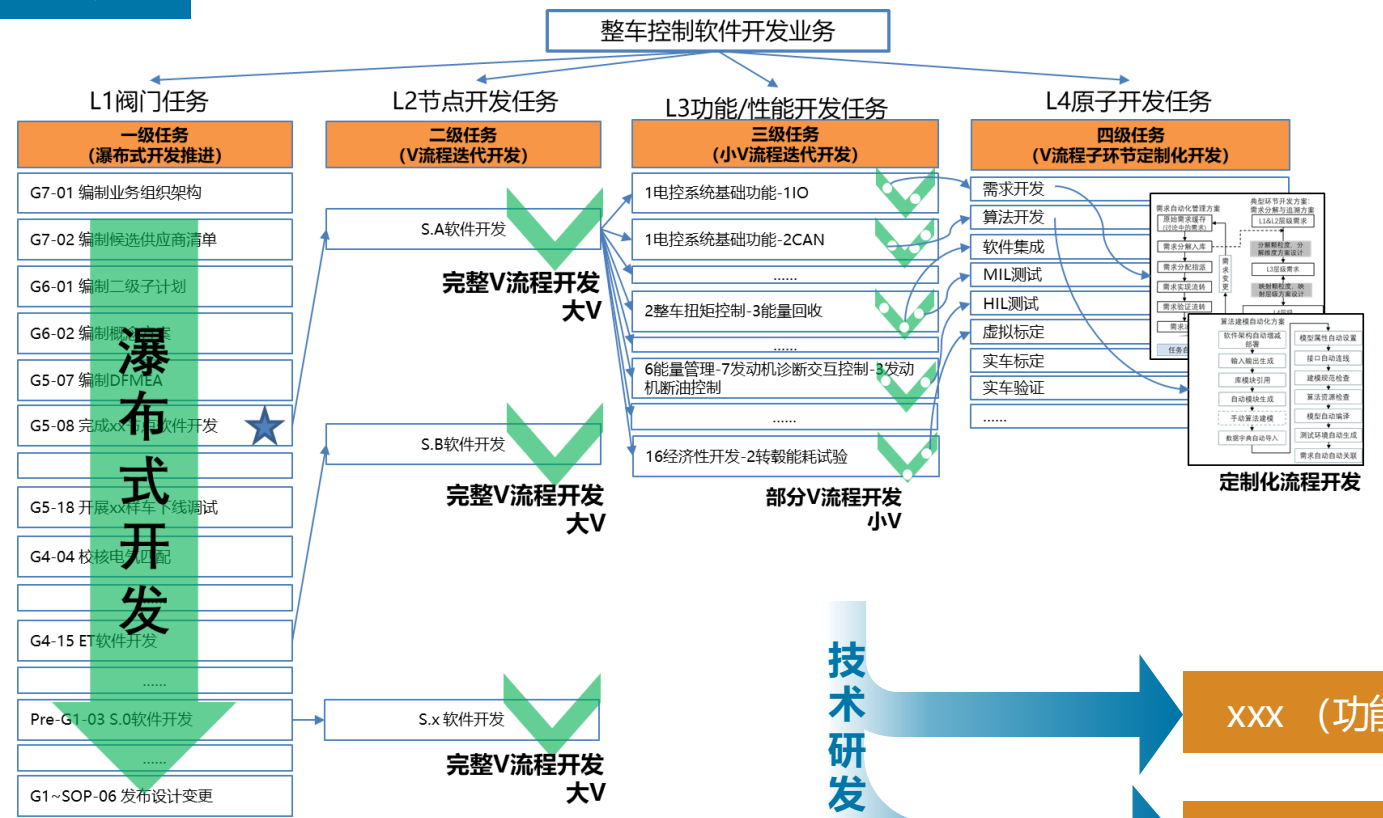
- ❑ 流程人为管控、业务手动操作多、工具链隔离、交付差异化的**分散开发模式**
- ❑ 流程自动管控、业务操作自动化、工具链联通、交付标准化的一站式**平台开发模式**



软件开发流程数字化 | 业务流程梳理：PLM流程管理 vs MBD流程执行

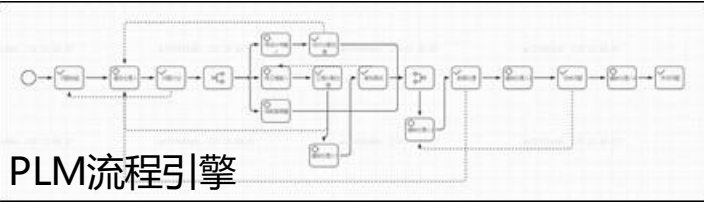
自定义轻量化PLM

业务流程



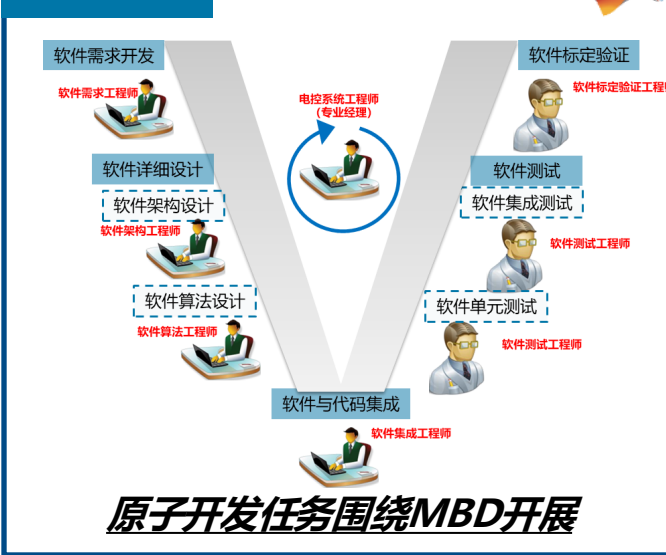
技术研发

- xxx (功能/性能) 技术开发
- xxx (功能/性能) 技术开发
- xxx (功能/性能) 技术开发



MBD平台

业务手段



原子开发任务围绕MBD开展

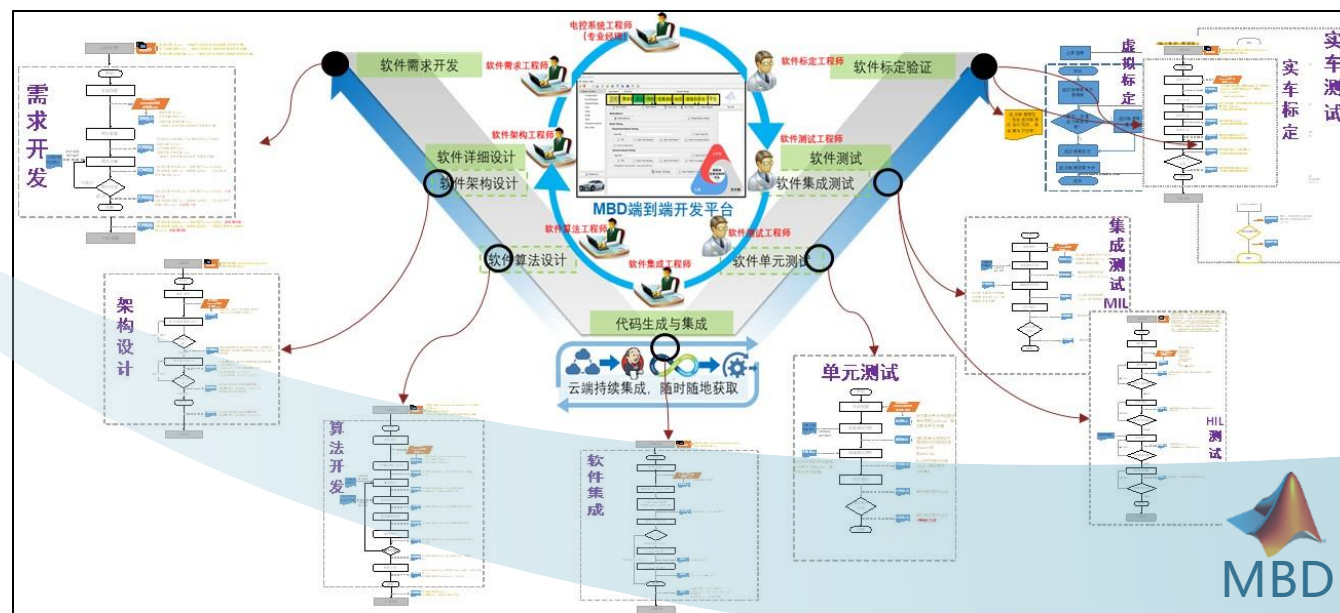
- 自动化流程 (PLM) 和自动化工具 (MBD) 助力技术研发快速迭代和落地

软件开发流程数字化 | 业务流程梳理：PLM流程管理 vs MBD流程执行

- 流程载体：PLM定制分发4级流程，自动流转管控；
- 流程执行：MBD平台按L2/L3/L4级对应流程步骤执行；
- 节点交付：各级节点交付物（文档/用例/报告...）MBD与PLM接口交互上传。

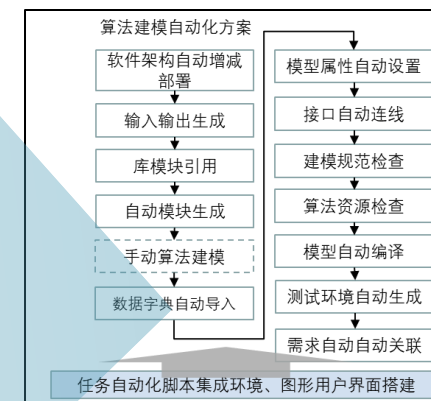


L1-L4四级开发任务基于PLM平台
流程分解与推进



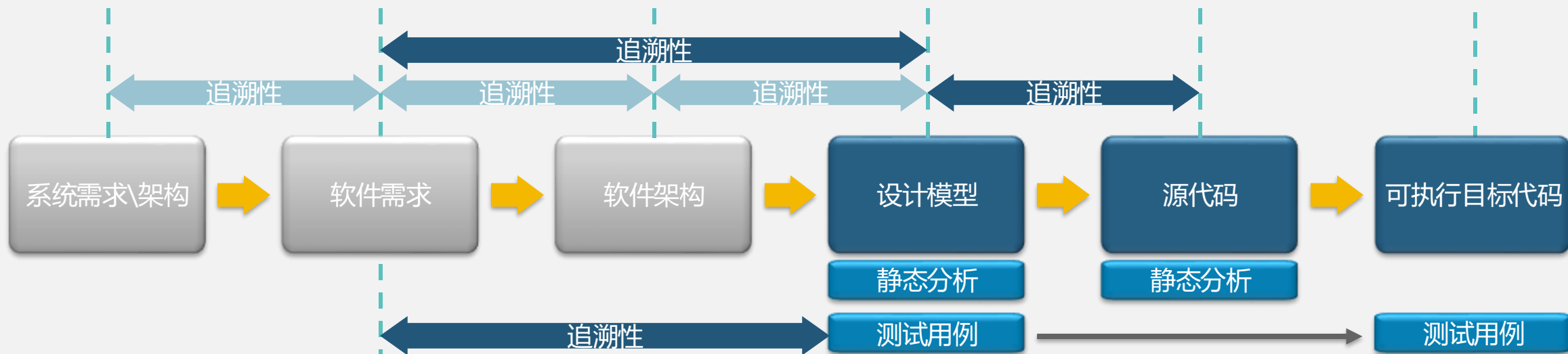
L4 原子开发任务基于MBD平台按流程执行

Eg: MBD详细操作流程
L4-算法建模流程模板



- ✓ 流程设计之初便定义好“流程任务 vs 工具 vs 交付”的关联；
- ✓ 流程发起、执行、交付过程形成PLM-MBD-PLM的任务闭环。

软件开发流程数字化 | 流程业务基座：MBD平台全流程开发能力及工具协同能力



交付物管理 (对接PLM/ALM)

- 结合第三方源代码管理工具管理设计交付物、检查清单和各类报告。
- 代码生成定制与追溯性报告定制
- 与第三方生命周期管理工具 (PLM/ALM)交互, 自动导入/导出



模型/代码分析 (对接SCM)

- 模型/代码规范检查
- 模型/代码metrics
- 模型/代码设计缺陷检测
- 模型/代码形式化验证



自动化测试&虚拟标定 (对接SCM)

- 使用 界面/Excel 编写测试用例
- 自动建立测试用例与需求双向追溯
- 为缺少的结构覆盖度生成测试用例
- 运行测试用例并生成测试报告
- 虚拟标定: 建模与参数寻优



持续集成 (对接ALM/SCM)

- 使用 界面/Excel 编写测试用例
- 自动建立测试用例与需求双向追溯
- 为缺少的结构覆盖度生成测试用例
- 运行测试用例并生成测试报告
- 运行 MIL/SIL



质量评审 (对接PLM)

- 可定制的指标图表和质量目标与设计检查项之间的映射关系。
- 基于浏览器查看的静态分析 / 动态测试 / 项目管理Dashboard

目录

Outline

01 | 数字化软件开发的内外部需求

02 | 开发流程数字化

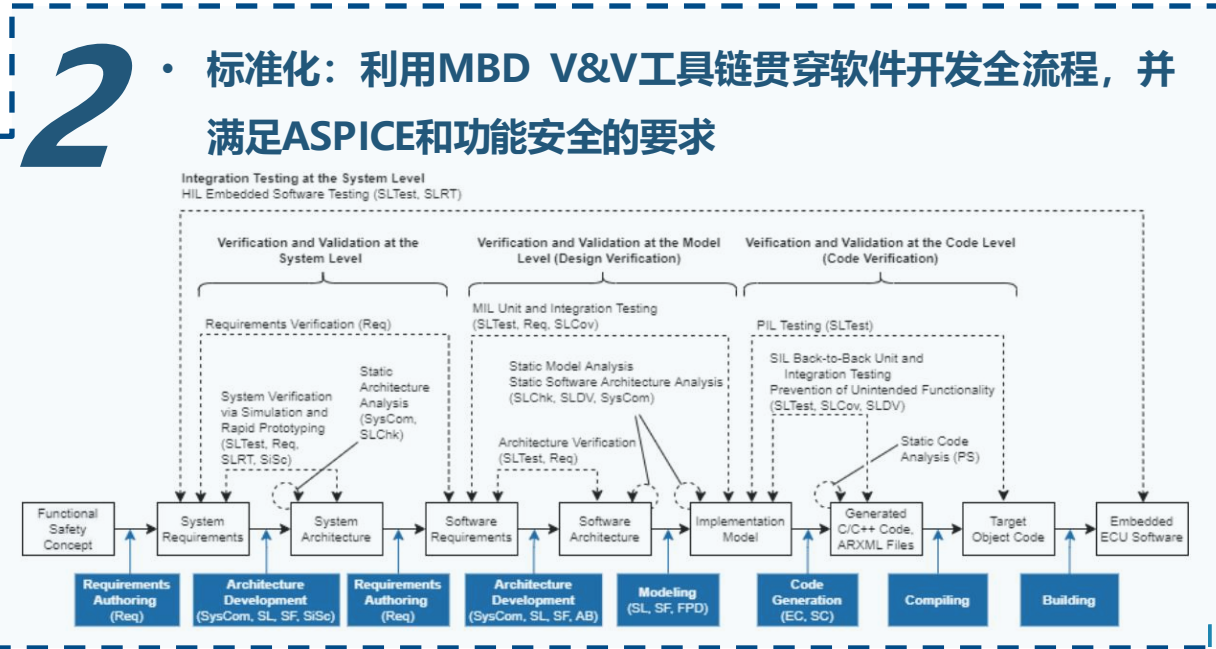
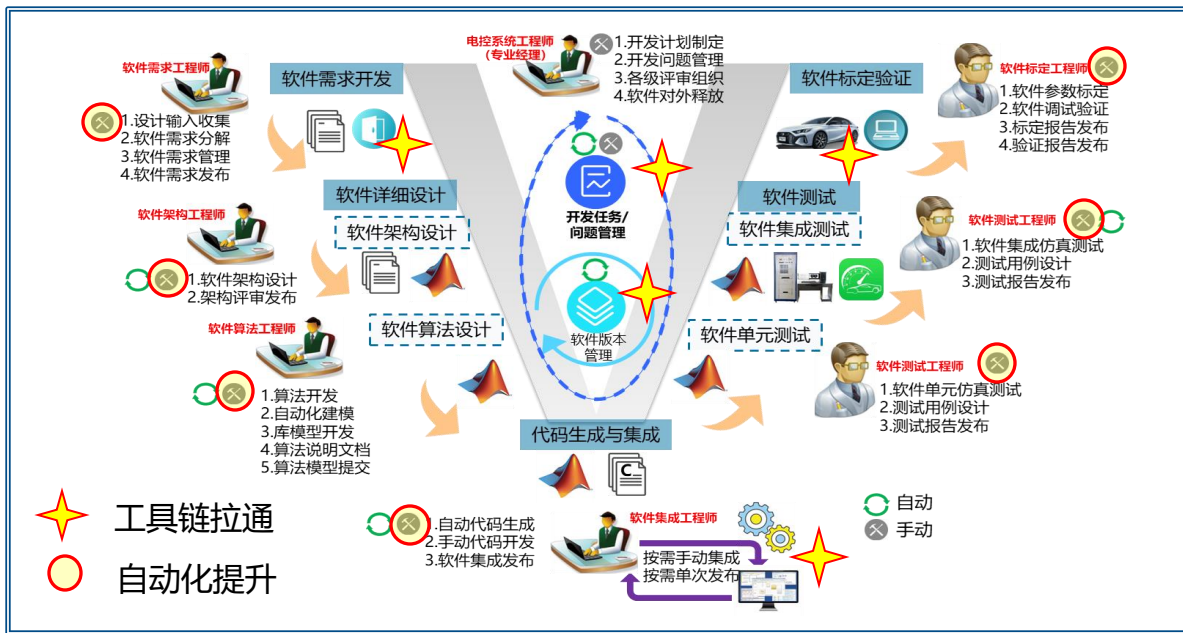
03 | **业务活动数字化**

04 | 过程交付数字化

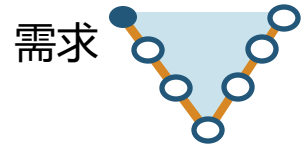
05 | 平台部署与总结

MBD端到端
平台解决方案

软件开发业务活动数字化 | 工具链的数字化提升：统一化、标准化、自动化

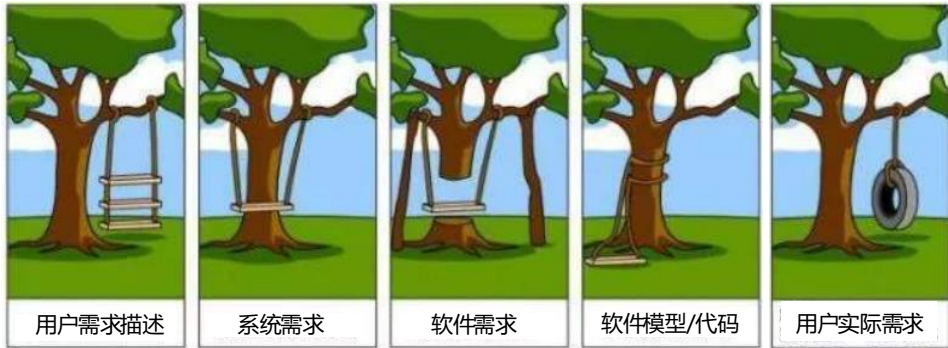


软件开发业务活动数字化 | 工具链优化：需求端



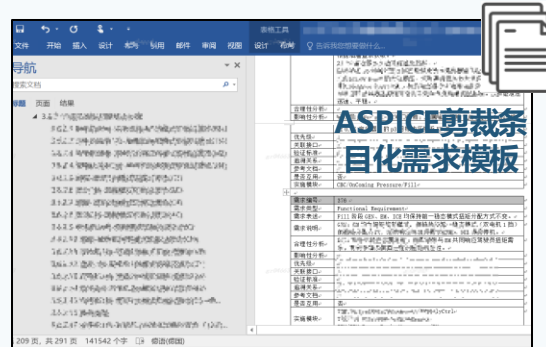
需求工程：

需求文档化、条目化、追溯、验证、管理

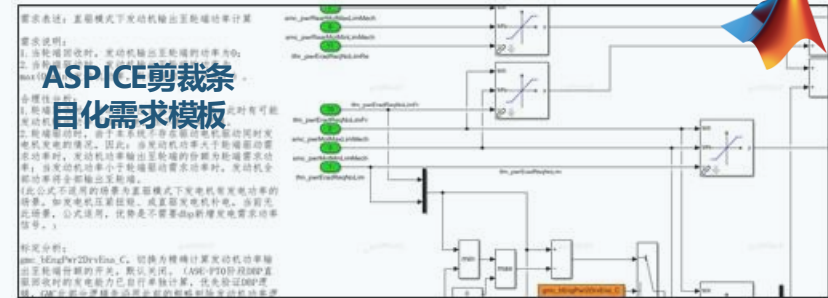


特征	要求
一致性	需求和模型一致性要便于时刻相互校验（正向审视），测试之前就尽量保证需求和实现是一致的
追溯性	需求-模型-测试双向追溯，自动化映射关联
易维护	需求易于修正、再校验、再关联

1 基础实践：条目化需求文档 一致性难保证-追溯性差-维护性差



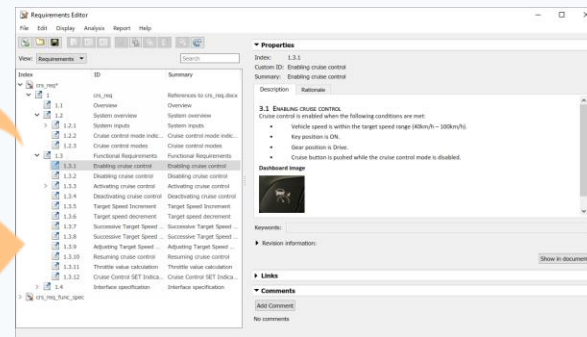
2 进阶实践：条目化模型注释 一致性易保证-追溯性较好-维护性一般（建模和需求交付绑定）



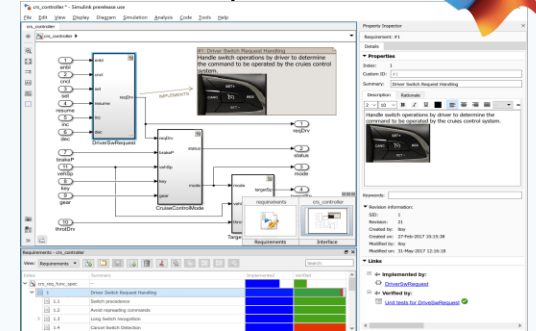
外部需求
SYS

内部需求
SWR

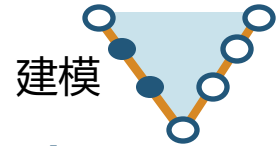
3 最佳实践：MATLAB Requirement 内外部工具协同 一致性易保证-追溯性好-维护性好



MATLAB Requirement



MBD平台(需求端)：打通外部需求工具链，统一内部需求工具链



软件开发业务活动数字化 | 工具链优化：算法/模型端

- 软件规模增大、需求多样导致功能/子功能数量成倍增加，建模工作量大。(Eg. 1个车型配置, SVN1350/y, 5v/d)
- 建模自动化 & 规范检查自动化 降低建模耗时和出错率，同时提高建模效率和质量 (提质增效)。

自动化建模：MBD建模助手工具

建模出错率



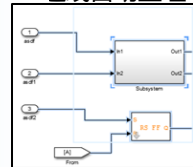
建模耗时率



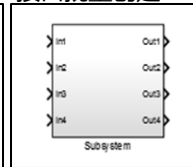
常用工具示例

1. Port接口批量生成;
2. 数据字典自动生成;
3. 一键建模规范检查;
4. 模块属性自动调整;
5. bus信号自动查找和链接
6. 接口自动检查;.....

车线自动整理



接口批量创建



属性集自动设定



库模块批量创建复用

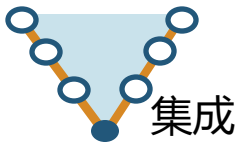
接口自动全局查找并连线

- 逻辑/算法建模
 - 信号连线(大批量)
 - 数据字典定义(大批量)
 - 架构、时序
 - 其他
- 脚本自动化 辅助建模 > 70项脚本

自动化检查：MBD模型代码规范检查工具

MBD平台(模型端)：提供统一的建模自动化辅助及检查工具

- 建模规范检查
- 设计缺陷检查
- 代码规范检查
- 代码缺陷检查



软件开发业务活动数字化 | 工具链优化：集成端 (一)

- ❑ 软件版本增多，验证频次增加，集成需求增多、耗时； (Eg. 1次编译时长20min-90min/v → 2-4h/v)
- ❑ 集成需求不定时、不定地点，人工按需集成方式逐渐难以适应业务的发展；
- ❑ 云端持续集成和发布，解放集成&测试人力，提高软件迭代效率。

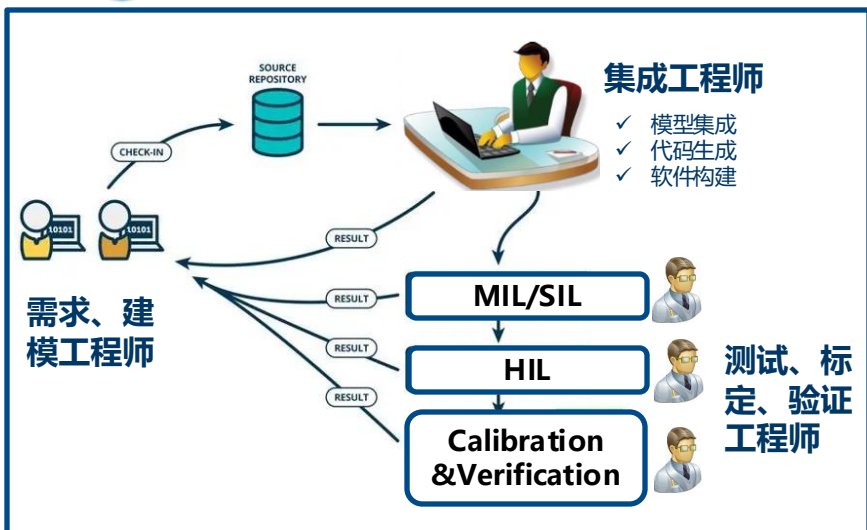


需求扩增

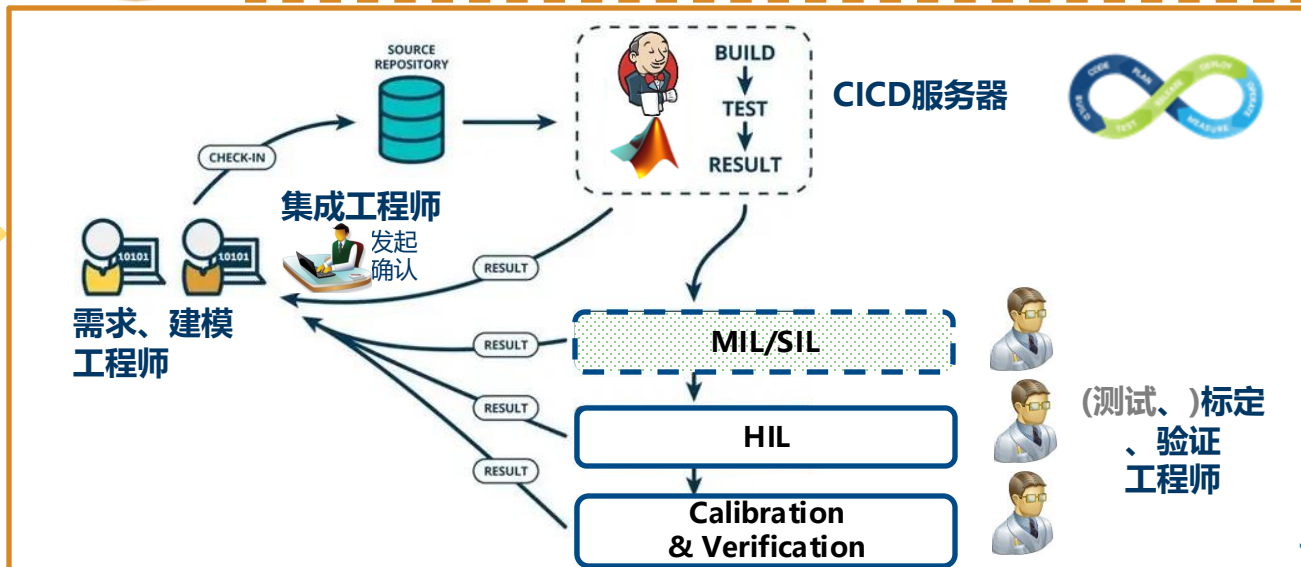


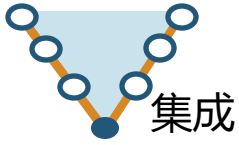
按需集成、测试、发布，手动单步执行

MBD平台(集成端)：持续集成、测试、发布，流水线自动执行



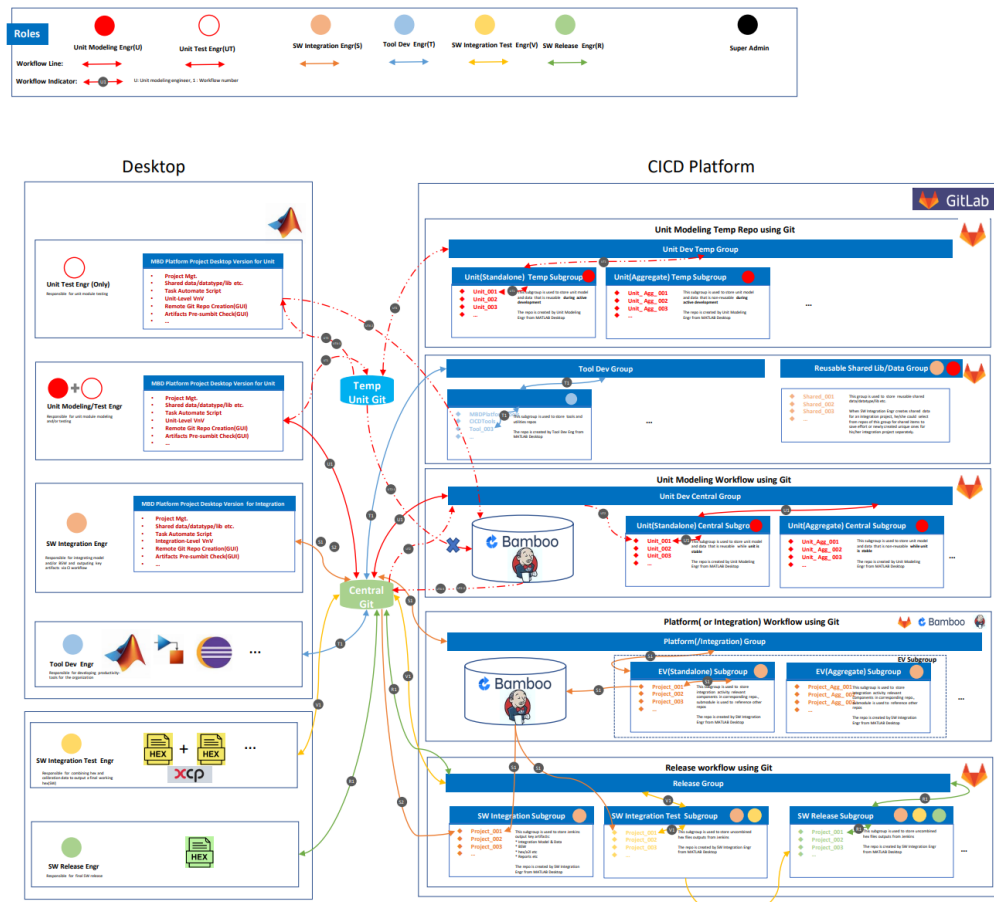
加快制品速率



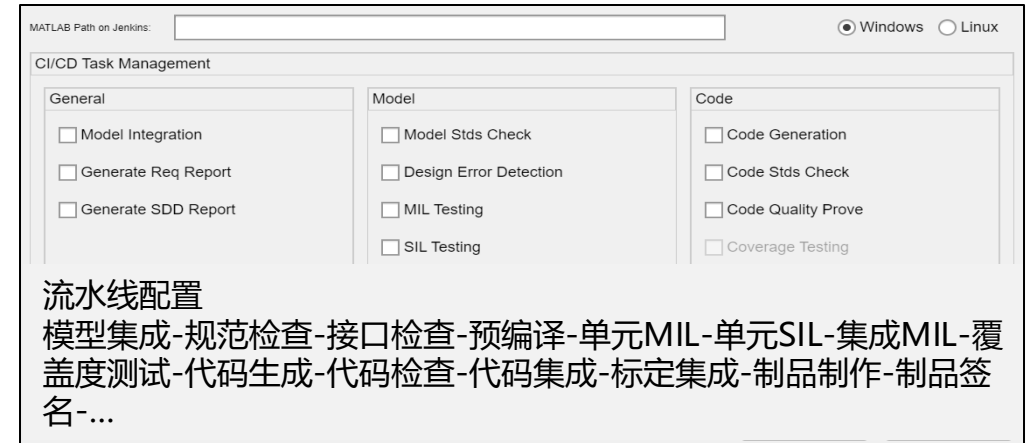
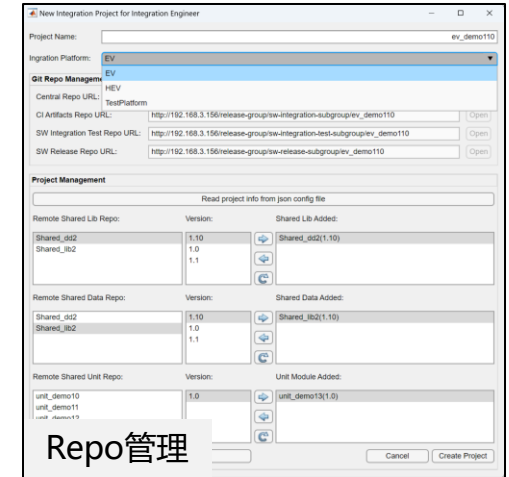
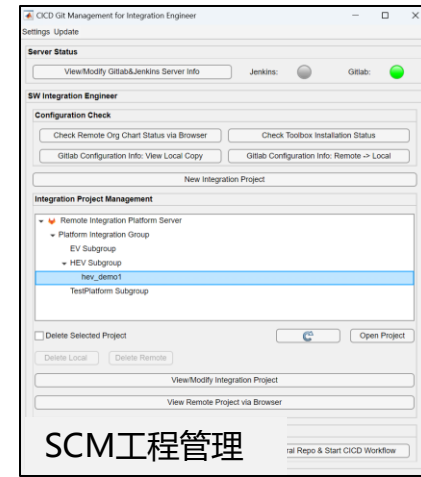


软件开发业务活动数字化 | 工具链优化：集成端 (二)

- ❑ CICD前端界面工具，打通SCM工具(SVN/Git)并自定义配置流水线任务（模型/代码检查、集成、测试、编译、发布等），开发人员一站式操作。
- ❑ CICD云端部署系统，调用MBD V&V工具链及自定义脚本，全天候持续执行前端预设的流水线任务。



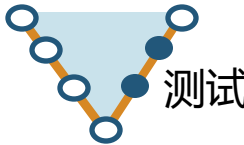
MBD平台CICD云端部署架构



流水线配置

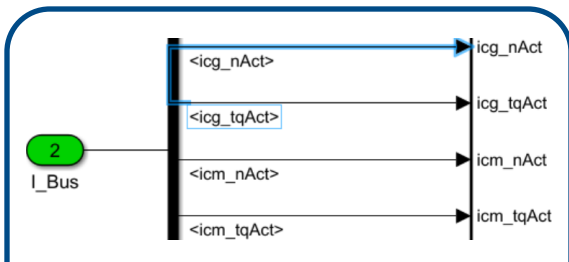
模型集成-规范检查-接口检查-预编译-单元MIL-单元SIL-集成MIL-覆盖度测试-代码生成-代码检查-代码集成-标定集成-制品制作-制品签名-...

MBD平台CICD前端界面工具



软件开发业务活动数字化 | 工具链优化：测试端（一、单元测试）

- 测试是软件质量的守门员；
- 高质量测试挑战巨大：海量测试任务，测试用例不足，测试环境/用例不标准，测试过程不敏捷。
- 高效敏捷和标准化的测试工具能够解决上述问题。



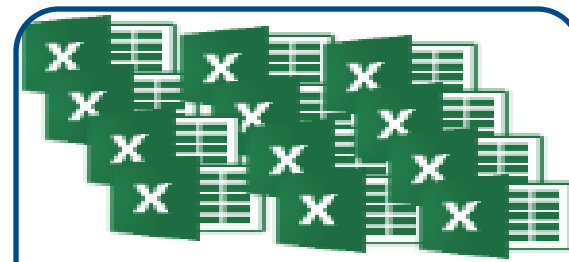
接口测试覆盖度



回归测试不足



冒烟测试不够敏捷



MIL/HIL/实车测试用例不统一

MBD平台（测试端）：测试高度自动化、标准化

Simulink Test+定制界面&脚本+ workflow

单元测试工具

基于需求的测试用例设计
测试用例调试

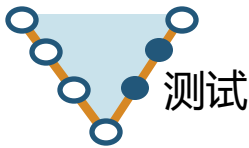
统一测试用例：双向自动导入和管理

测试项目	一级子项	二级子项	用例编号	用例名称	需求描述	用例描述	测试步骤	测试预期	标定方案	需求ID

测试结果管理

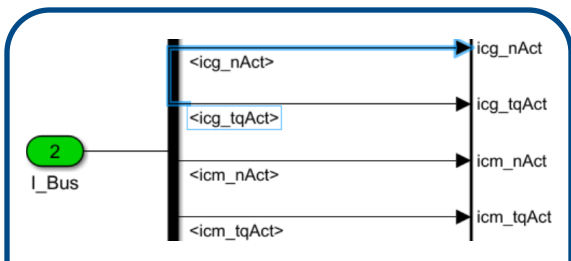
测试评估报告

Unit Test Dashboard



软件开发业务活动数字化 | 工具链优化：测试端 (二、集成测试)

- 新能源整车控制器VCU对接其他控制节点数多，交互变更频繁，“控制+对象”的**集成测试模型反复重构**。
- 统一的**集成测试框架+测试模型自动集成**，**标准化快速构建集成测试环境**，提高集成测试效率。



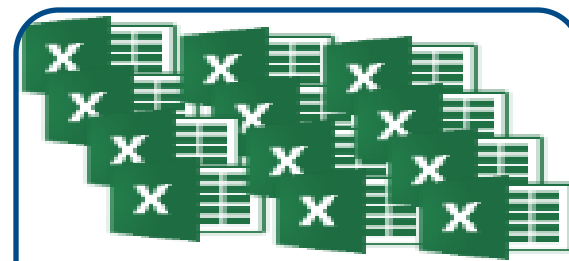
接口测试覆盖度



回归测试不足



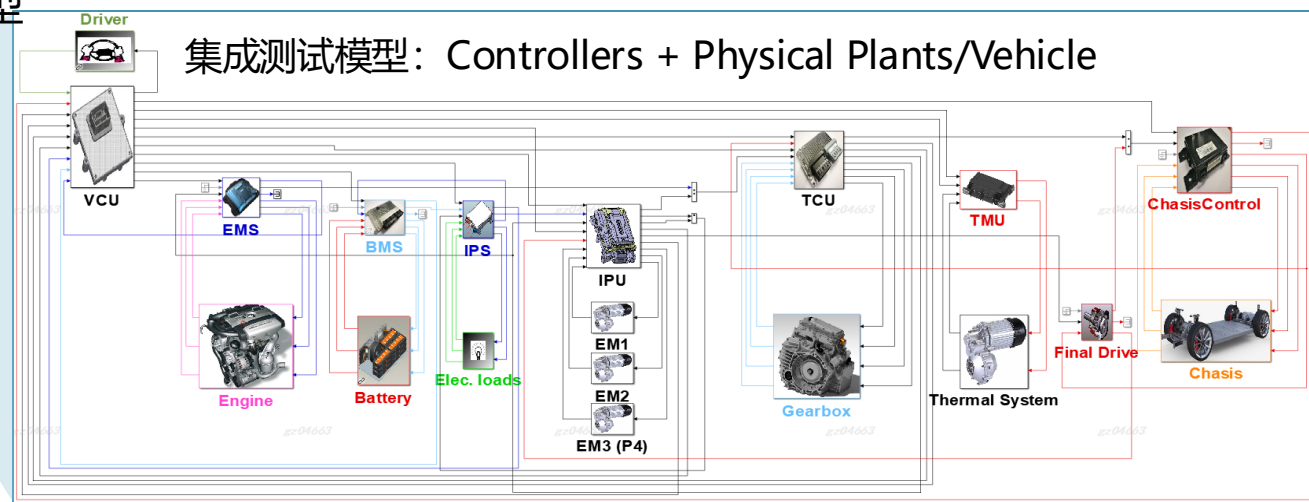
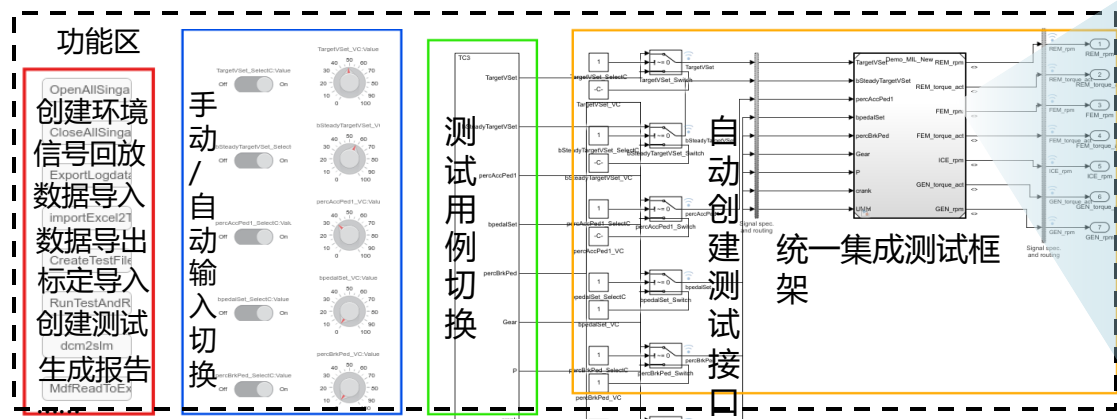
冒烟测试不够敏捷



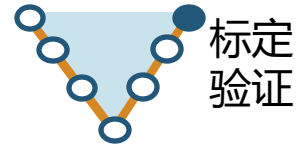
MIL/HIL/实车测试用例不统一

MBD平台 (测试端)：测试高度自动化、标准化

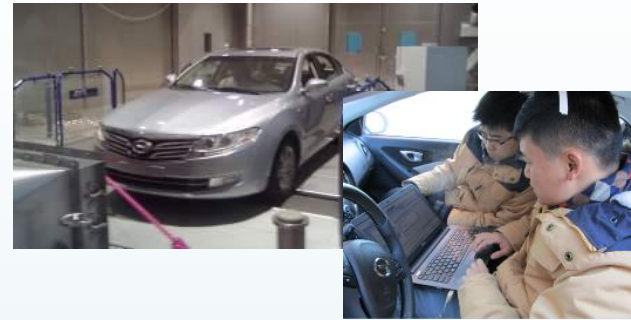
集成测试环境：通用集成测试框架 + 自动集成控制&对象模型



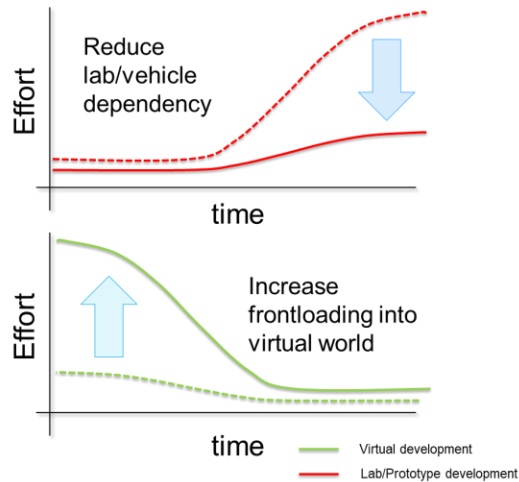
软件开发业务活动数字化 | 工具链优化：标定验证端(虚拟替代需求)



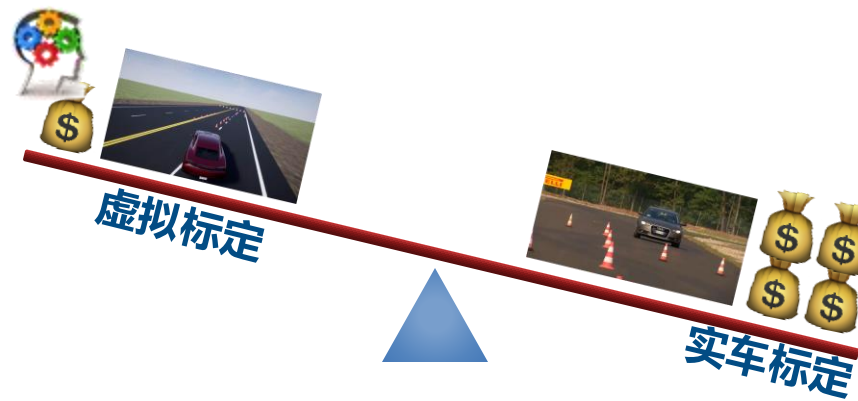
- **成本**：越来越多的功能/变量，导致样车需求增加，实车试验成本增加；
- **效率**：过分依赖现地环境（温/湿度、海拔、地形、路况等），效率低；
- **传承**：标定专家的经验知识缺少有效载体，传承效率低。



MBD平台（标定端）：虚拟样车模型 & 虚拟标定工具

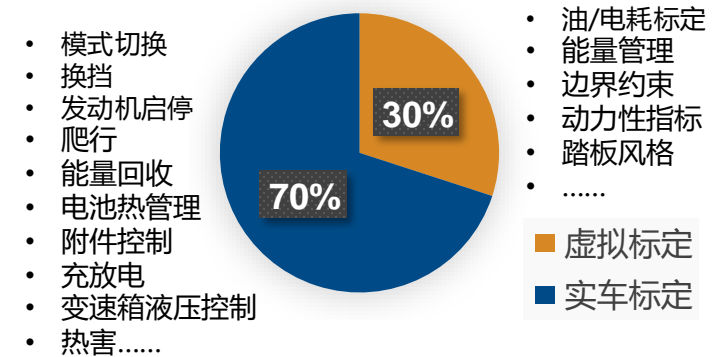


标定过程转移到虚拟样车（模型）



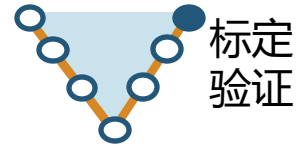
目标：样车减少30%，周期缩减15%

标定虚拟化潜力

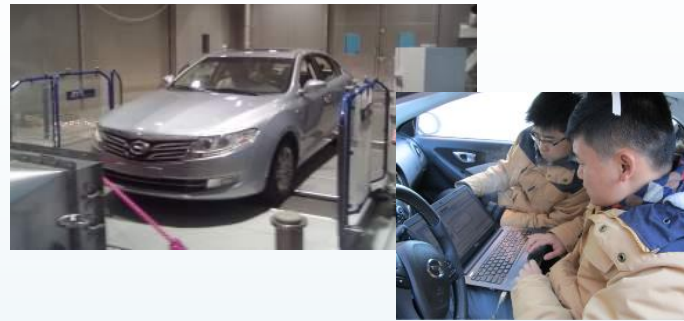


实车标定次数替代

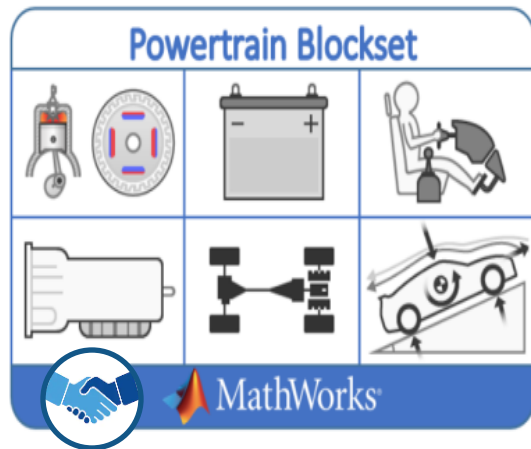
软件开发业务活动数字化 | 工具链优化：标定验证端(虚拟替代方案)



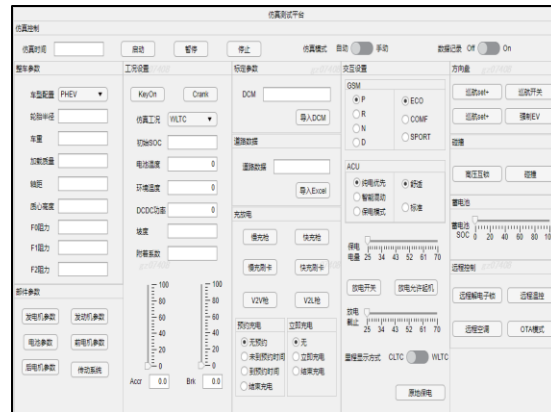
- **建模**：面向整车控制的物理建模
- **定制**：定制化面向对象调参界面、实车测试数据环境导入、自动化数据处理和标定报告生成
- **优化**：差异化功能标定参数的自动化寻优工具



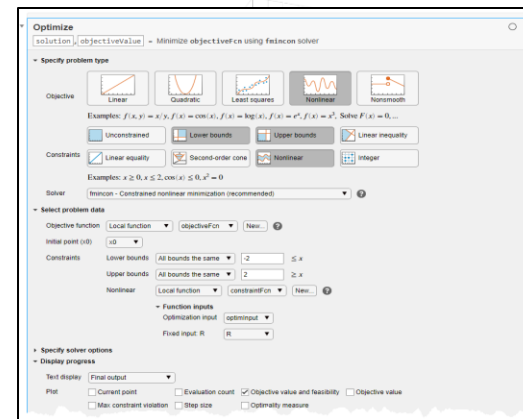
MBD平台 (标定端)：物理模型、虚拟标定界面 (手动&自动)、寻优工具



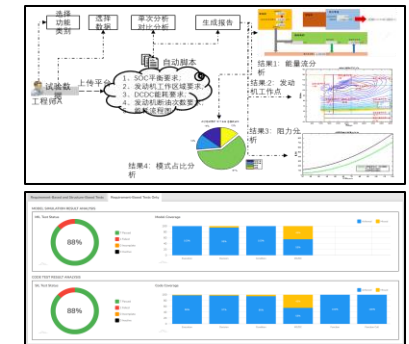
- 共用集成测试环境
- 基于任务差异化物理建模&降维



- 前端虚拟标定界面支持手动调参，重现实车标定场景



- 基于不同功能标定特征寻找合适优化方法自动寻优，云端执行。



- 各标定任务的数据处理脚本固化，报告自动生成；
- 专家经验由模型、优化方法和数据脚本和相关报告共同承载。

目录

Outline

01 | 数字化软件开发的内外部需求

02 | 开发流程数字化

03 | 业务活动数字化

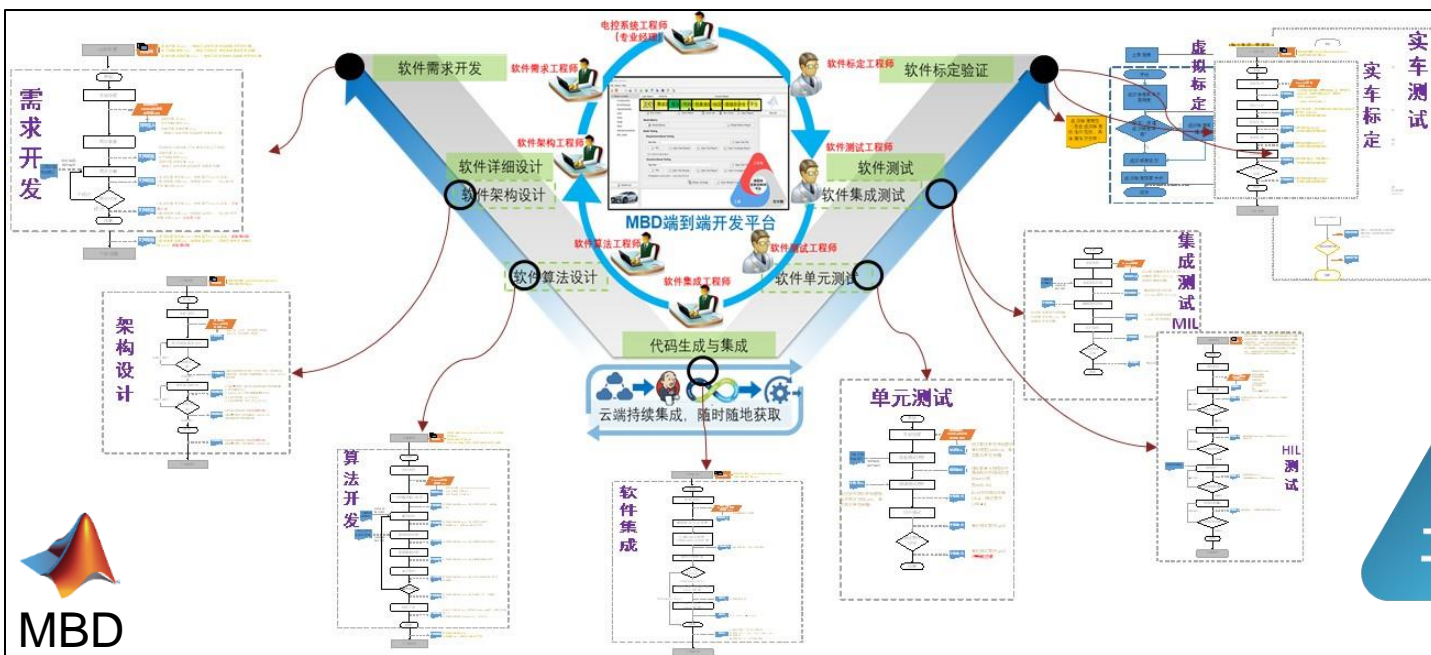
04 | 过程**交付**数字化

05 | 平台部署与总结

**MBD端到端
平台解决方案**

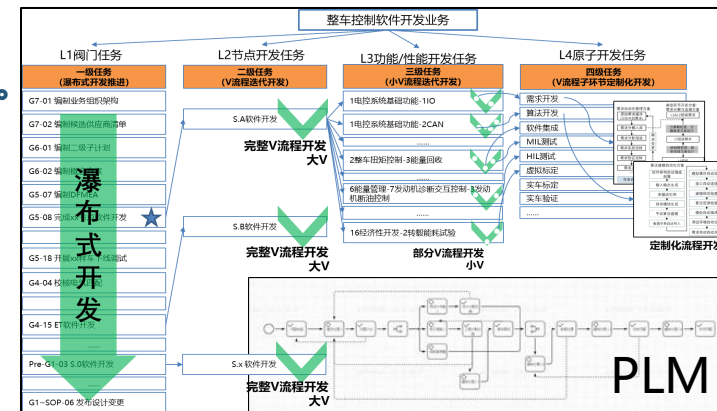
软件开发过程交付数字化 | 端到端标准化交付

- ❑ 分解-执行-交付：PLM发起流程、MBD执行任务、PLM过程交付；
- ❑ 固化模板：上下游工作输入输出物固化，减少传递误差；
- ❑ 交付检查：PLM设定过程交付强制(上传)检查项，节点流转标准管控，交付自动归档。

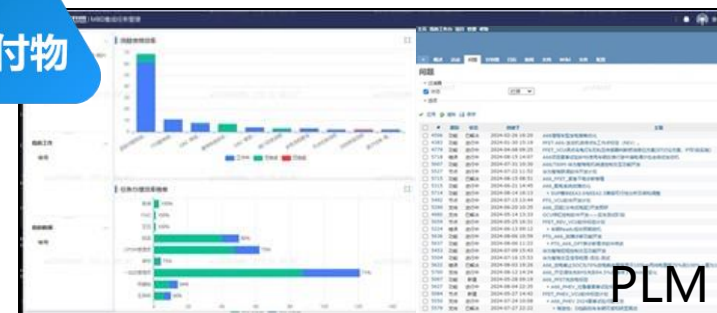


L4 原子开发任务基于MBD平台按流程执行

- ✔ 流程效率提升
- ✔ 业务经验积累、知识体系建立
- ✔ 满足认证要求



L1-L4流程任务层级划分



L1-L4流程任务及交付物一览界面

目录

Outline

01 | 数字化软件开发的内外部需求

02 | 开发流程数字化

03 | 业务活动数字化

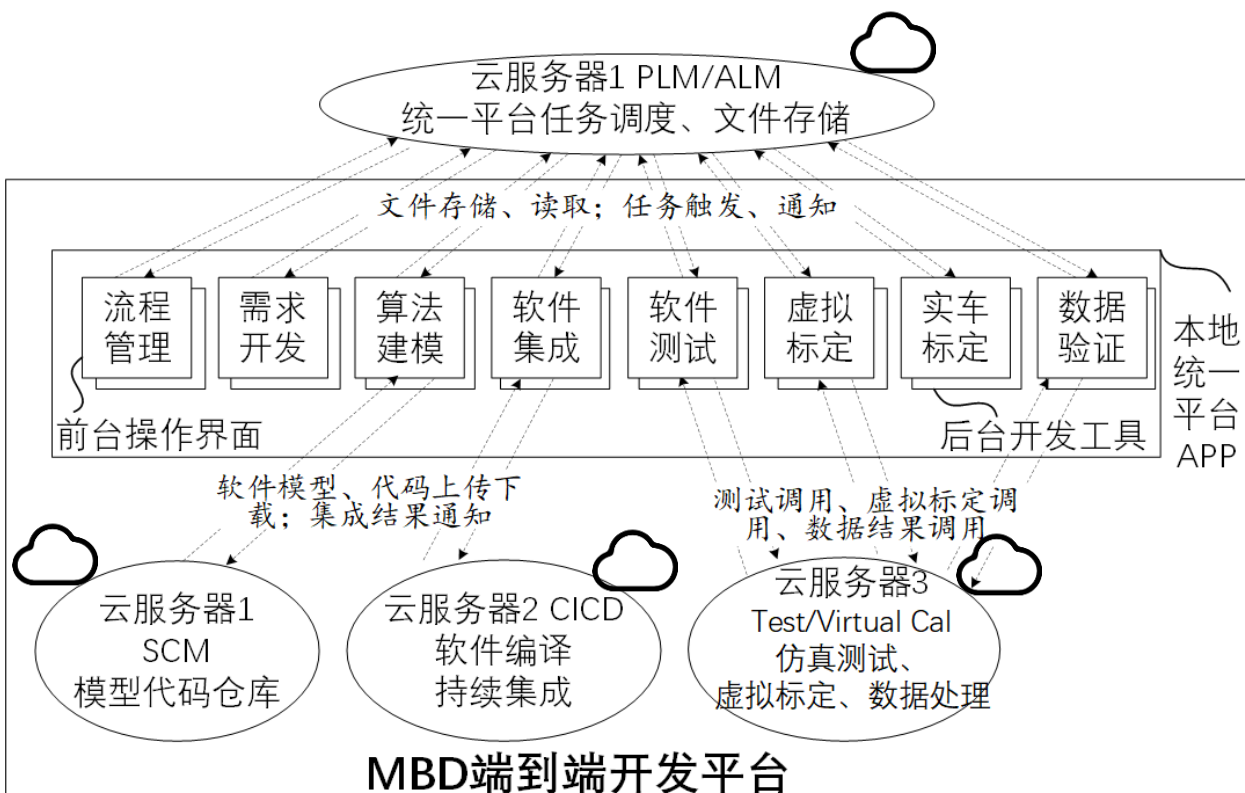
04 | 过程交付数字化

05 | 平台部署与总结

**MBD端到端
平台解决方案**

端到端MBD平台部署与总结

平台部署



总结

- 1 端到端MBD软件开发数字化实践的核心是流程、工具、交付三者的平台化统一和融合，目的是提质降本增效。
- 2 开发流程数字化的关键是固化任务/流程层级，流程标准化和自动化流转管控。
- 3 业务活动数字化的关键是工具链统一、联通，打通工具壁垒，最大程度自动化，关键业务虚拟化。
- 4 过程交付数字化的关键是交付物的标准固化、依托任务自动归档。

2024 MathWorks 中国汽车年会

Thank you

