MATLAB EXPO 2021

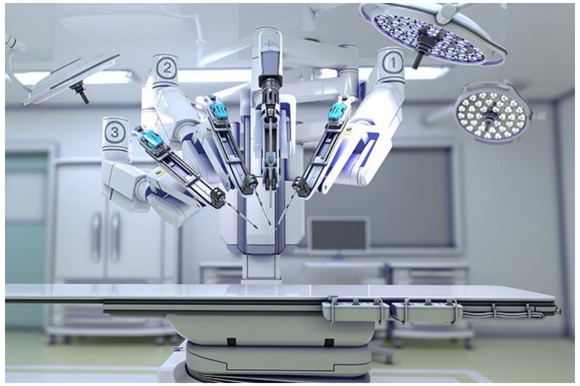
基于模型的设计让嵌入式软件开发满足认证标准

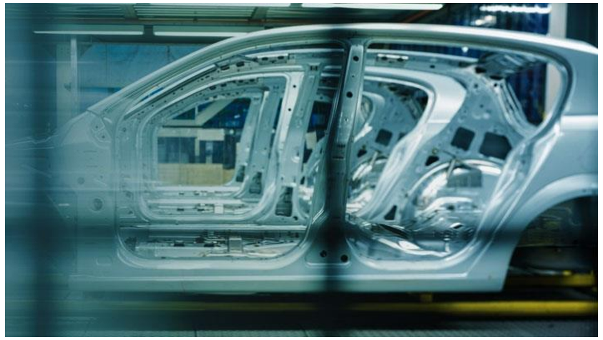
徐天皓， MathWorks

MathWorks

软件安全在各个行业中都非常重要

- 软件错误很常见
  - 汽车行业
  - 医疗设备
- 召回非常昂贵
  - MCAS故障
  - 髋关节置换术
  - 踏板故障
  - 安全气囊问题





Source: MedTech Dive, Ars Technica, Automotive News, Allianz, Sibros, Medical Design, New York Times, Bloomberg

为什么要做功能安全?

安全关键系统

最佳实践和准则

功能安全标准

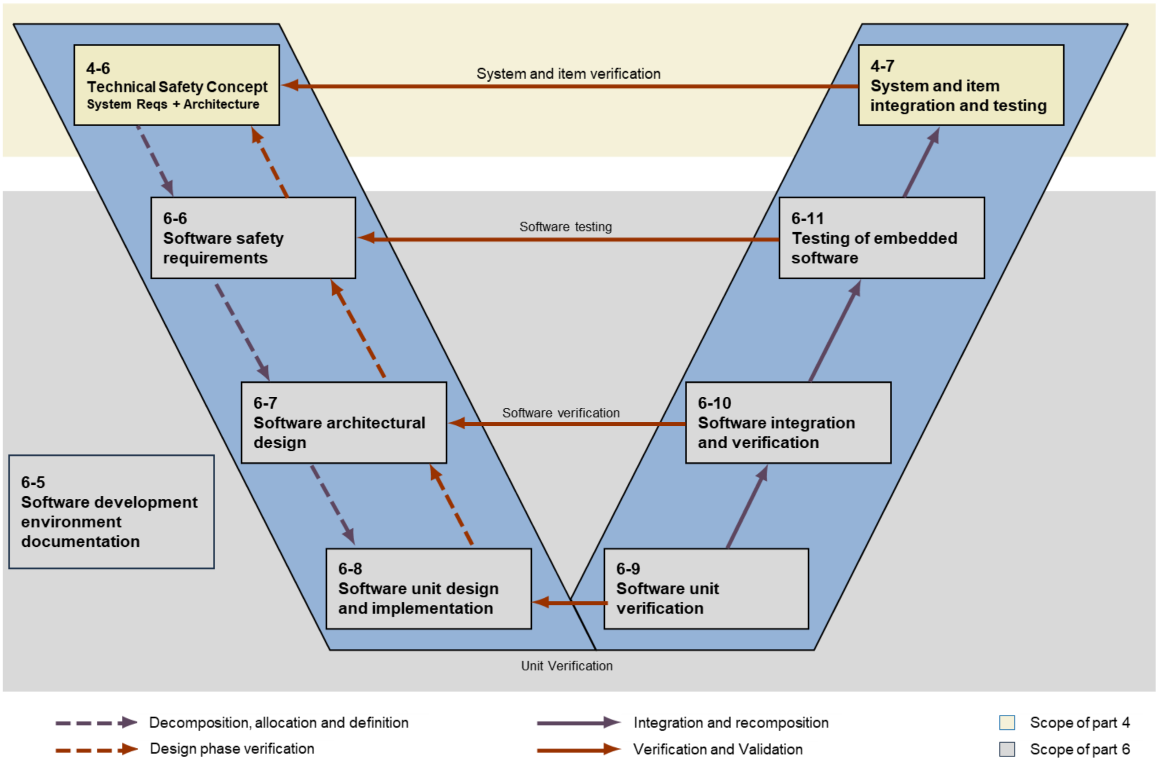
认证

要点

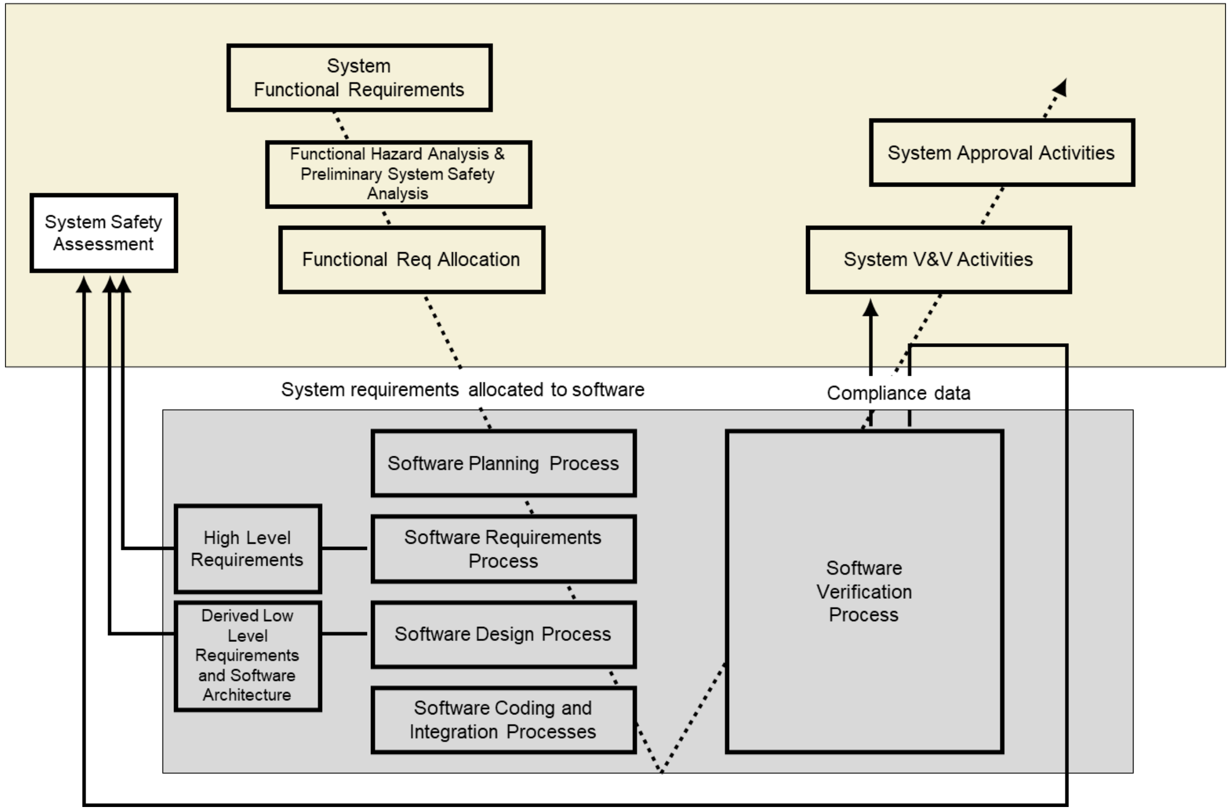
将参考工作流程与基于模型的设计结合使用，以达到认证标准

- 创建需求、架构、设计、测试及代码之间的**追溯关系**
- 通过持续测试在**早期**发现错误
- 通过自动代码生成**减少编码错误**
- **自动**生成文档和报告以进行评审和审计

符合认证需求的V模型

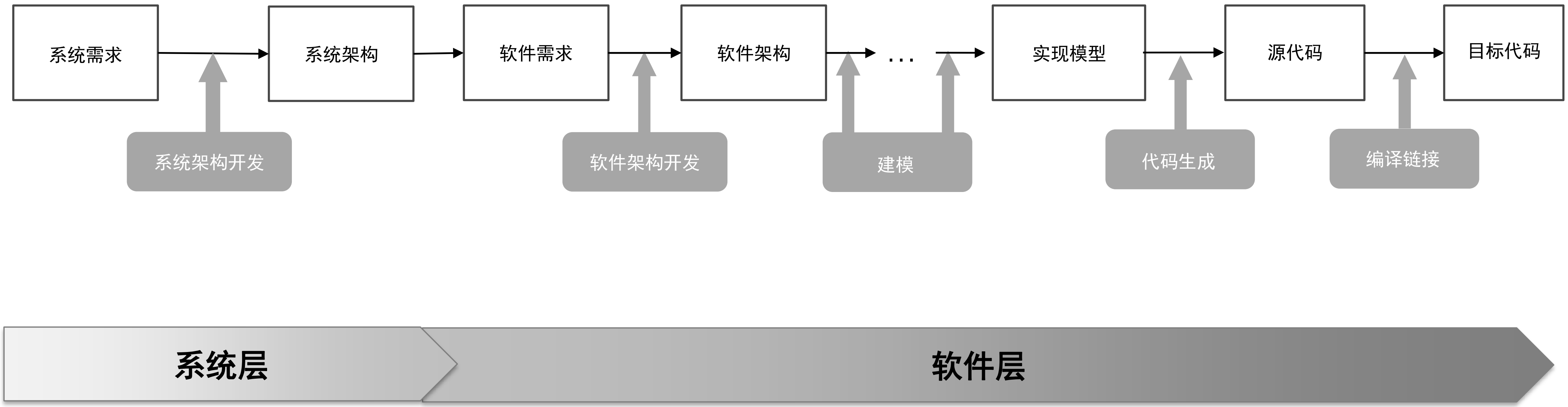


ISO 26262: 2018

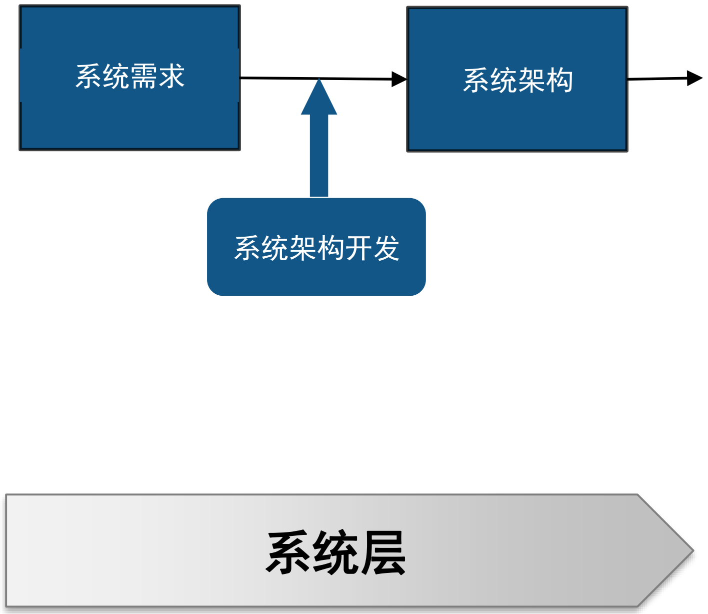


DO 178C, ARP 4754A, ARP 4761

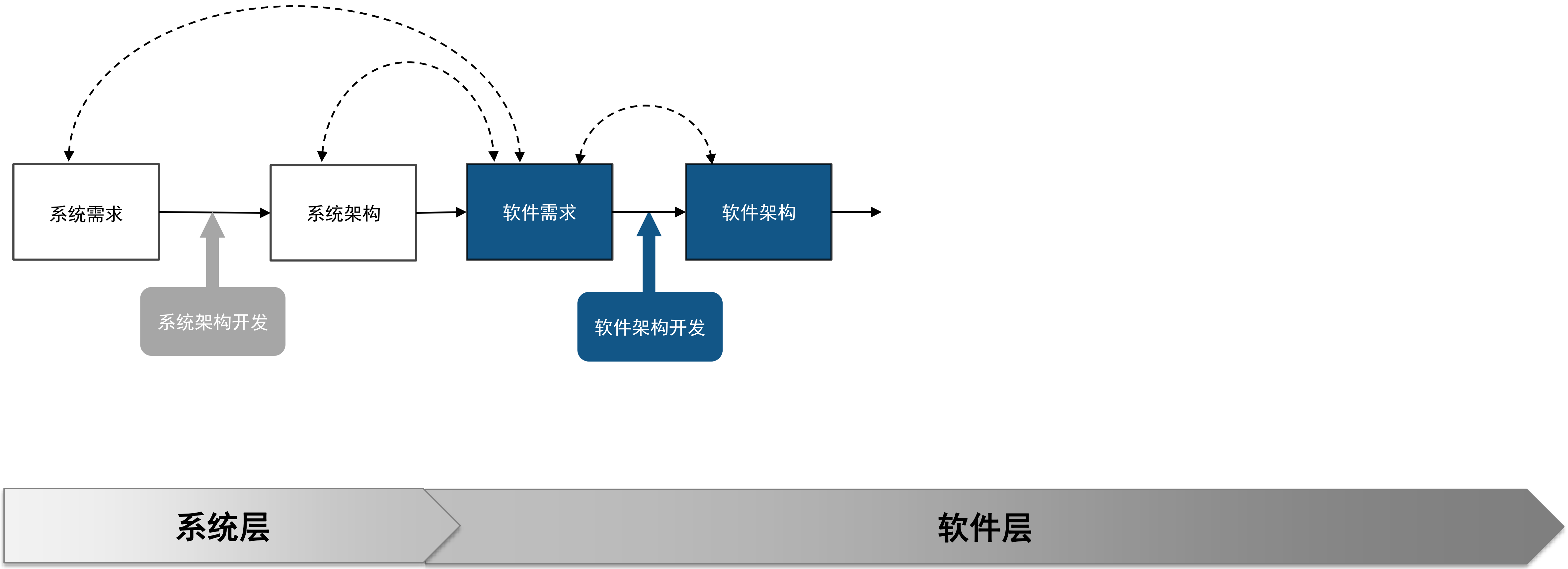
适用于认证的参考工作流程



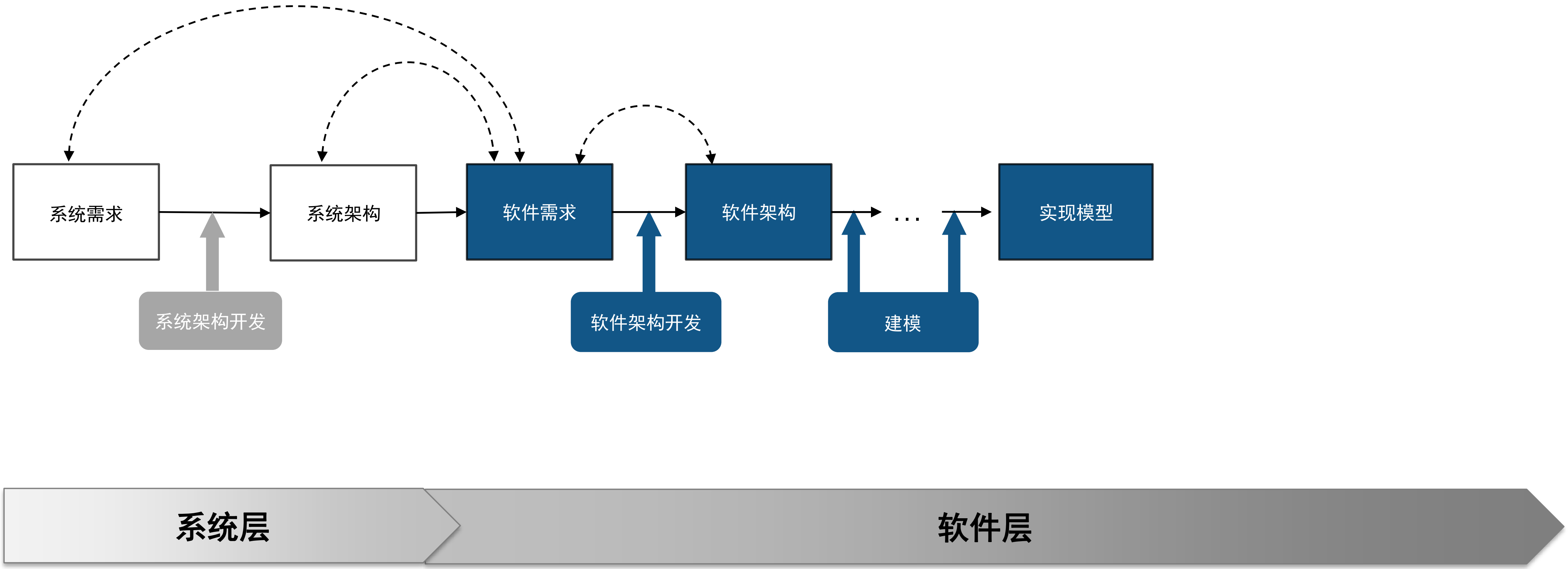
适用于认证的参考工作流程 – 系统需求与架构



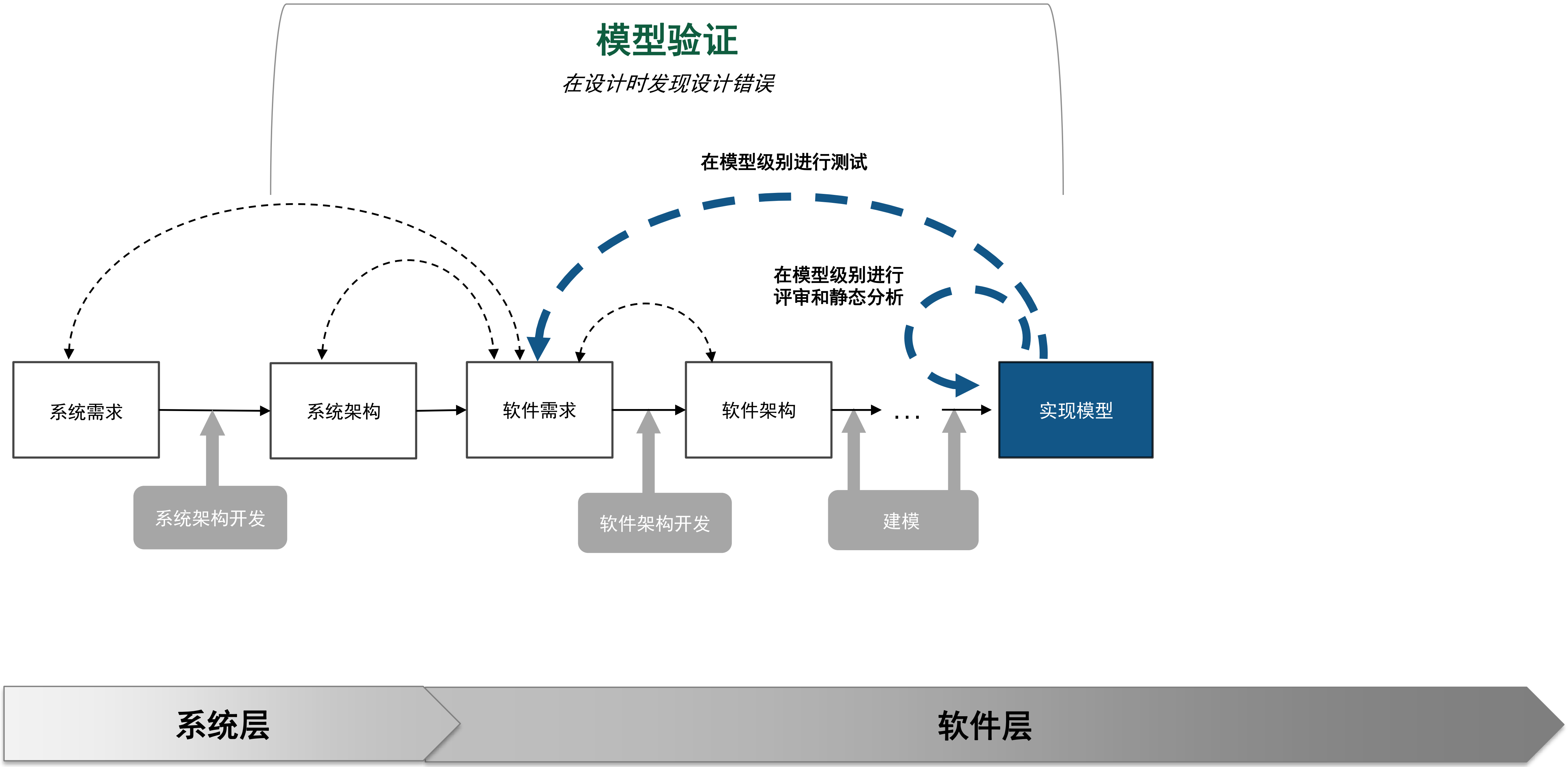
适用于认证的参考工作流程 – 软件需求与架构



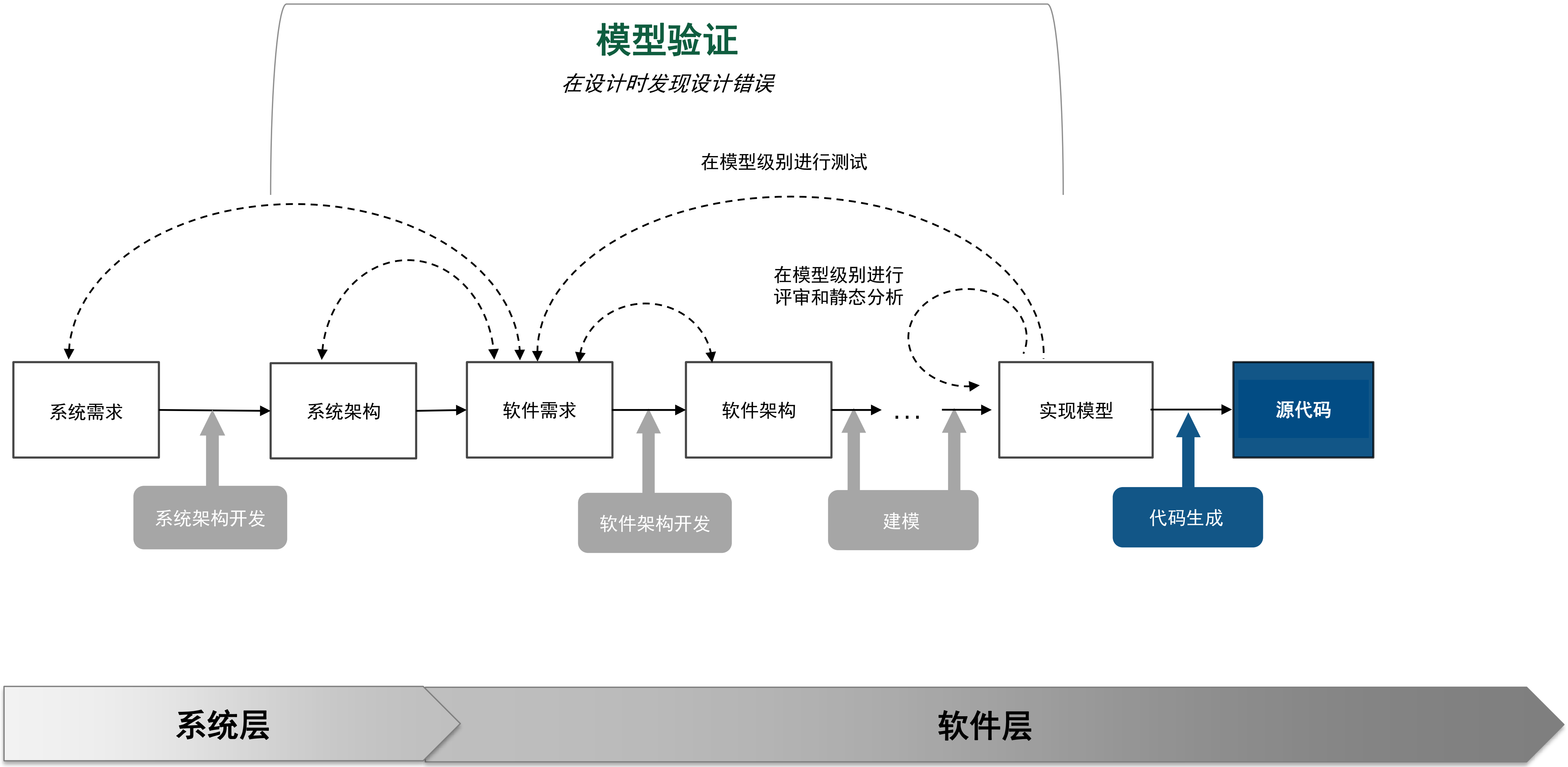
适用于认证的参考工作流程 – 详细设计



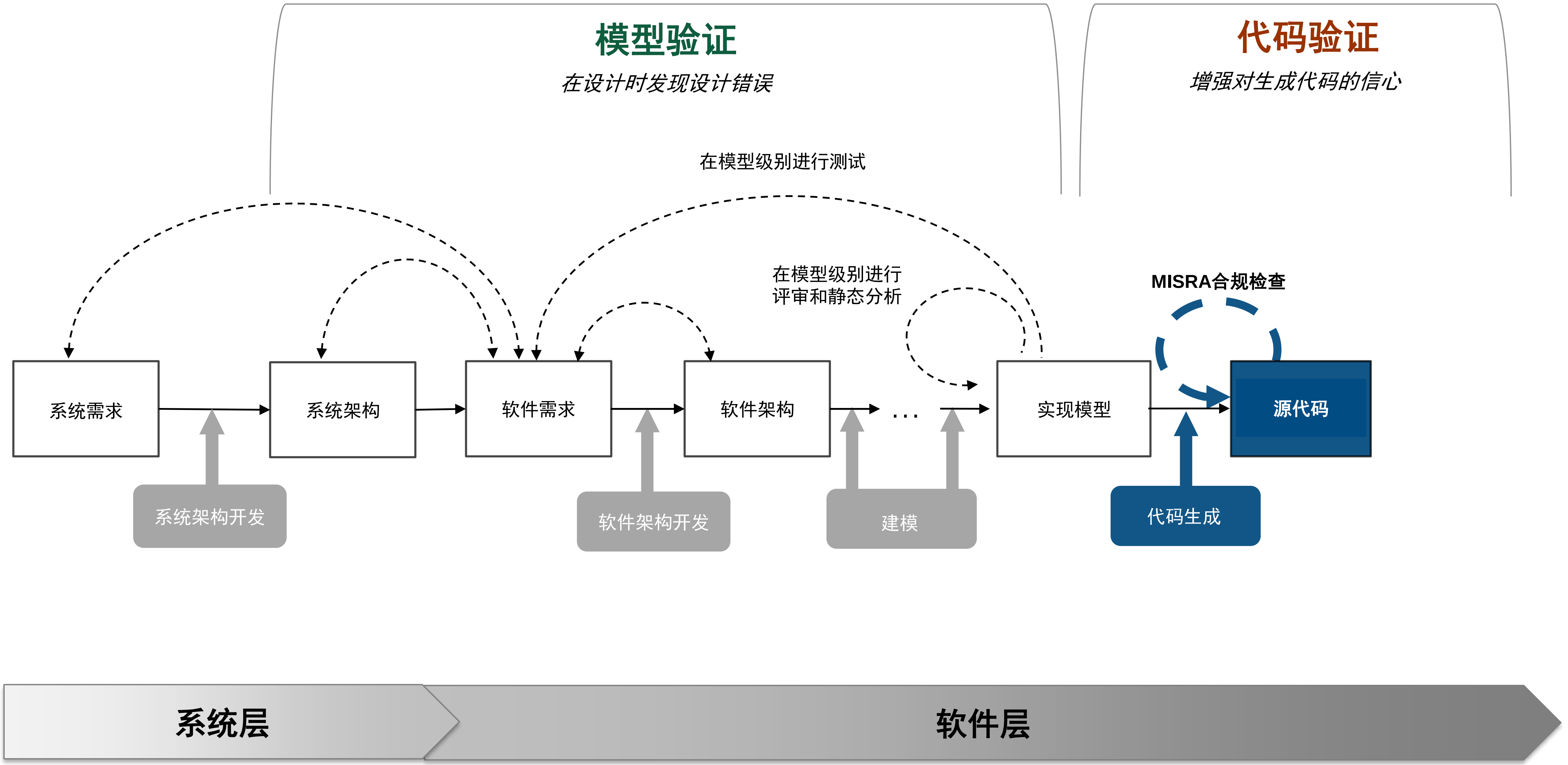
适用于认证的参考工作流程 – 模型验证



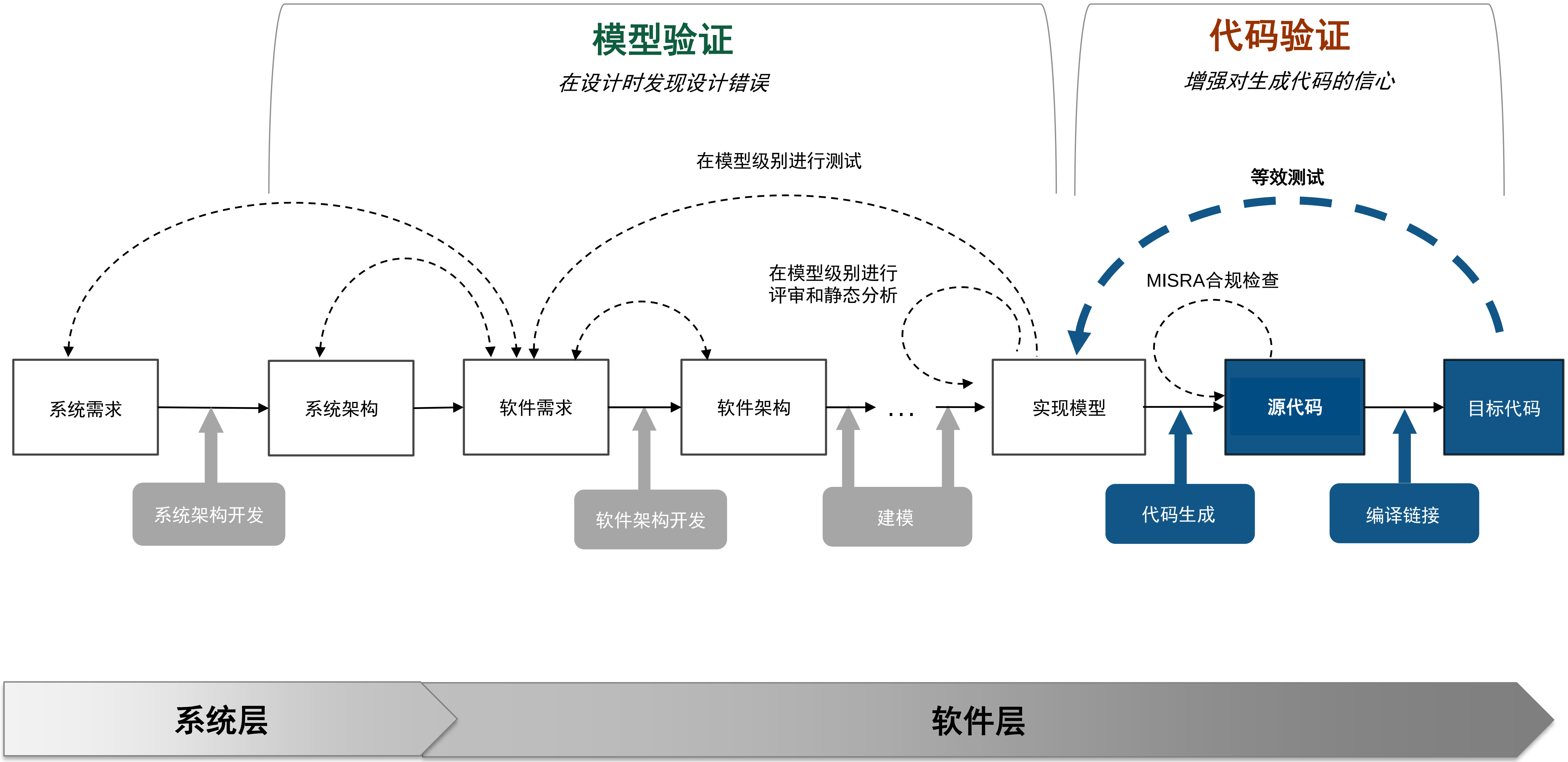
适用于认证的参考工作流程 – 代码生成



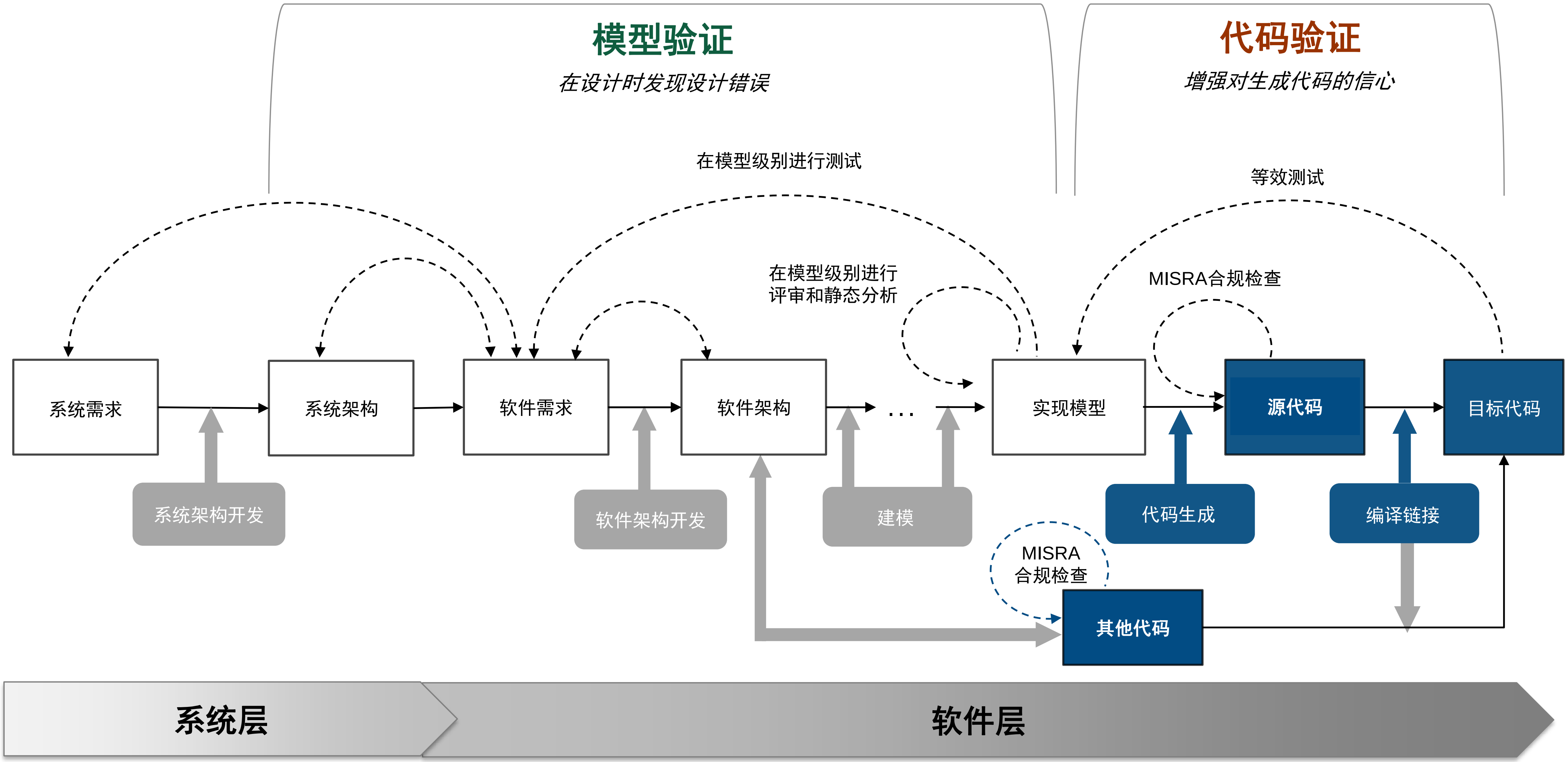
适用于认证的参考工作流程 – 代码静态分析验证



适用于认证的参考工作流程 – 代码动态测试验证



适用于认证的参考工作流程 – 代码静态分析验证



投资回报 结果

Continental

验证时间最多减少50%

Leonardo

重新认证周期减少了90%以上

Tessella

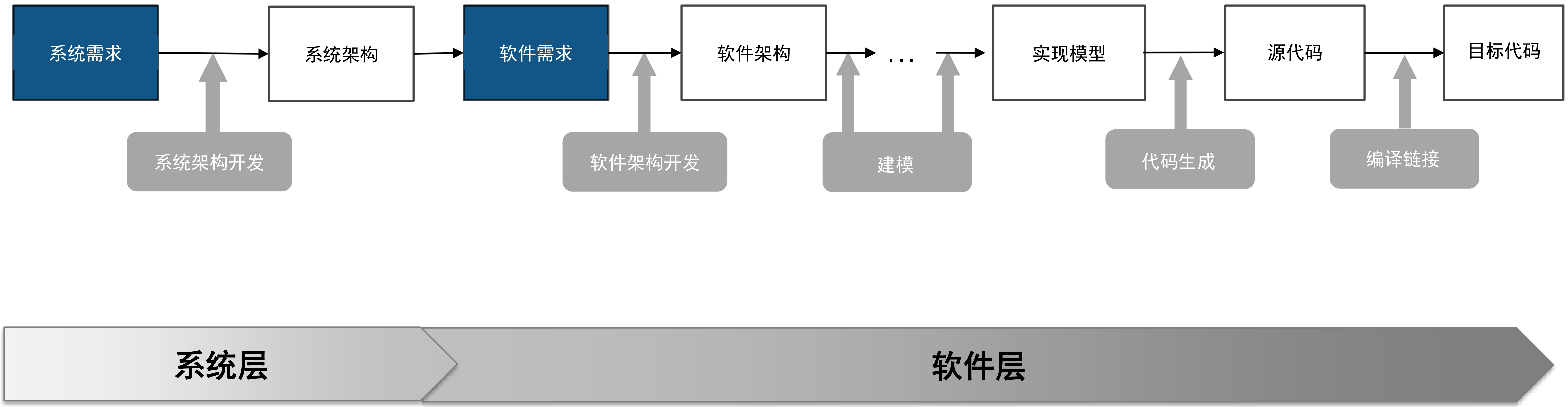
模型可在后续项目中重复使用，最多可将设计工作量减少80%

Corindus

开发时间减少了一半，工程工作量减少了80%

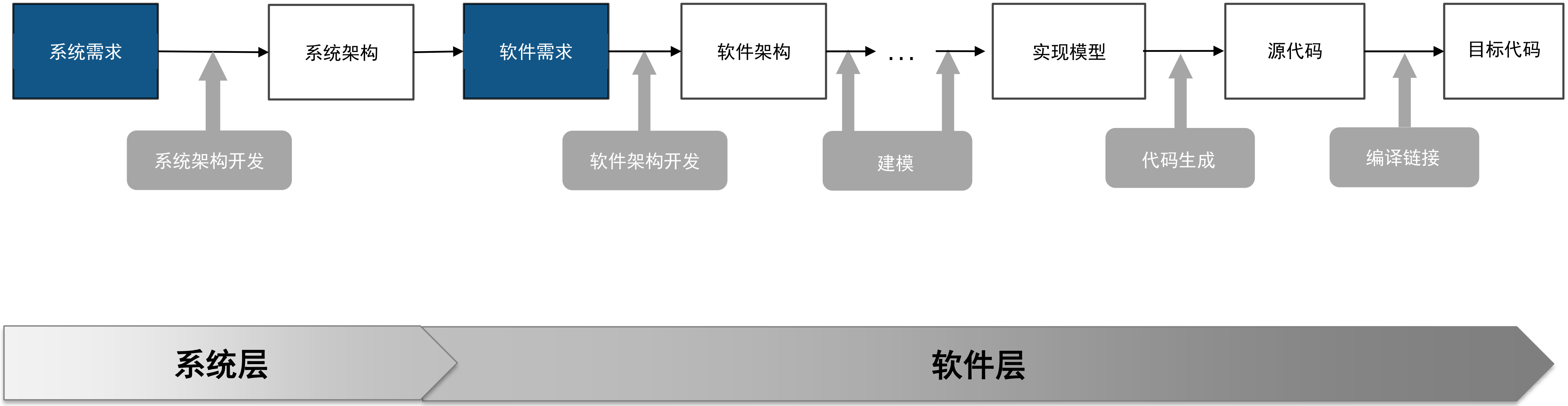
更多用户故事：www.mathworks.com/company/user_stories.html

需求

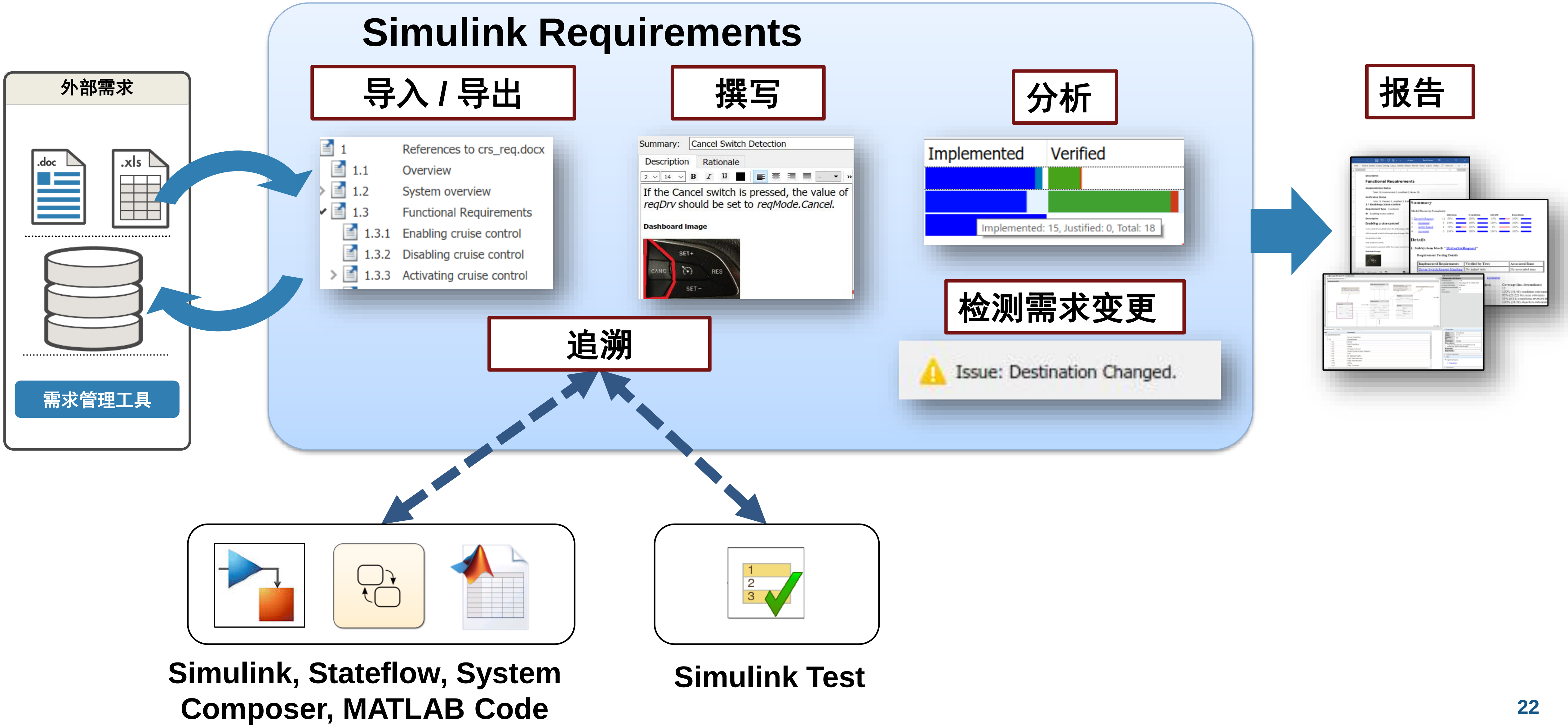

系统需求
系统架构
软件需求
软件架构
…
实现模型
源代码
目标代码
系统架构开发
软件架构开发
建模
代码生成
编译链接
系统层
软件层

需求跨越整个工作流程的多个级别

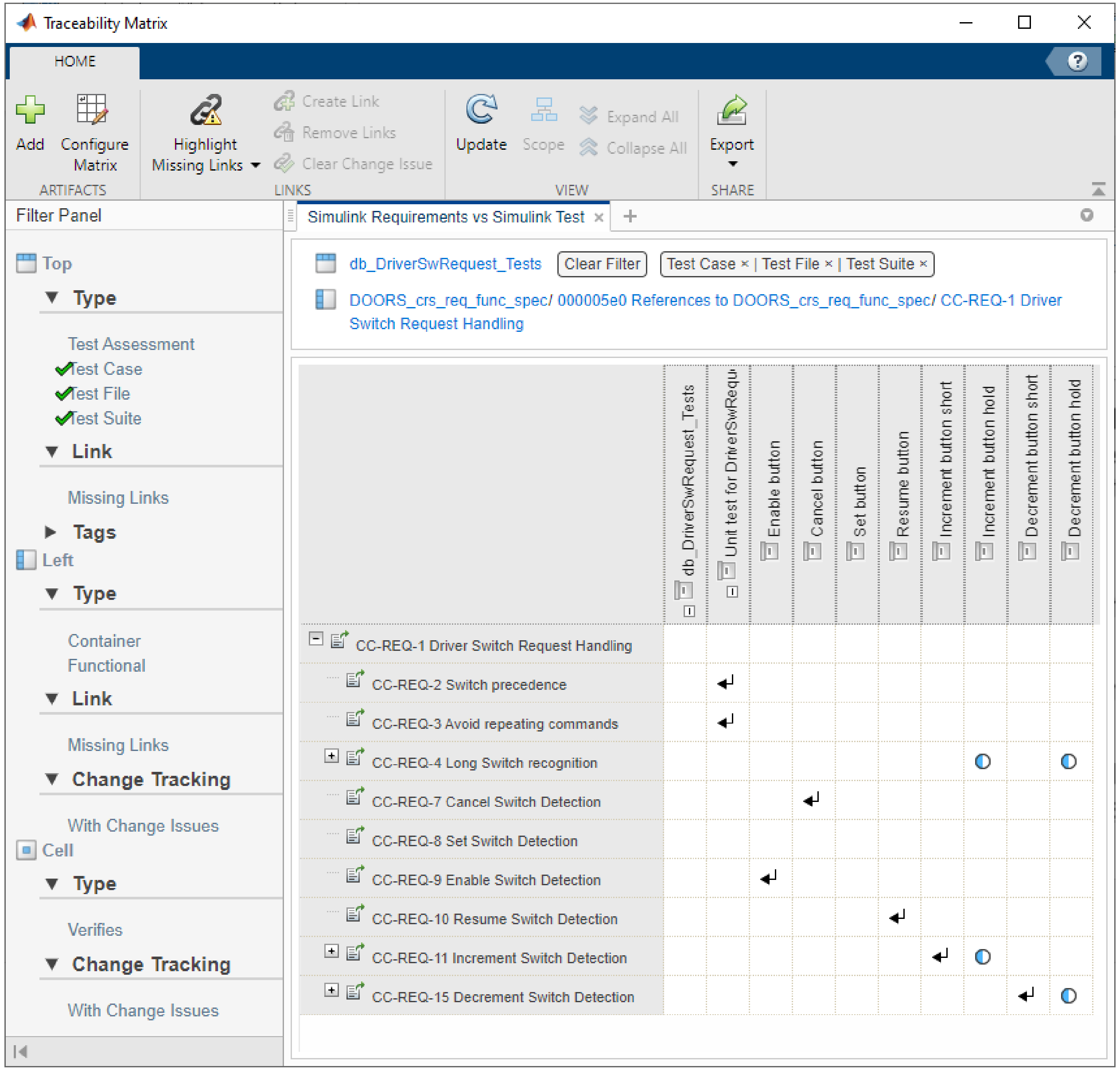
- 所有的需求都被满足了吗?
- 每一个需求都通过测试了吗?
- 需求变更会影响哪些交付物?



一起处理需求，架构和设计

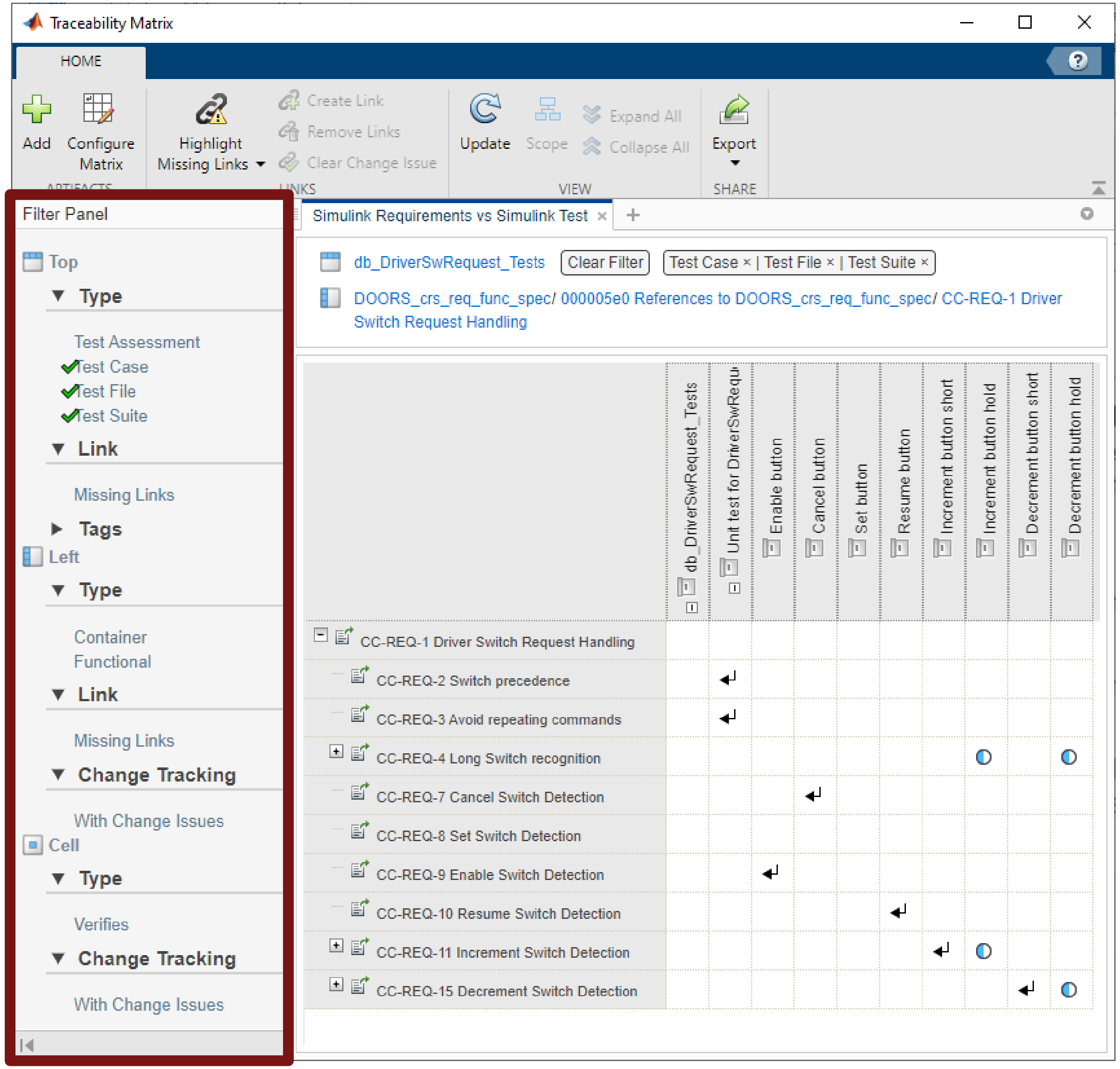


使用Traceability Matrix评审和分析追溯性

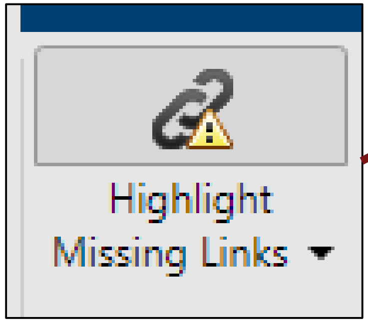


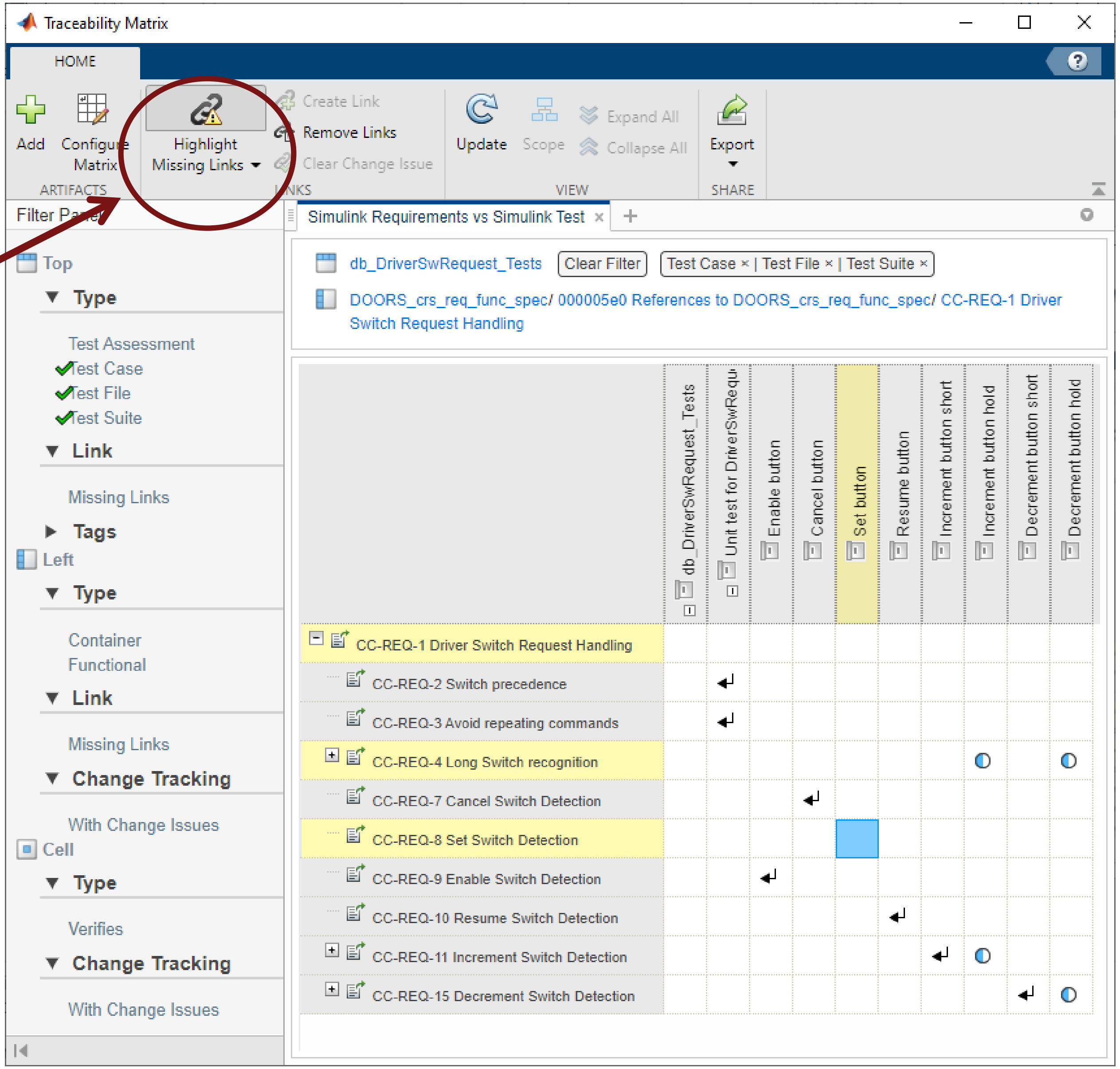
使用Traceability Matrix评审和分析追溯性

- **筛选以聚焦视图**

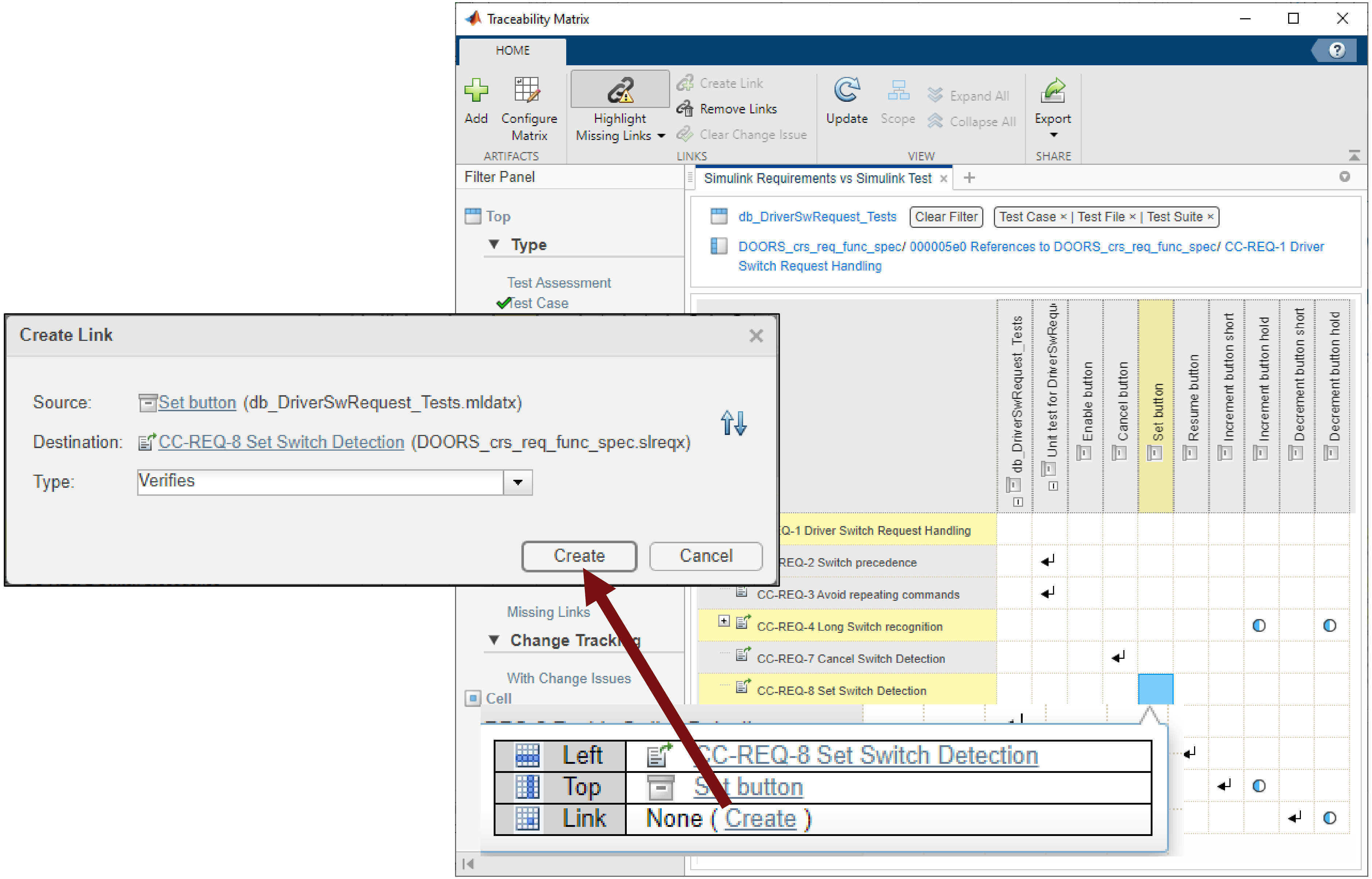


使用Traceability Matrix评审和分析追溯性



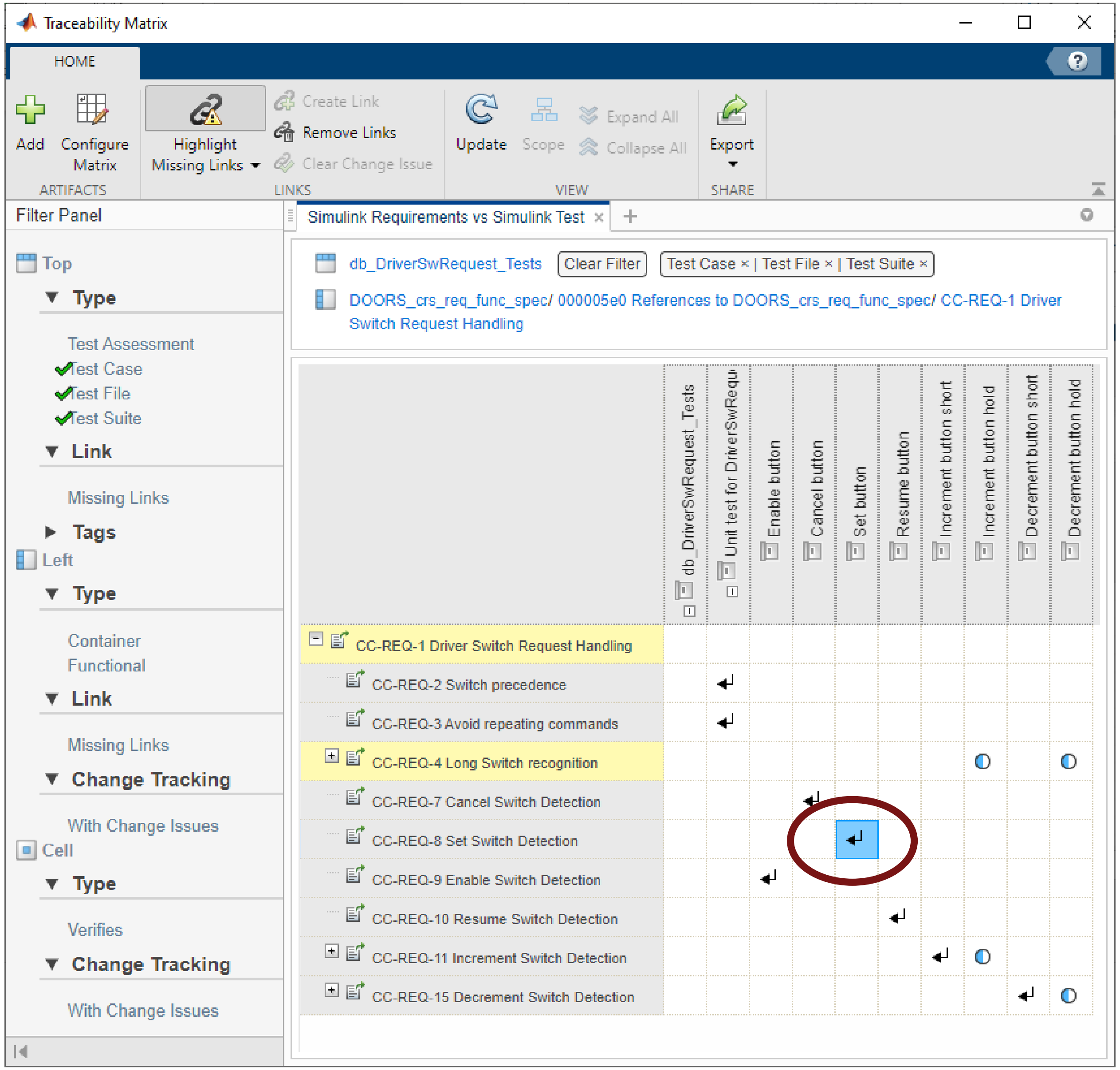


使用Traceability Matrix评审和分析追溯性



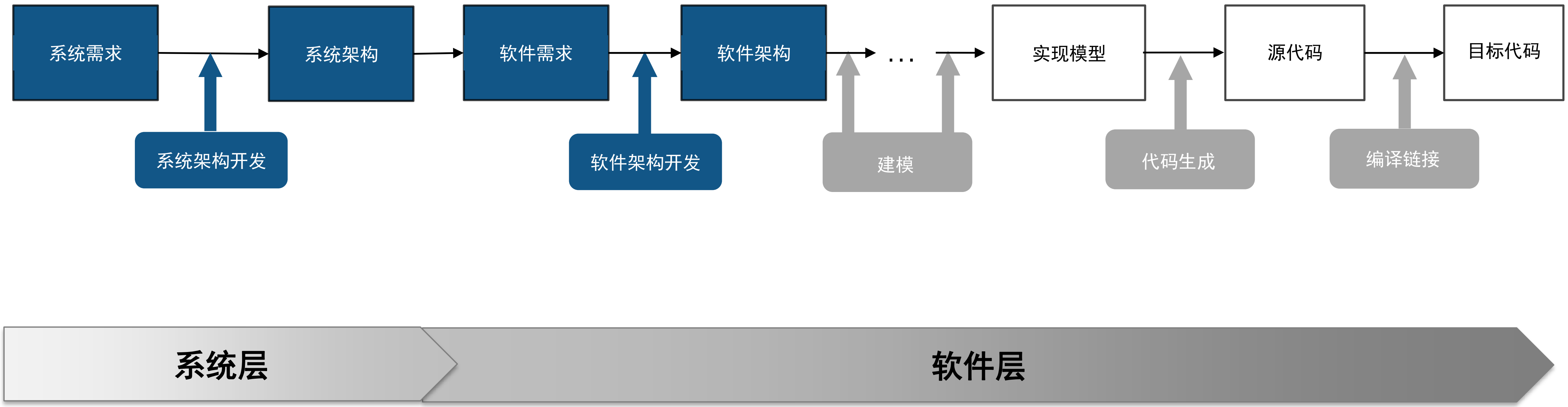
- **创建链接以弥补追溯性缺失**

使用Traceability Matrix评审和分析追溯性



- **创建链接以弥补追溯性缺失**

系统与软件架构


系统需求
系统架构
软件需求
软件架构
…
实现模型
源代码
目标代码
系统架构开发
软件架构开发
建模
代码生成
编译链接
系统层
软件层

为什么架构很重要？

尽早编写和验证需求

分摊安全目标

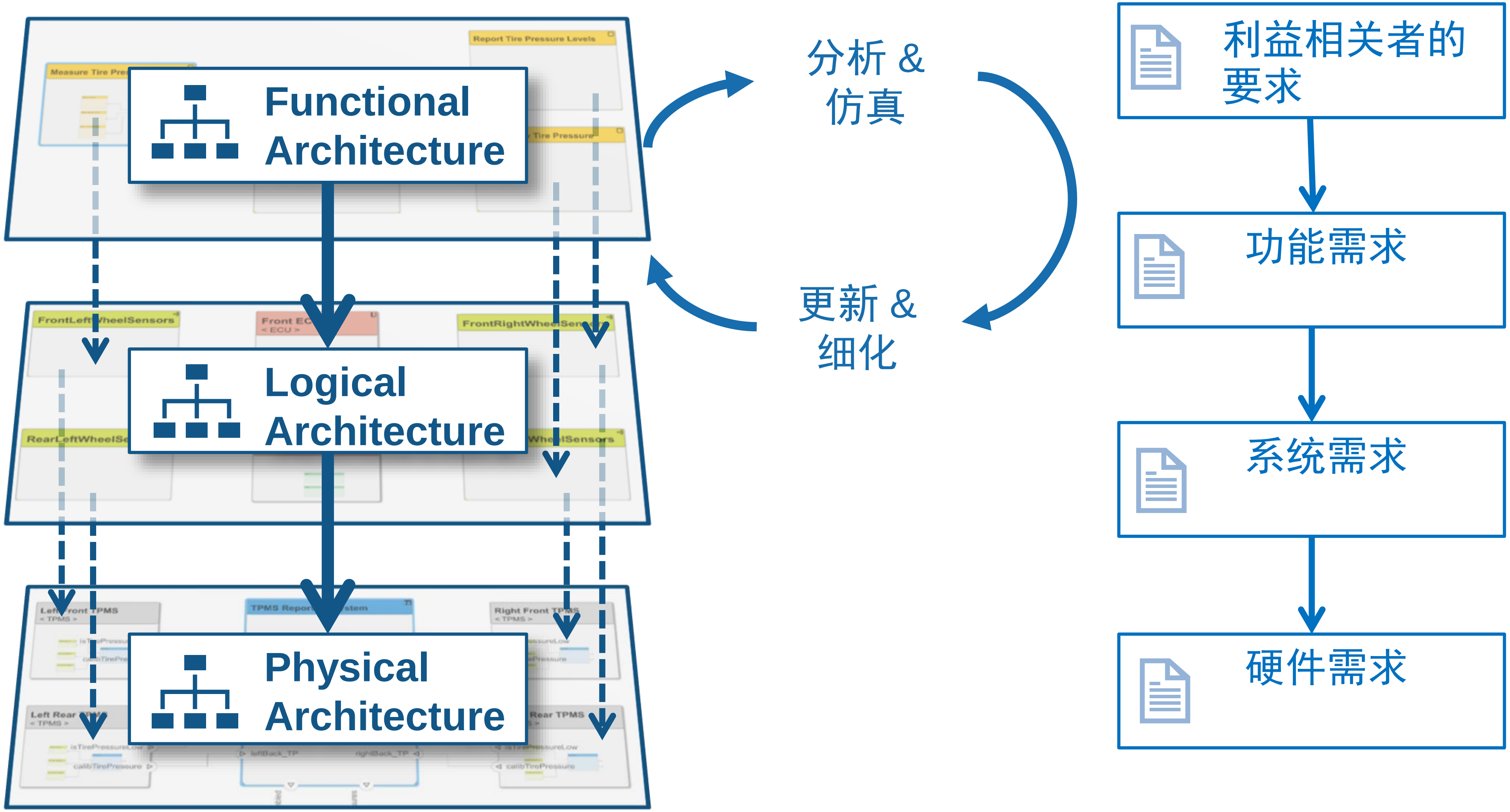
协调软件和硬件接口

无歧义地分解需求

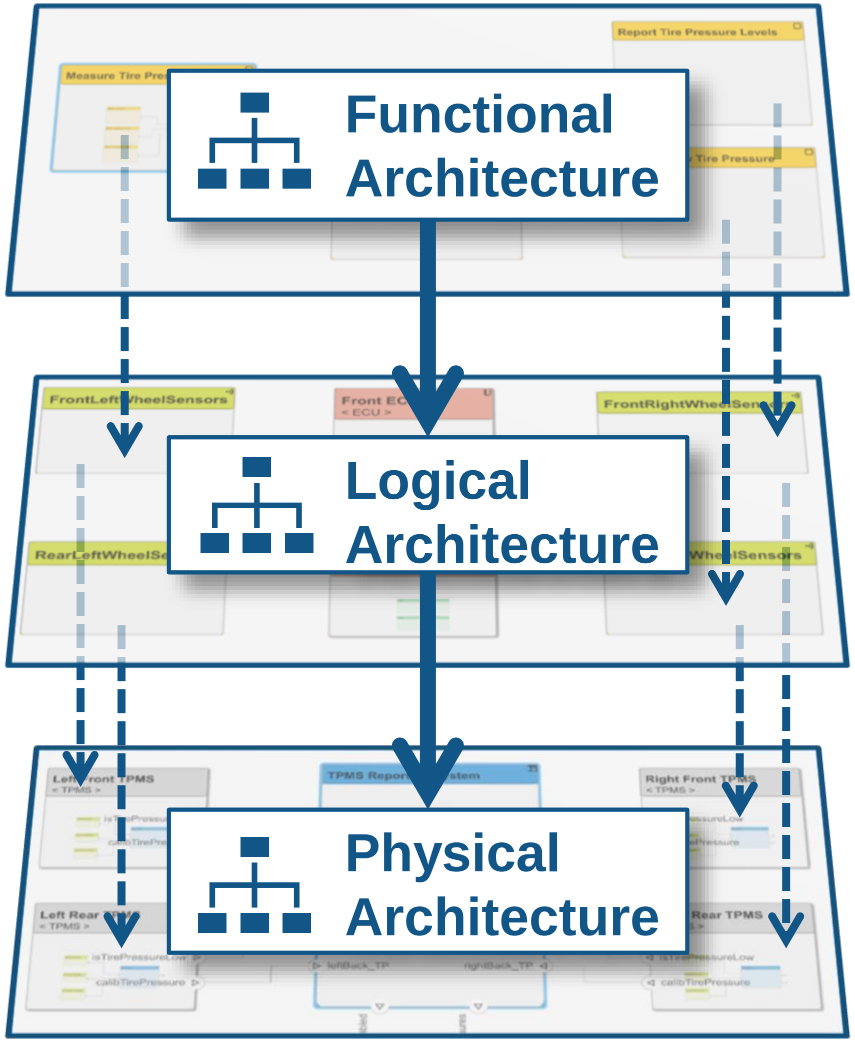
确定分配给软件或硬件

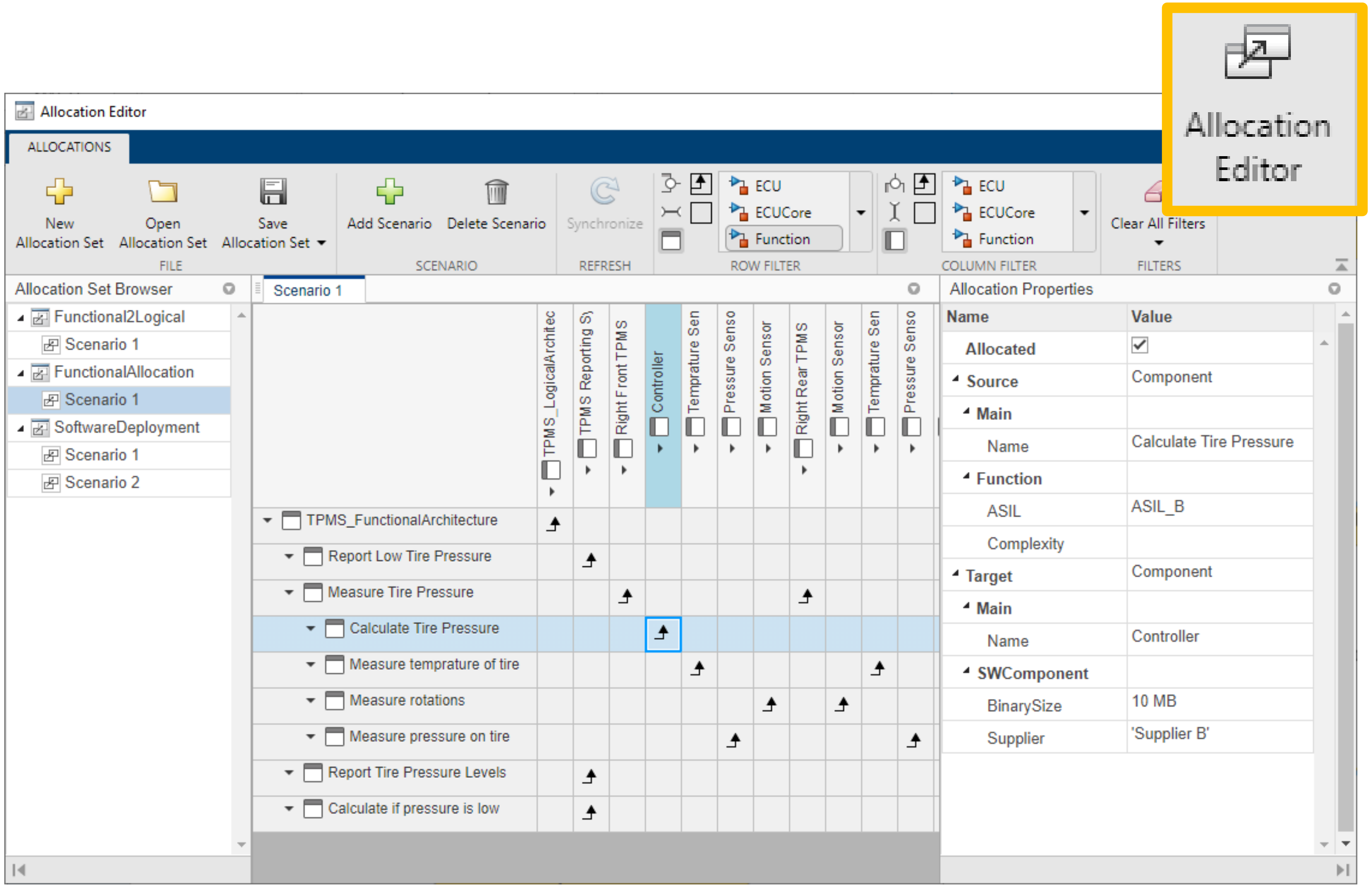
链接到行为模型

使用多种架构模型来描述系统以组织并检查需求实现的完整性。

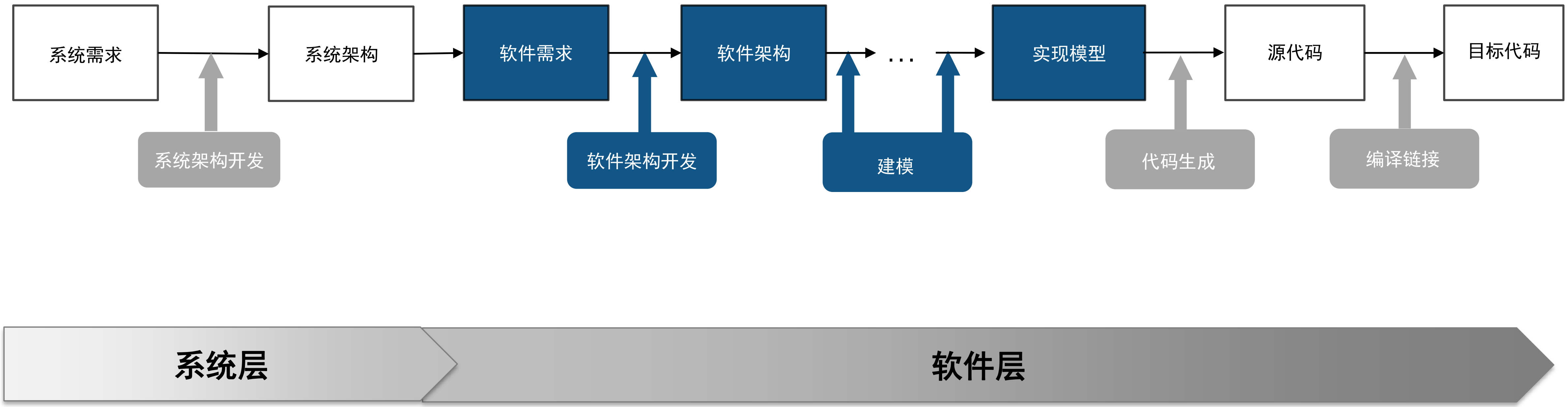


将元素从一个模型分配到另一个模型以创建可追溯和可分析的链接

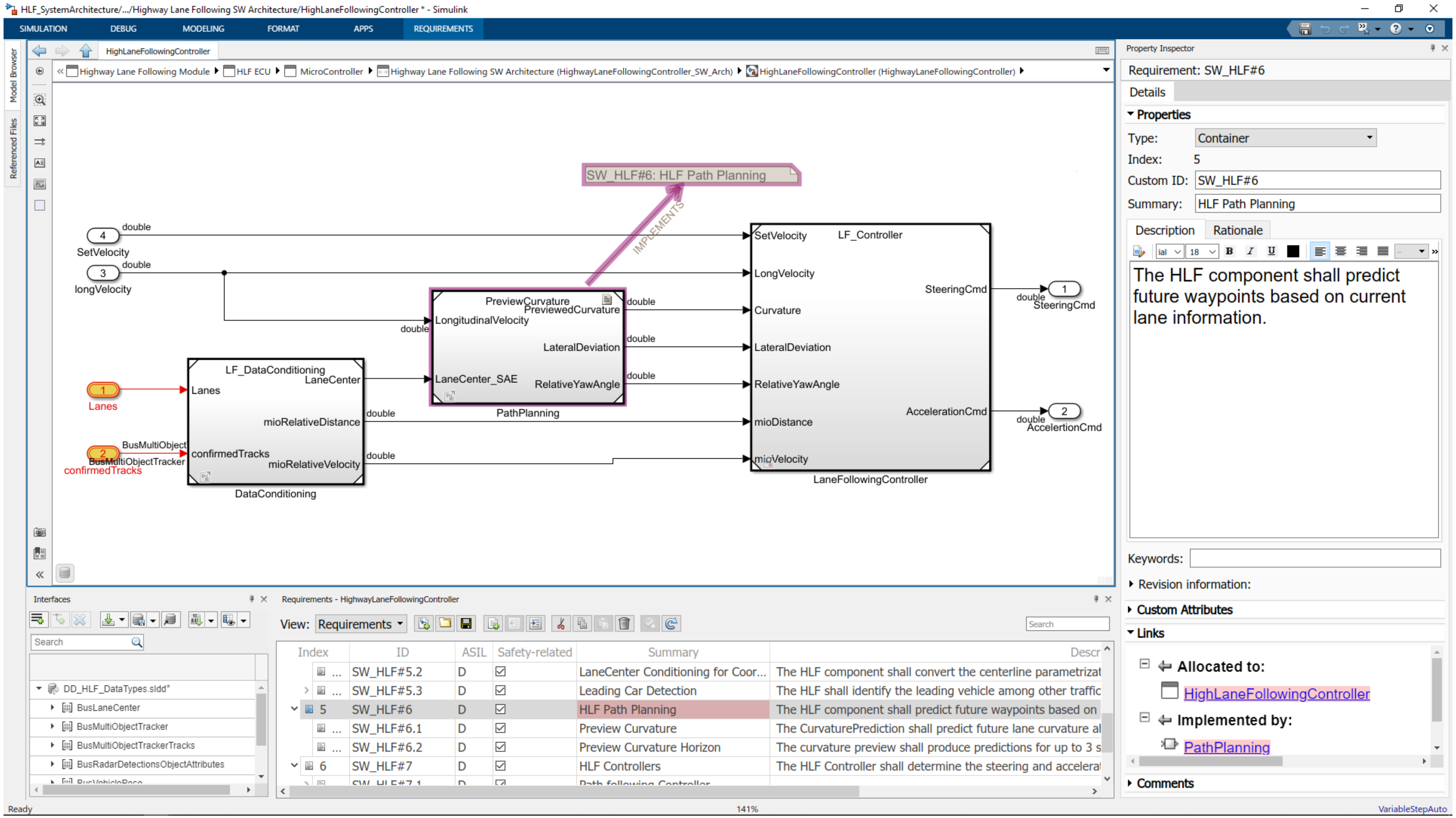




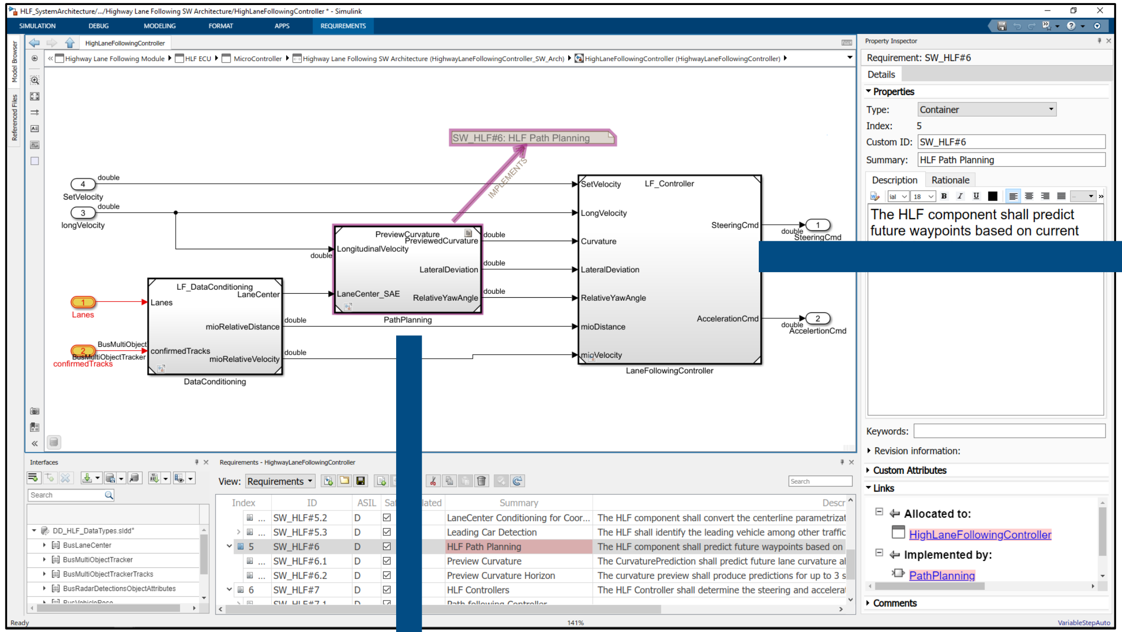
适用于认证的参考工作流程 – 详细设计

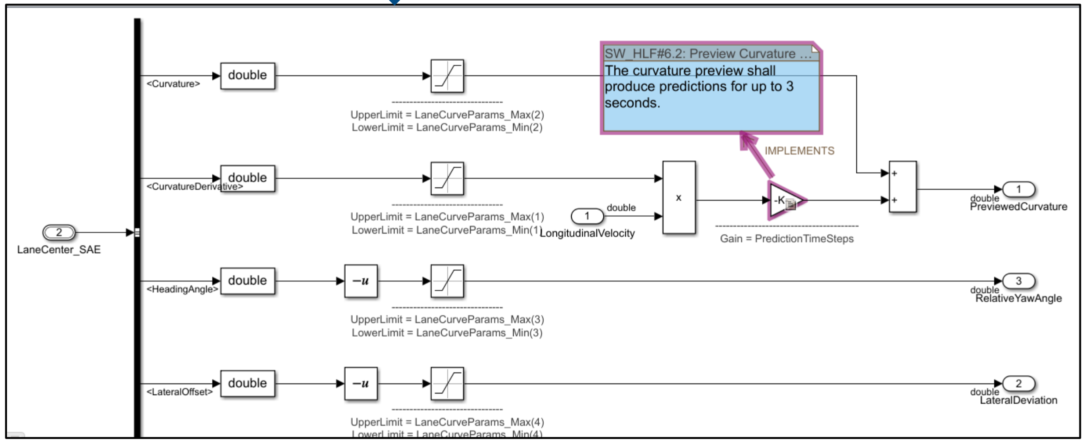


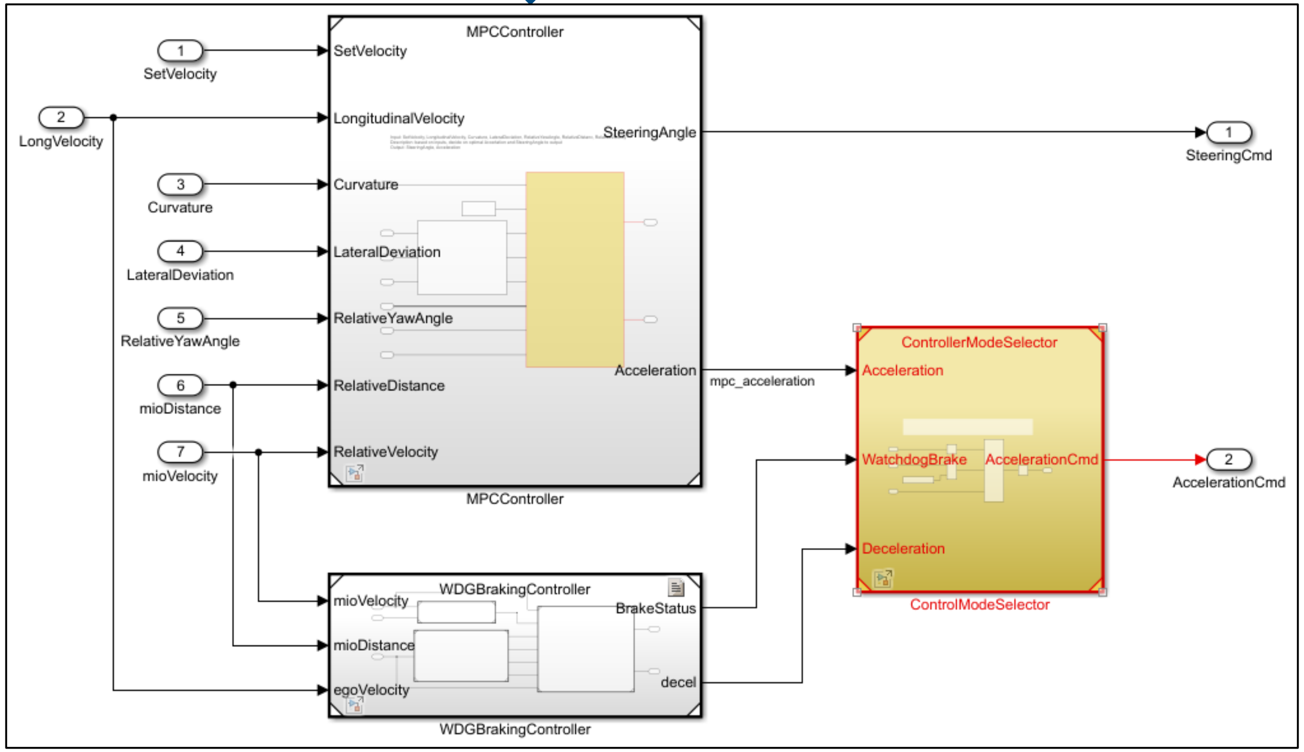
适用于认证的参考工作流程 – 详细设计



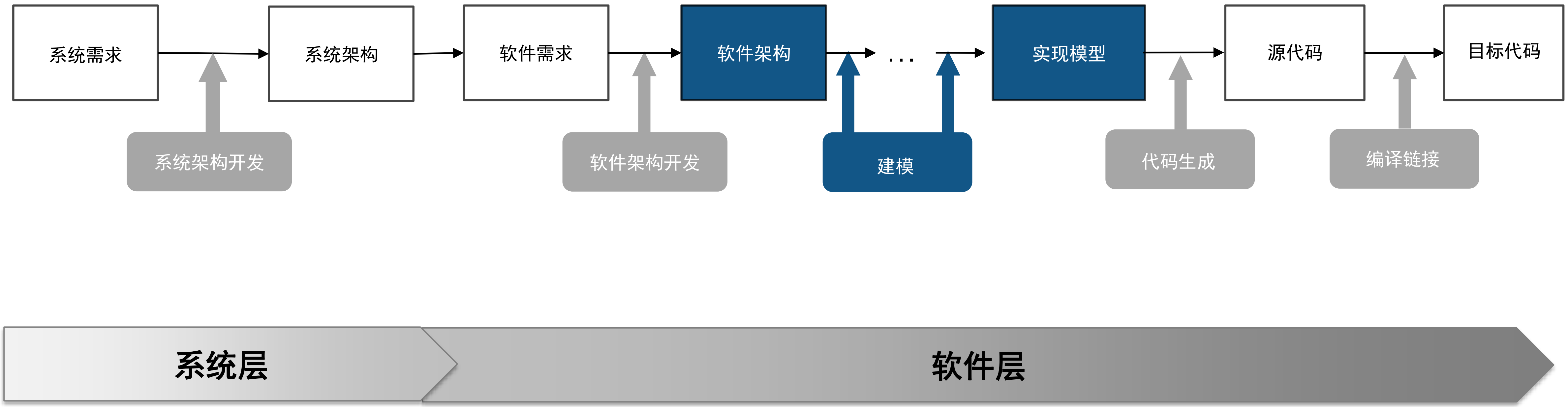
适用于认证的参考工作流程 – 详细设计



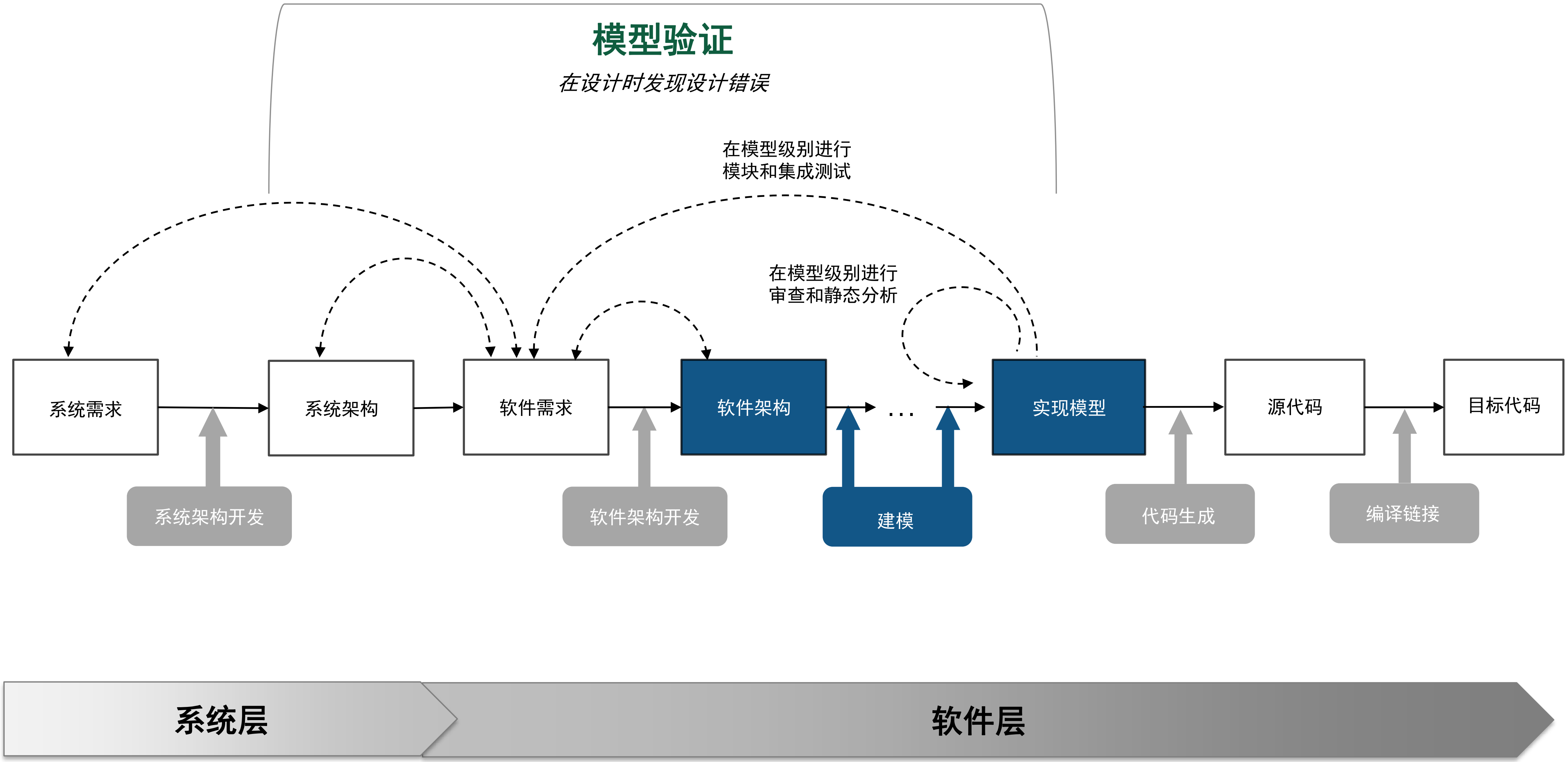




适用于认证的参考工作流程 – 详细设计



适用于认证的参考工作流程 – 模型验证

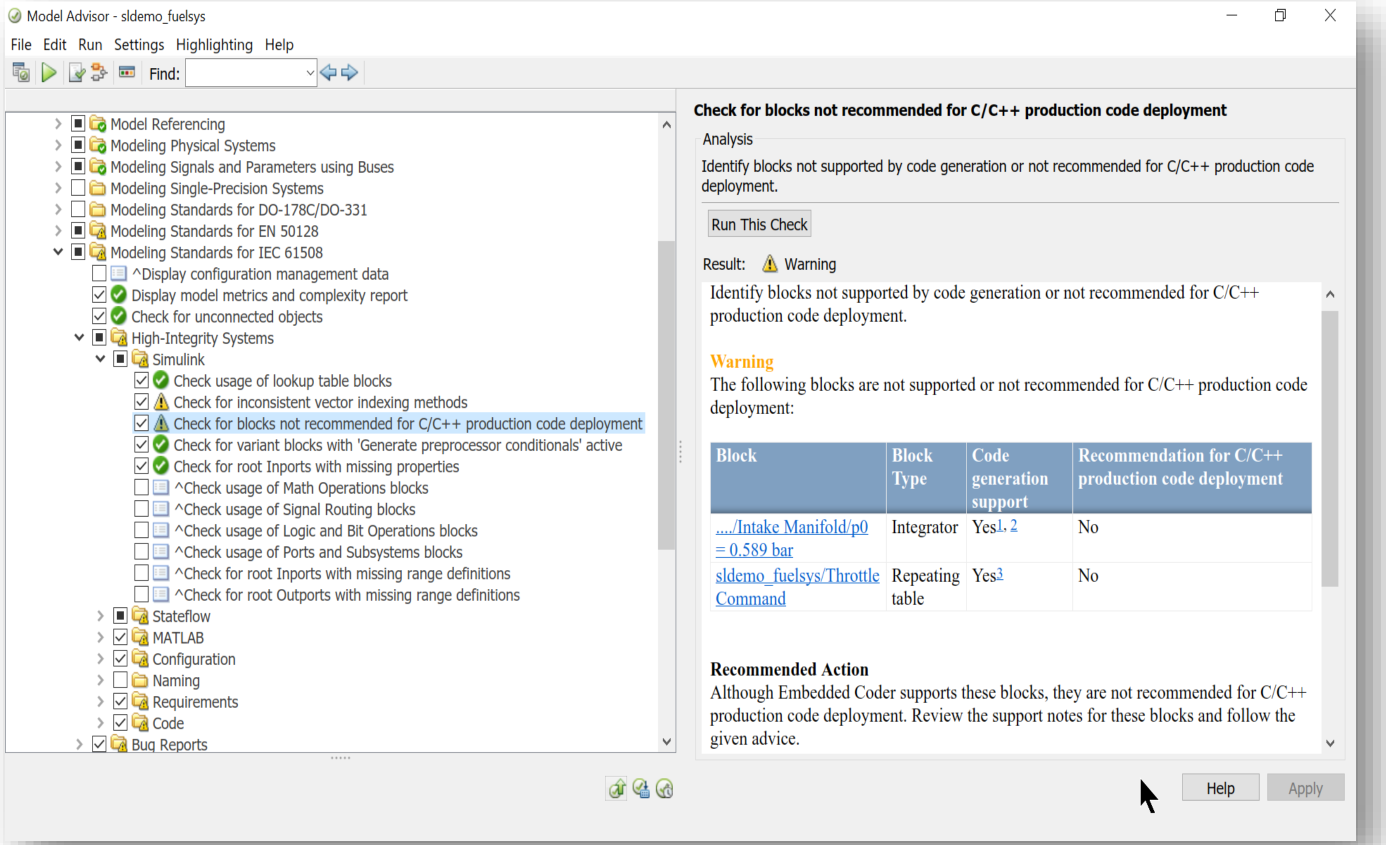


验证模型是否符合规范和标准

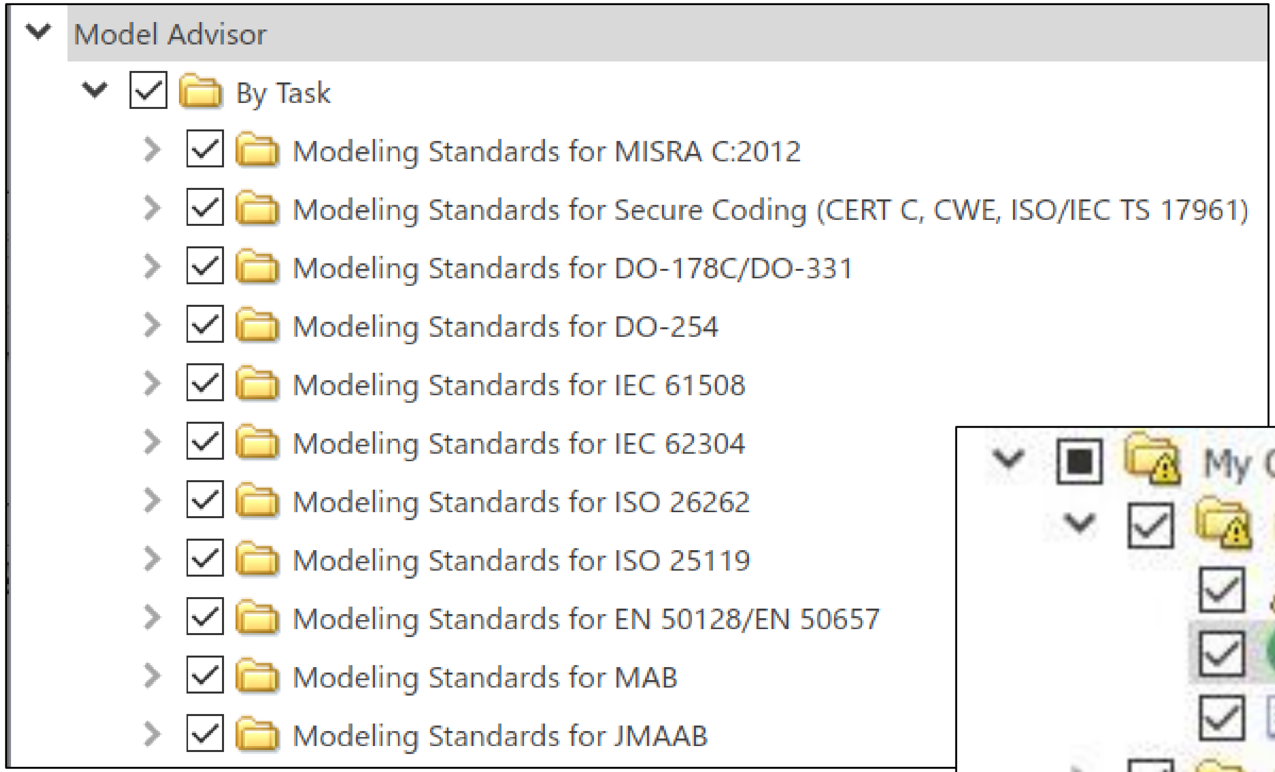
静态分析检查:

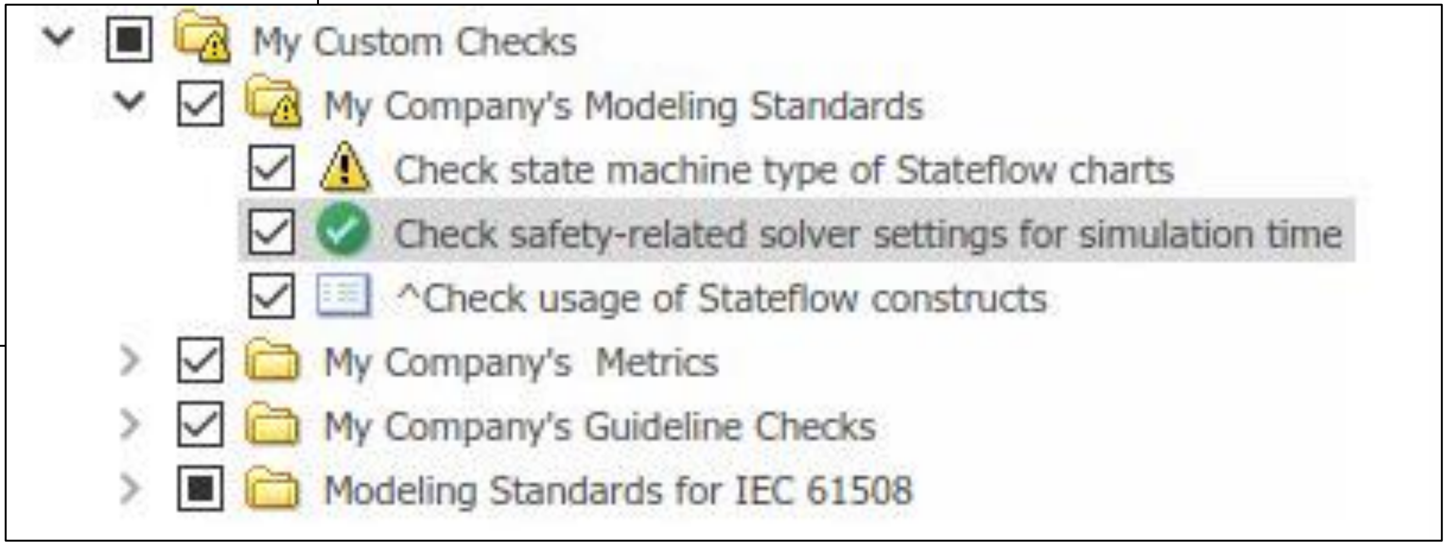
- 可读性和语义
- 性能与效率
- 复用性
- 其他更多……

Model Advisor Analysis

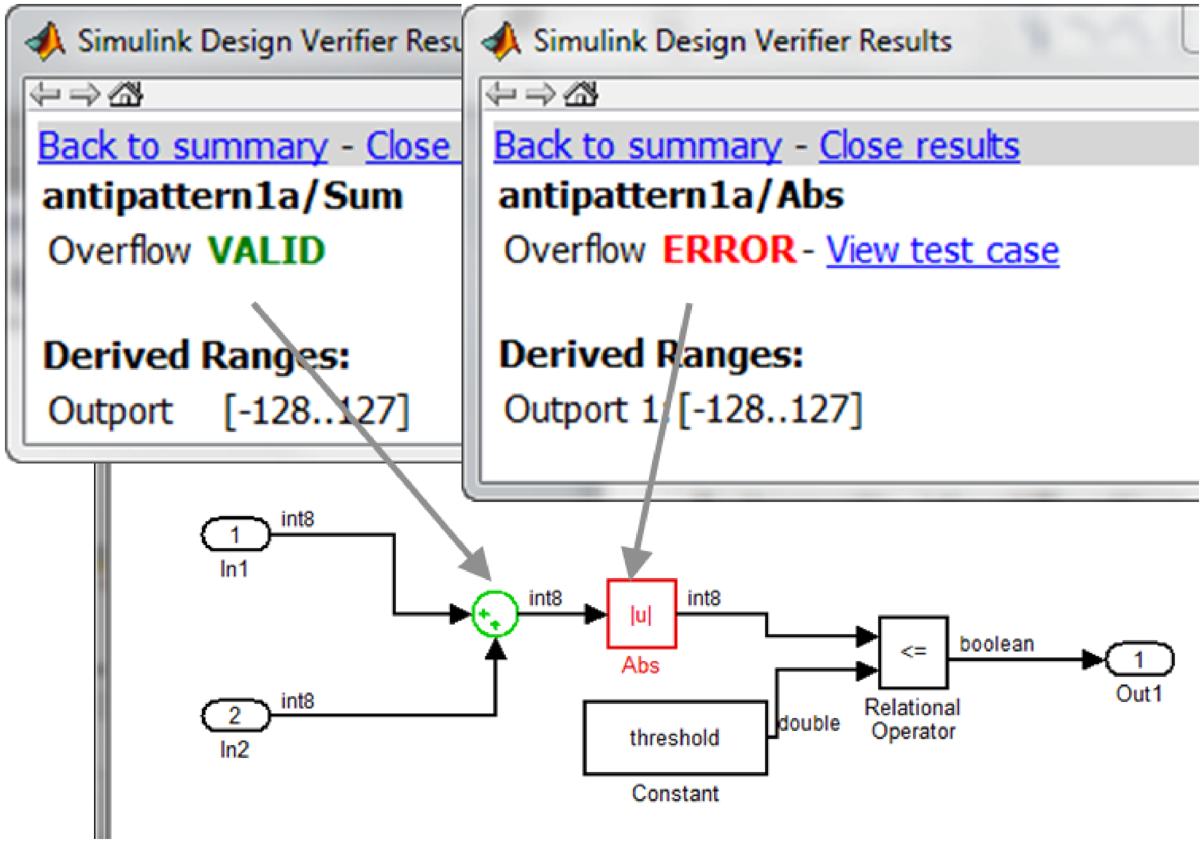


内置的行业标准与规范检查





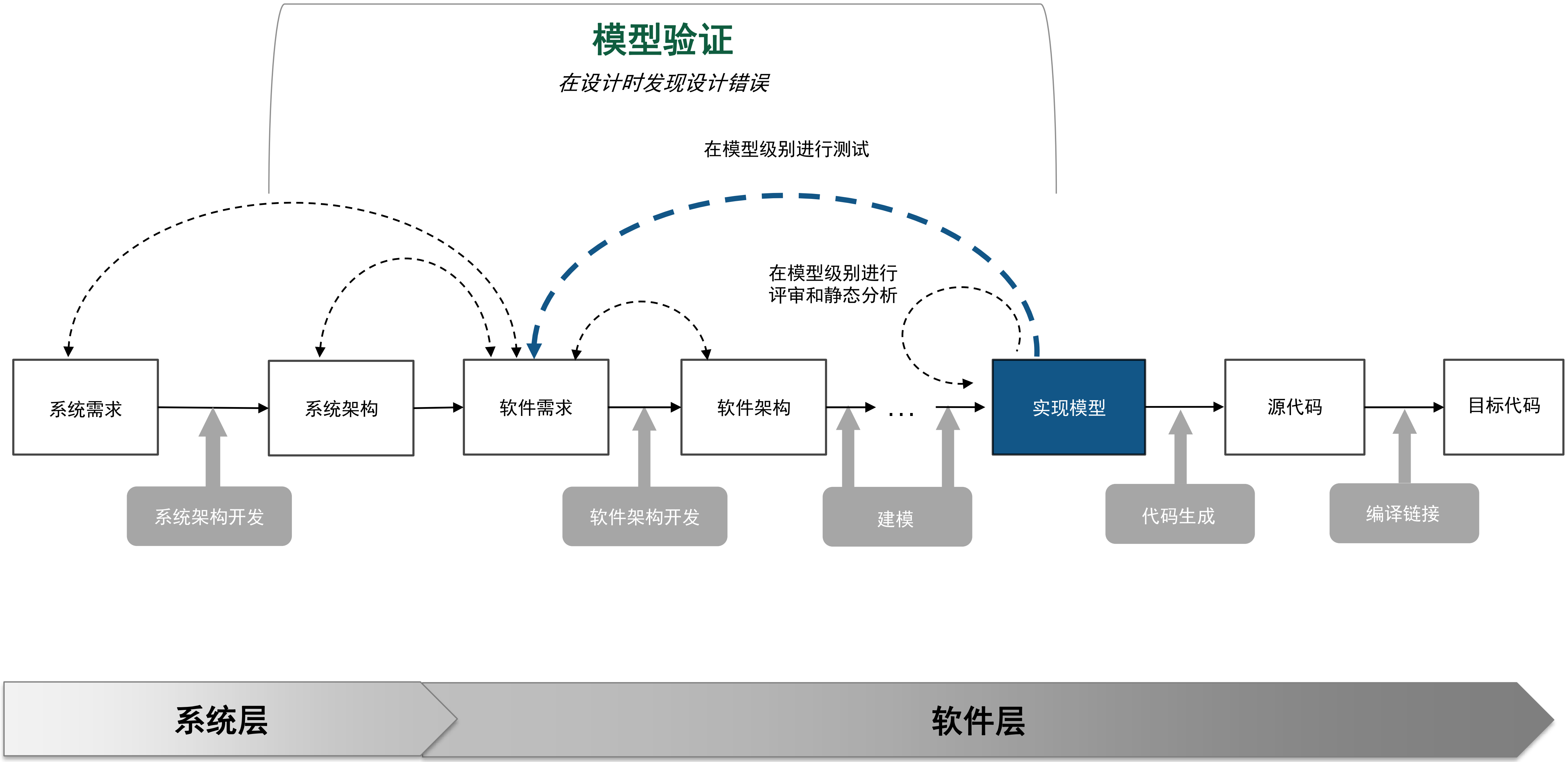
使用形式化方法检测设计错误



- 查找设计错误
  - 整数溢出
  - 死逻辑
  - 除零
  - 数组越界
  - 数值越界

- 自动生成反例以复现错误

适用于认证的参考工作流程



使用Simulink Test 进行系统化的功能测试

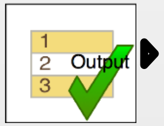
测试用例

- 通过SIL，PIL和HIL运行模型
- 通过Parallel Computing Toolbox 和持续集成扩展测试规模

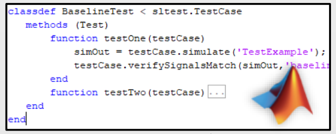
Inputs

Data file (input)

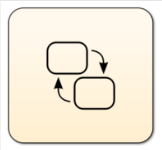
Signal Editor



Test Sequence

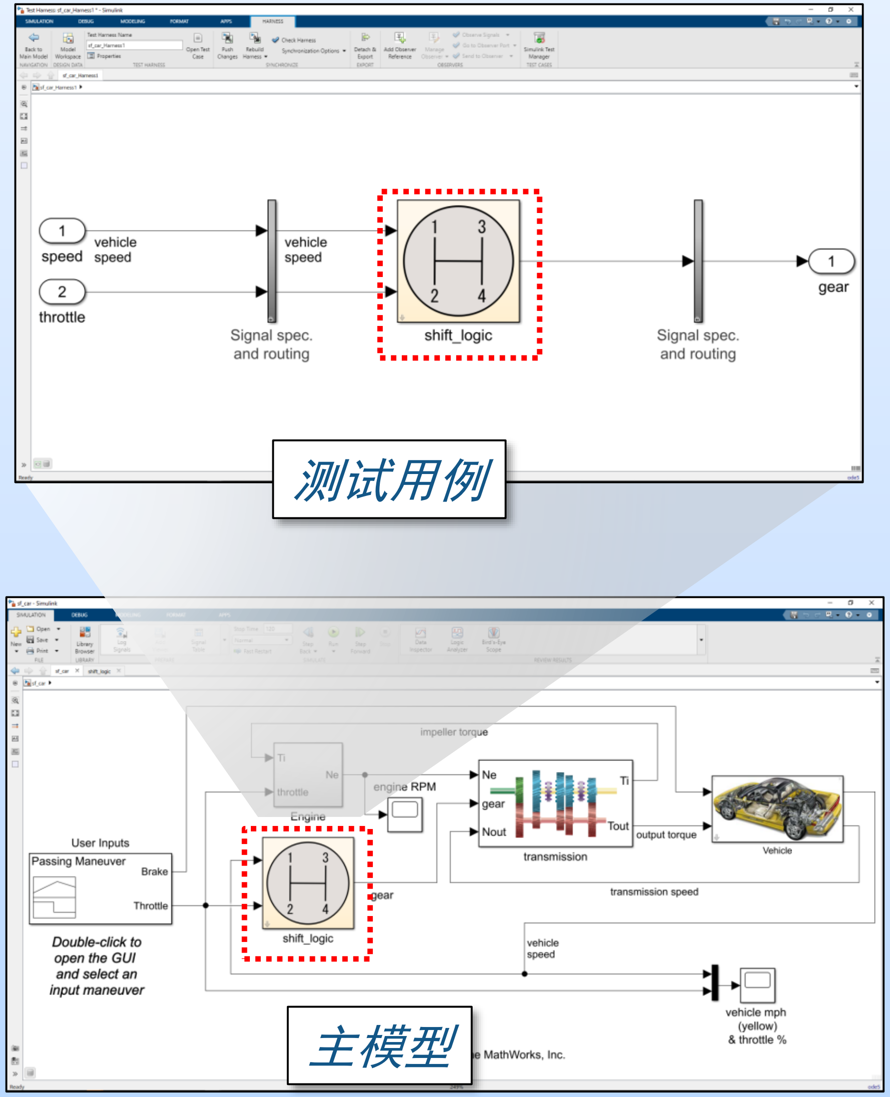


MATLAB 代码

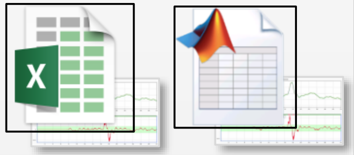


Stateflow

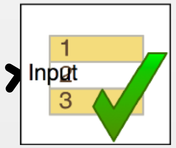
还有更多！



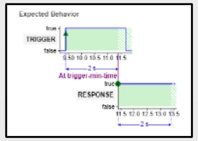
Assessments



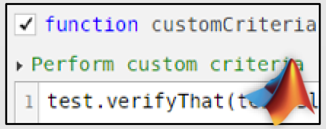
Data file baseline)



Test Assessment



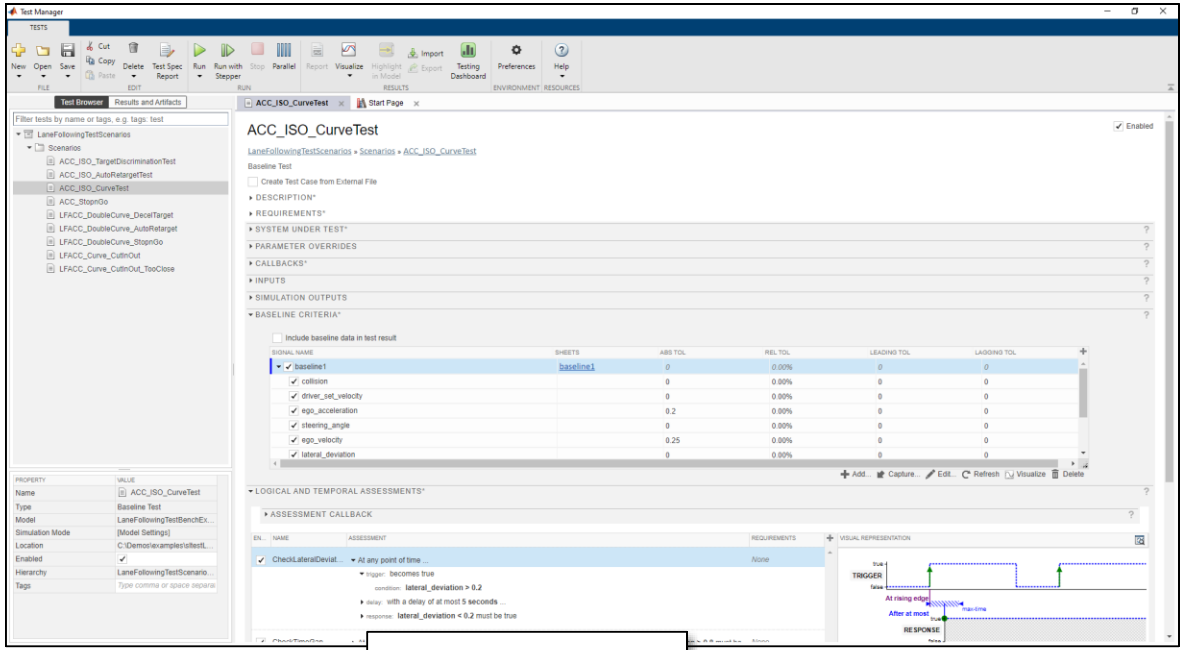
Temporal Assessment



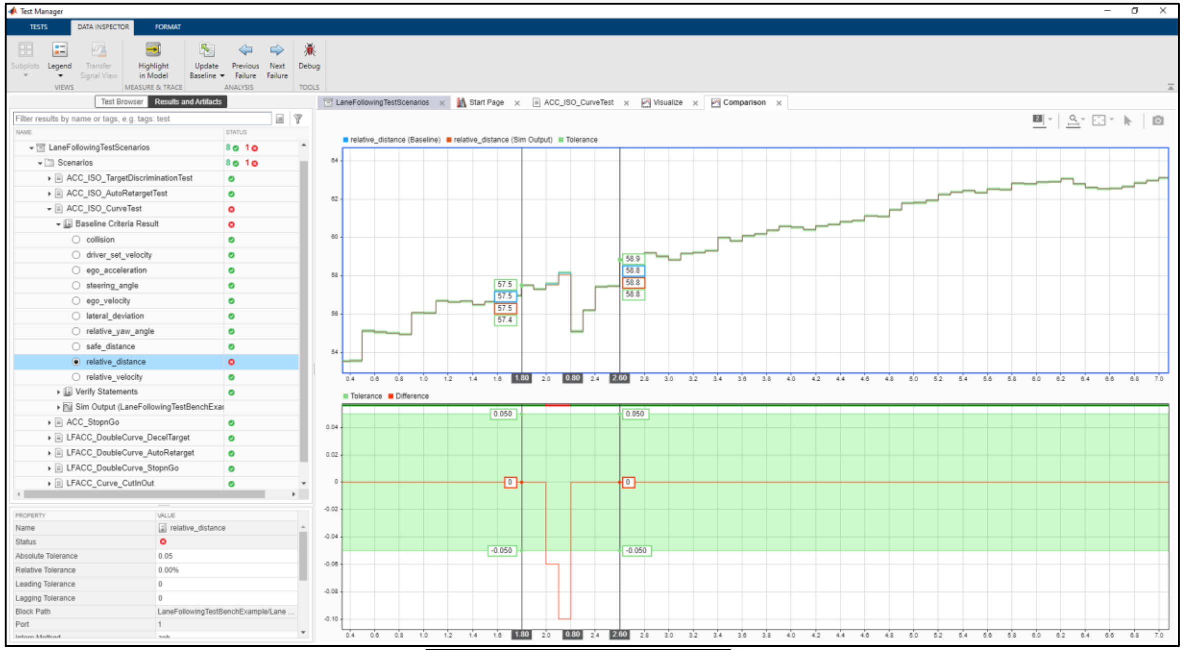
MATLAB Code

还有更多!

管理测试和测试结果



Test Browser

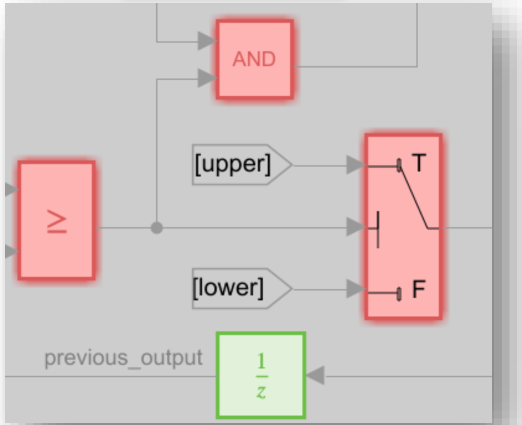


Test Results

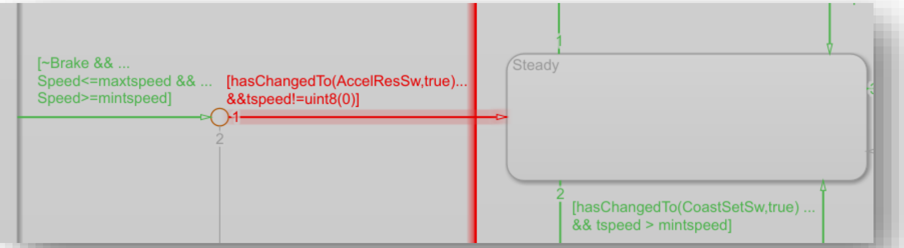
测试覆盖度分析

- 找出缺失测试
- 缺失的需求
- 意外的功能
- 设计错误

Simulink



Stateflow



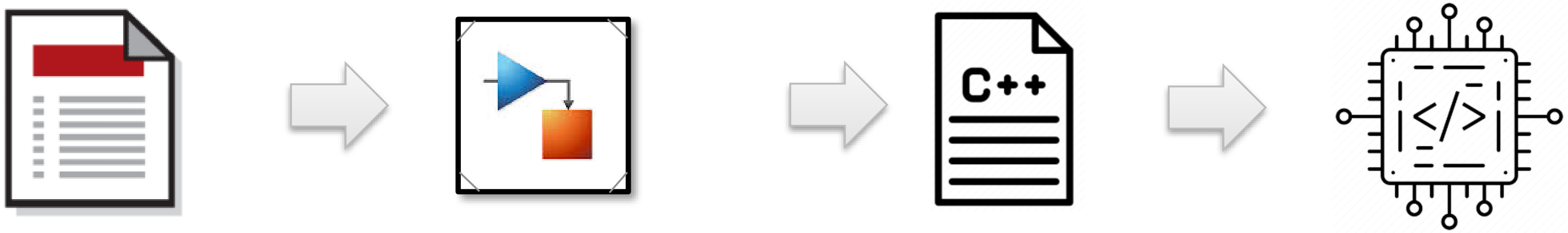
Code

Coverage Reports

Summary

Model Hierarchy/Complexity		Test 1 Decision	Condition	MCDC	Execution	Relational Boundary	Saturation on integer overflow
1. sldemo_fuelsys	80	34%	34%	7%	90%	10%	50%
2.... Engine Gas Dynamics	13	71%	NA	NA	100%	50%	50%
3....... Mixing & Combustion	3	67%	NA	NA	100%	NA	50%
4.......... EGO Sensor	2	100%	NA	NA	NA	NA	NA
5.......... System Lag		NA	NA	NA	100%	NA	NA
6....... Throttle & Manifold	10	73%	NA	NA	100%	50%	50%
7.......... Intake Manifold	2	100%	NA	NA	100%	NA	50%
8............. MATLAB Function	2	100%	NA	NA	NA	NA	NA
9.......... Throttle	6	83%	NA	NA	100%	100%	50%

挑战：管理满足认证标准所需的各种工作流和交付物



需求
- 高层次需求
- 需求模型
- 评审核清
- 软件硬件/硬件层
- 需求追溯
- …

设计
- 建模标准规范
- 设计错误检测
- 测试用例
- 测试结果
- …

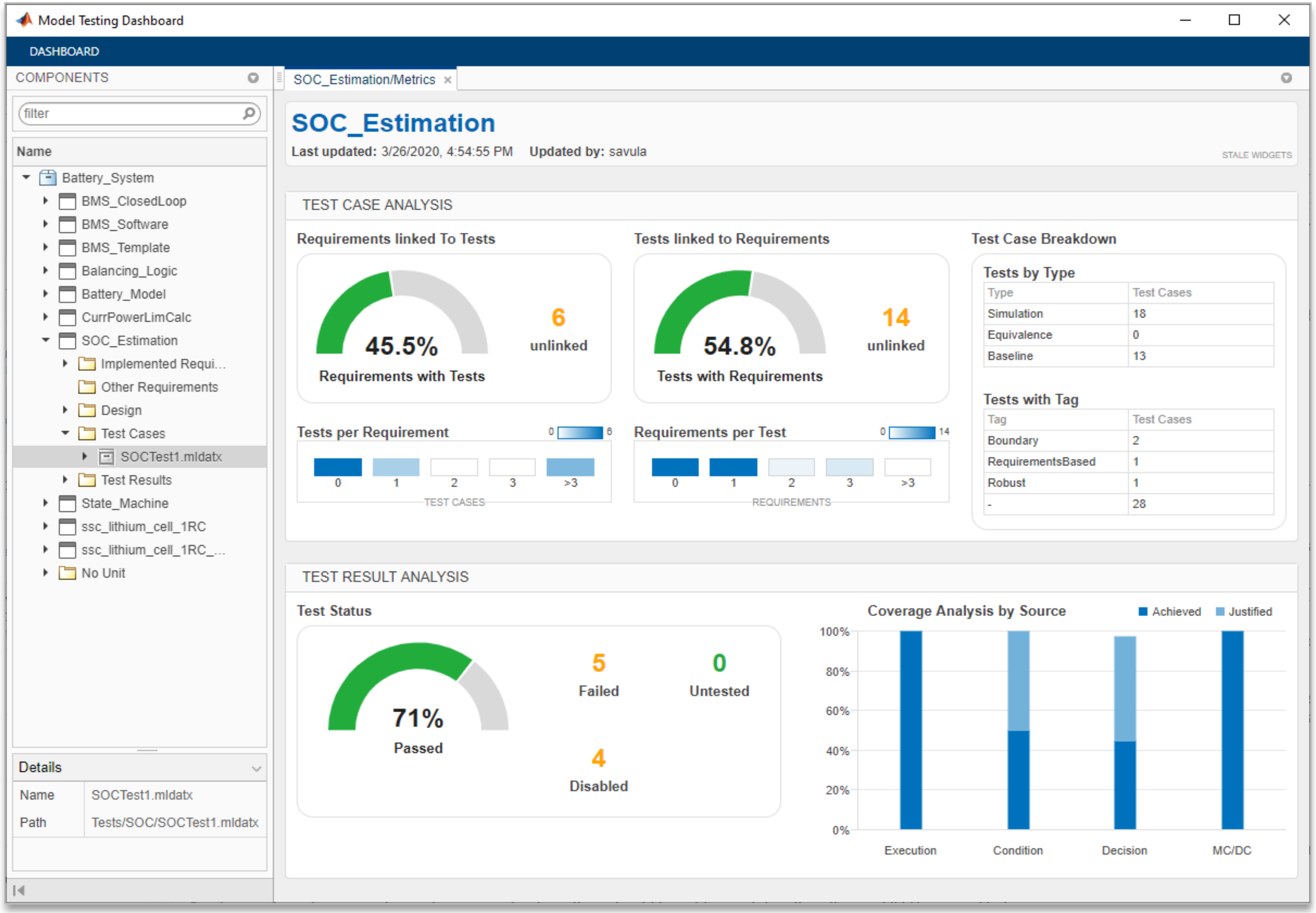
代码
- 代码标准规范
- Code Inspection
- 代码验证
- 代码指标
- 测试结果
- …

目标芯片
- 性能指标
- RAM/ROM 指标
- HW/SW 集成测试用例
- 测试结果
- …

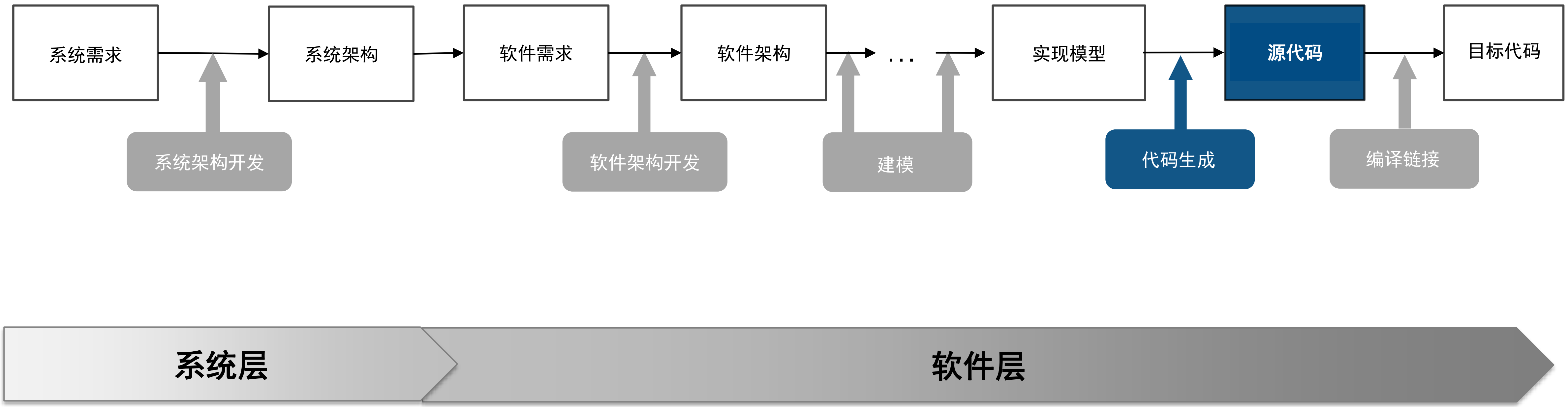
Model Testing Dashboard: 管理基于需求的测试的进度，完整性和质量

R2020b

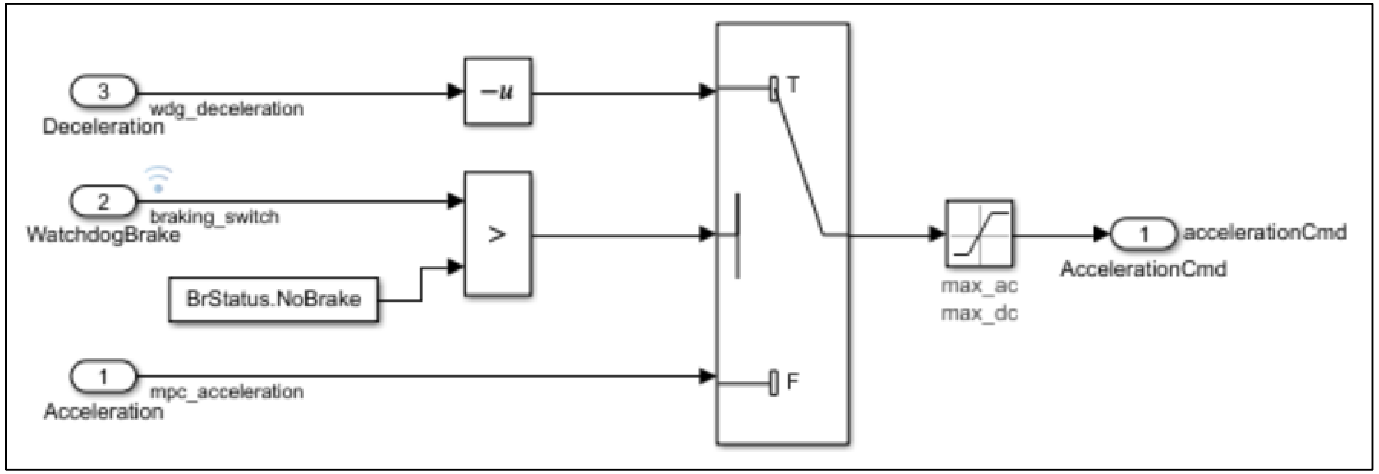
- 汇总测试数据和状态的中央视图
- 找出差距并更快地响应需求变更
- 查看来自持续集成系统的结果



代码生成

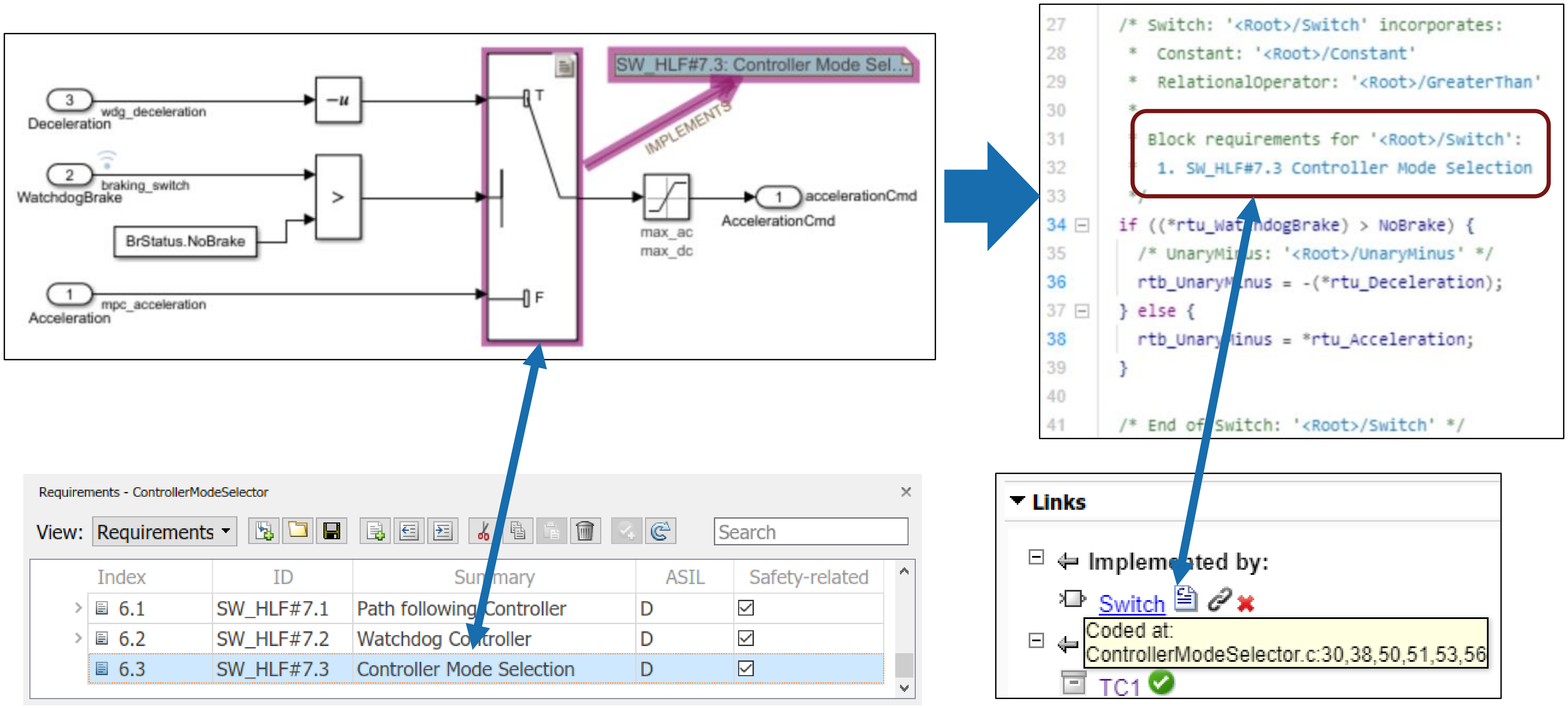

系统需求
系统架构
软件需求
软件架构
…
实现模型
源代码
目标代码
系统架构开发
软件架构开发
建模
代码生成
编译链接
系统层
软件层

自动生成与您在Simulink中创建的模型行为相同的量产级代码

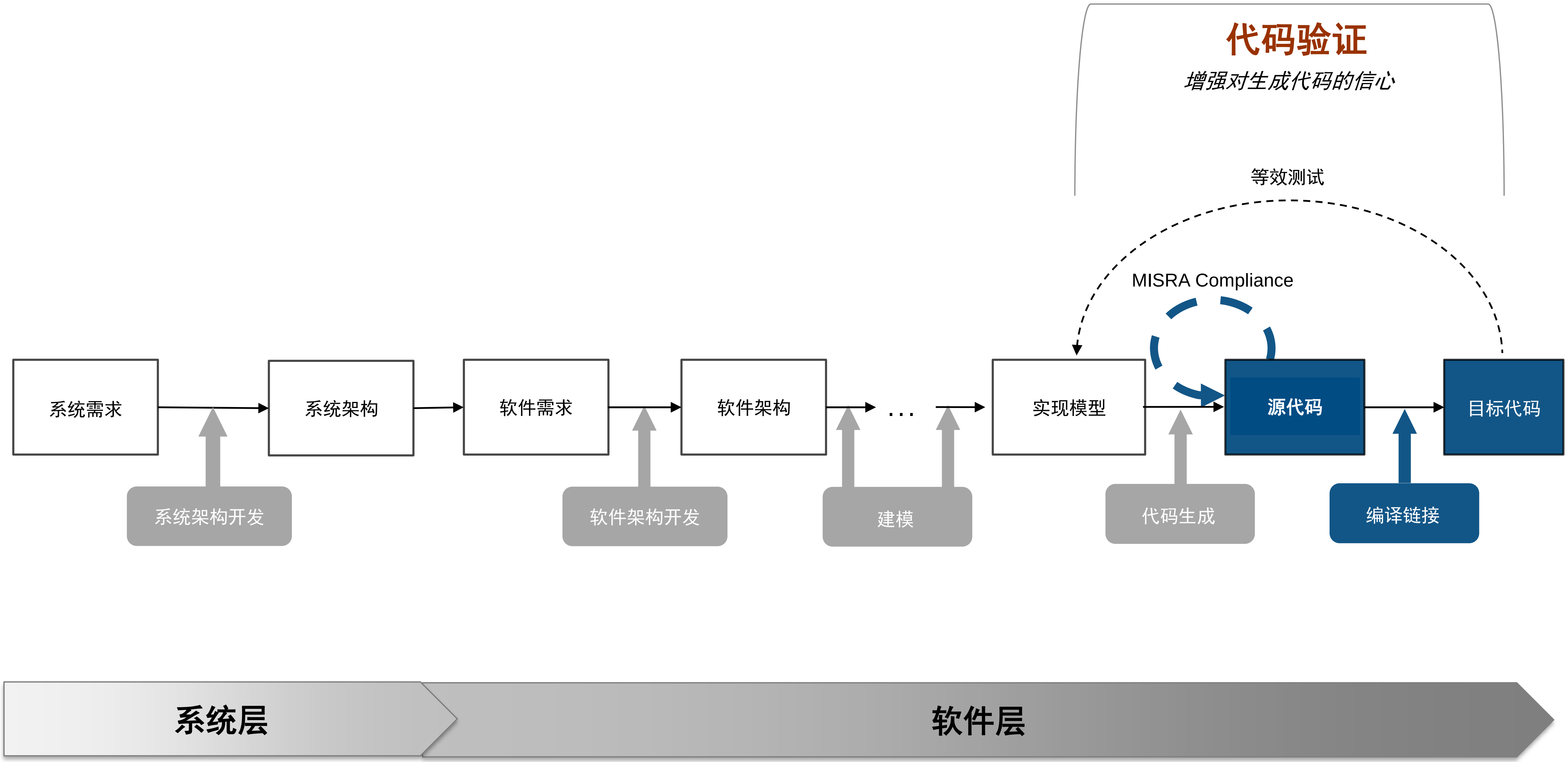



```c
/* Switch: '<Root>/Switch' incorporates:
 *  Constant: '<Root>/Constant'
 *  RelationalOperator: '<Root>/GreaterThan'
 *
 * Block requirements for '<Root>/Switch':
 *  1. SW_HLF#7.3 Controller Mode Selection
 */
if ((*rtu_WatchdogBrake) > NoBrake) {
  /* UnaryMinus: '<Root>/UnaryMinus' */
  rtb_UnaryMinus = -(*rtu_Deceleration);
} else {
  rtb_UnaryMinus = *rtu_Acceleration;
}

/* End of Switch: '<Root>/Switch' */
```

生成代码中包含的需求可追溯性

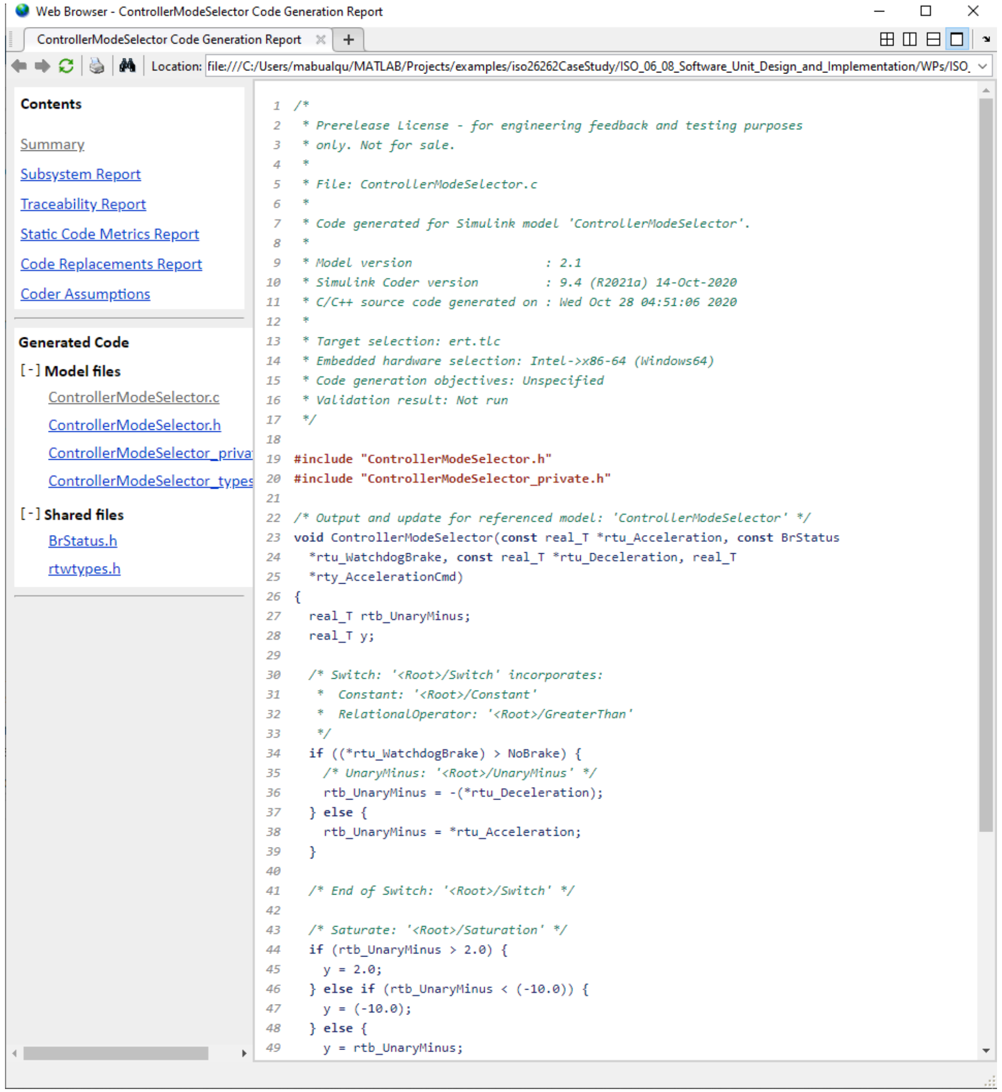


适用于认证的参考工作流程 – 代码验证



适用于认证的参考工作流程 – Code Verification

Static Code Verification for MISRA Compliance



Web Browser - ControllerModeSelector Code Generation Report

ControllerModeSelector Code Generation Report

Location: file:///C:/Users/mabualqu/MATLAB/Projects/examples/iso26262CaseStudy/ISO_06_08_Software_Unit_Design_and_Implementation/WPs/ISO_

Contents

- Summary
- Subsystem Report
- Traceability Report
- Static Code Metrics Report
- Code Replacements Report
- Coder Assumptions

Generated Code

[-] **Model files**

- ControllerModeSelector.c
- ControllerModeSelector.h
- ControllerModeSelector_priva
- ControllerModeSelector_types

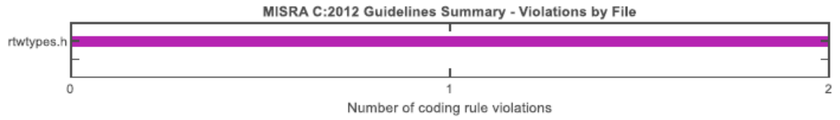
[-] **Shared files**

- BrStatus.h
- rtwtypes.h

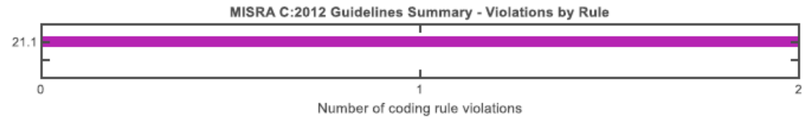
```c
/*
 * Prerelease License - for engineering feedback and testing purposes
 * only. Not for sale.
 *
 * File: ControllerModeSelector.c
 *
 * Code generated for Simulink model 'ControllerModeSelector'.
 *
 * Model version                  : 2.1
 * Simulink Coder version         : 9.4 (R2021a) 14-Oct-2020
 * C/C++ source code generated on : Wed Oct 28 04:51:06 2020
 *
 * Target selection: ert.tlc
 * Embedded hardware selection: Intel->x86-64 (Windows64)
 * Code generation objectives: Unspecified
 * Validation result: Not run
 */

#include "ControllerModeSelector.h"
#include "ControllerModeSelector_private.h"

/* Output and update for referenced model: 'ControllerModeSelector' */
void ControllerModeSelector(const real_T *rtu_Acceleration, const BrStatus
  *rtu_WatchdogBrake, const real_T *rtu_Deceleration, real_T
  *rty_AccelerationCmd)
{
  real_T rtb_UnaryMinus;
  real_T y;

  /* Switch: '<Root>/Switch' incorporates:
   *  Constant: '<Root>/Constant'
   *  RelationalOperator: '<Root>/GreaterThan'
   */
  if ((*rtu_WatchdogBrake) > NoBrake) {
    /* UnaryMinus: '<Root>/UnaryMinus' */
    rtb_UnaryMinus = -(*rtu_Deceleration);
  } else {
    rtb_UnaryMinus = *rtu_Acceleration;
  }

  /* End of Switch: '<Root>/Switch' */

  /* Saturate: '<Root>/Saturation' */
  if (rtb_UnaryMinus > 2.0) {
    y = 2.0;
  } else if (rtb_UnaryMinus < (-10.0)) {
    y = (-10.0);
  } else {
    y = rtb_UnaryMinus;
```

Chapter 1. MISRA C:2012 Guidelines

MISRA C:2012 Guidelines Summary - Violations by File



MISRA C:2012 Guidelines Summary - Violations by Rule



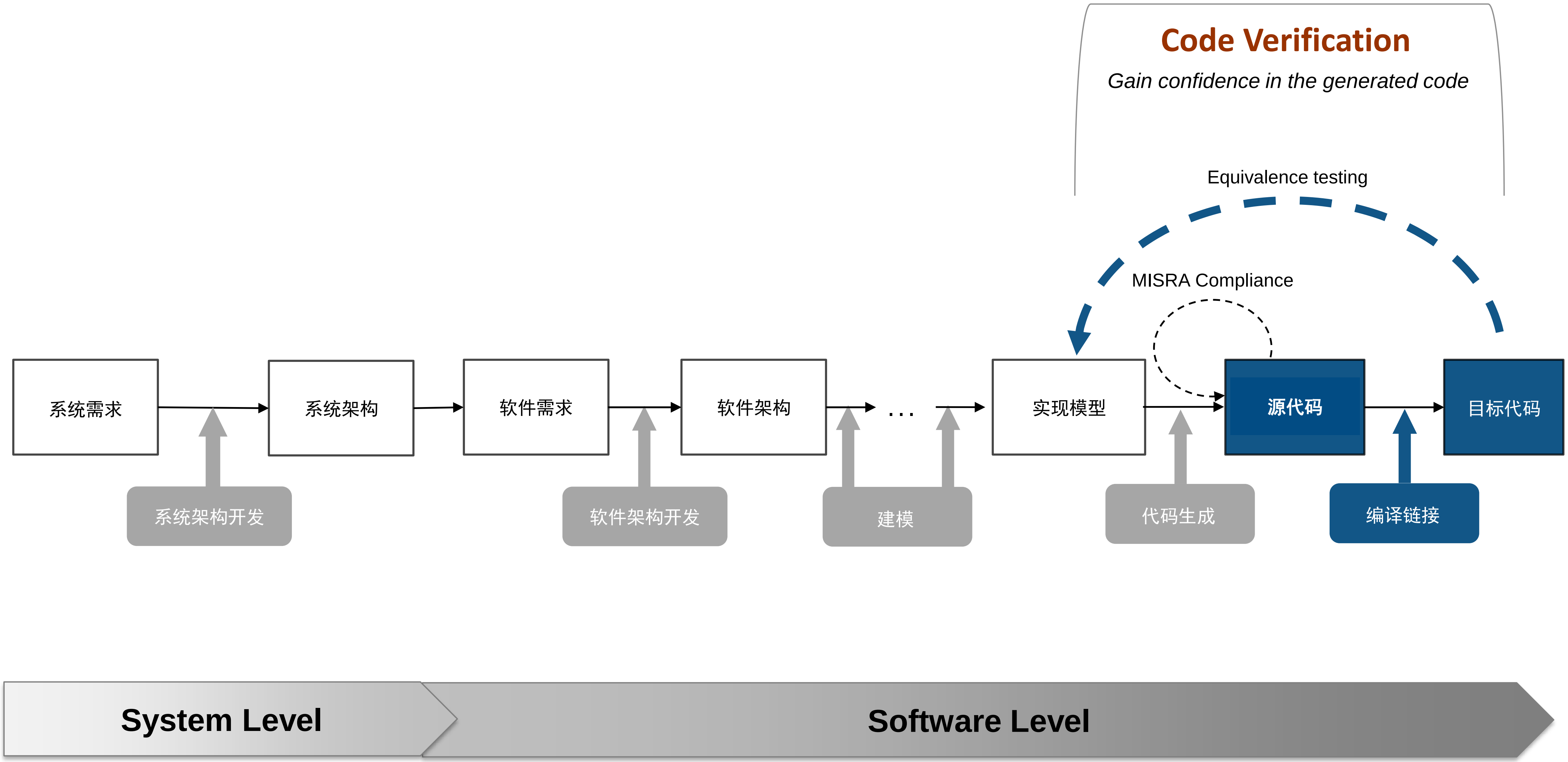
MISRA C:2012 Guidelines Summary for all Files

File	Total
C:\Users\mabualqu\MA\ISO_06_08_Software_Unit_Design_and_Implementation\WPs\ISO_6_8_5_2_Software_Unit_Implementation\slprj\ert\ControllerModeSelector\ControllerModeSelector.c	0
C:\Users\mabualqu\MA\ISO_06_08_Software_Unit_Design_and_Implementation\WPs\ISO_6_8_5_2_Software_Unit_Implementation\slprj\ert\ControllerModeSelector\ControllerModeSelector.h	0
C:\Users\mabualqu\MA\ISO_06_08_Software_Unit_Design_and_Implementation\WPs\ISO_6_8_5_2_Software_Unit_Implementation\slprj\ert\ControllerModeSelector\ControllerModeSelector_private.h	0
C:\Users\mabualqu\MA\ISO_06_08_Software_Unit_Design_and_Implementation\WPs\ISO_6_8_5_2_Software_Unit_Implementation\slprj\ert\ControllerModeSelector\ControllerModeSelector_types.h	0
C:\Users\mabualqu\MA\ISO_06_08_Software_Unit_Design_and_Implementation\WPs\ISO_6_8_5_2_Software_Unit_Implementation\slprj\ert_sharedutils\BrStatus.h	0
C:\Users\mabualqu\MA\ISO_06_08_Software_Unit_Design_and_Implementation\WPs\ISO_6_8_5_2_Software_Unit_Implementation\slprj\ert_sharedutils\rtwtypes.h	2
Total	2

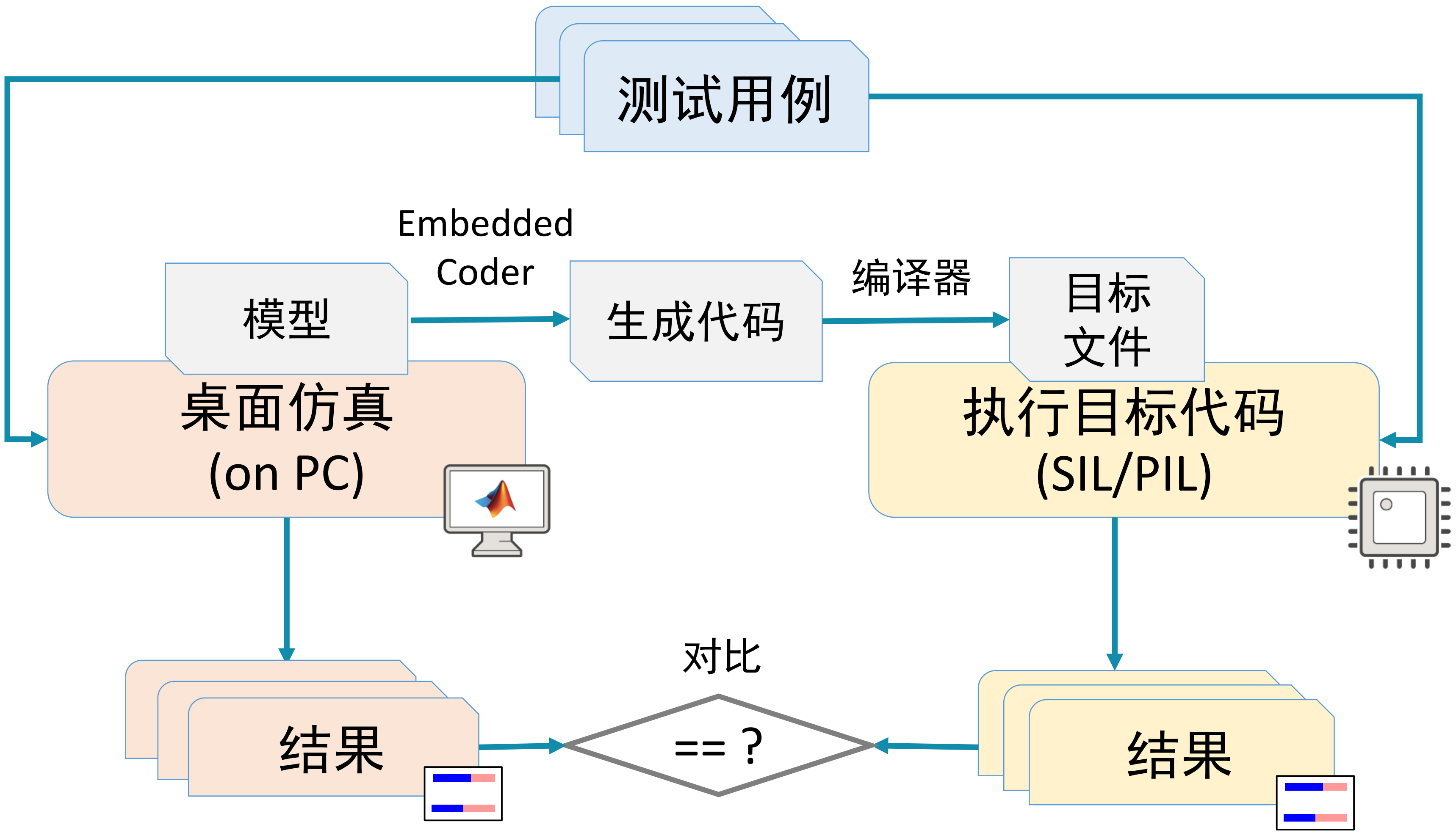
MISRA C:2012 Guidelines Summary for Enabled Guidelines

Guideline	Description	Mode	Total

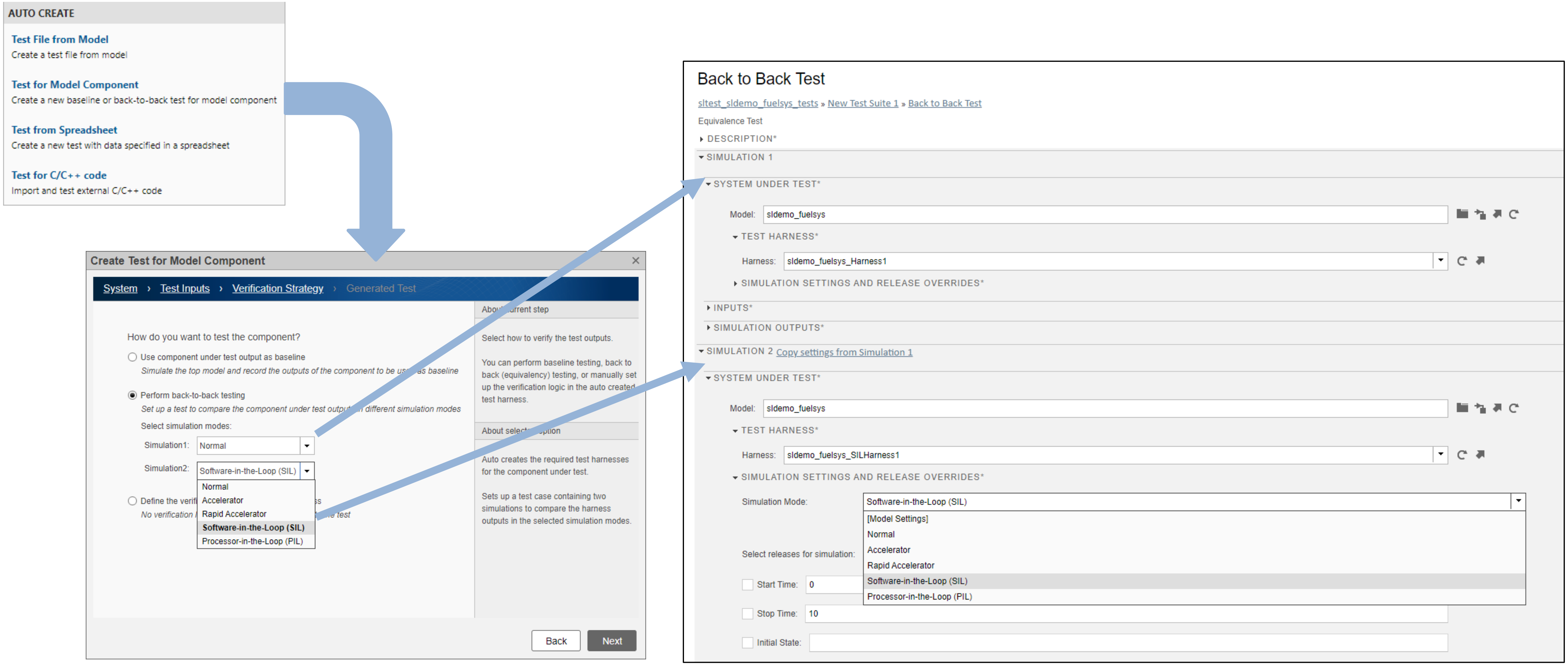
基于模型设计的参考工作流



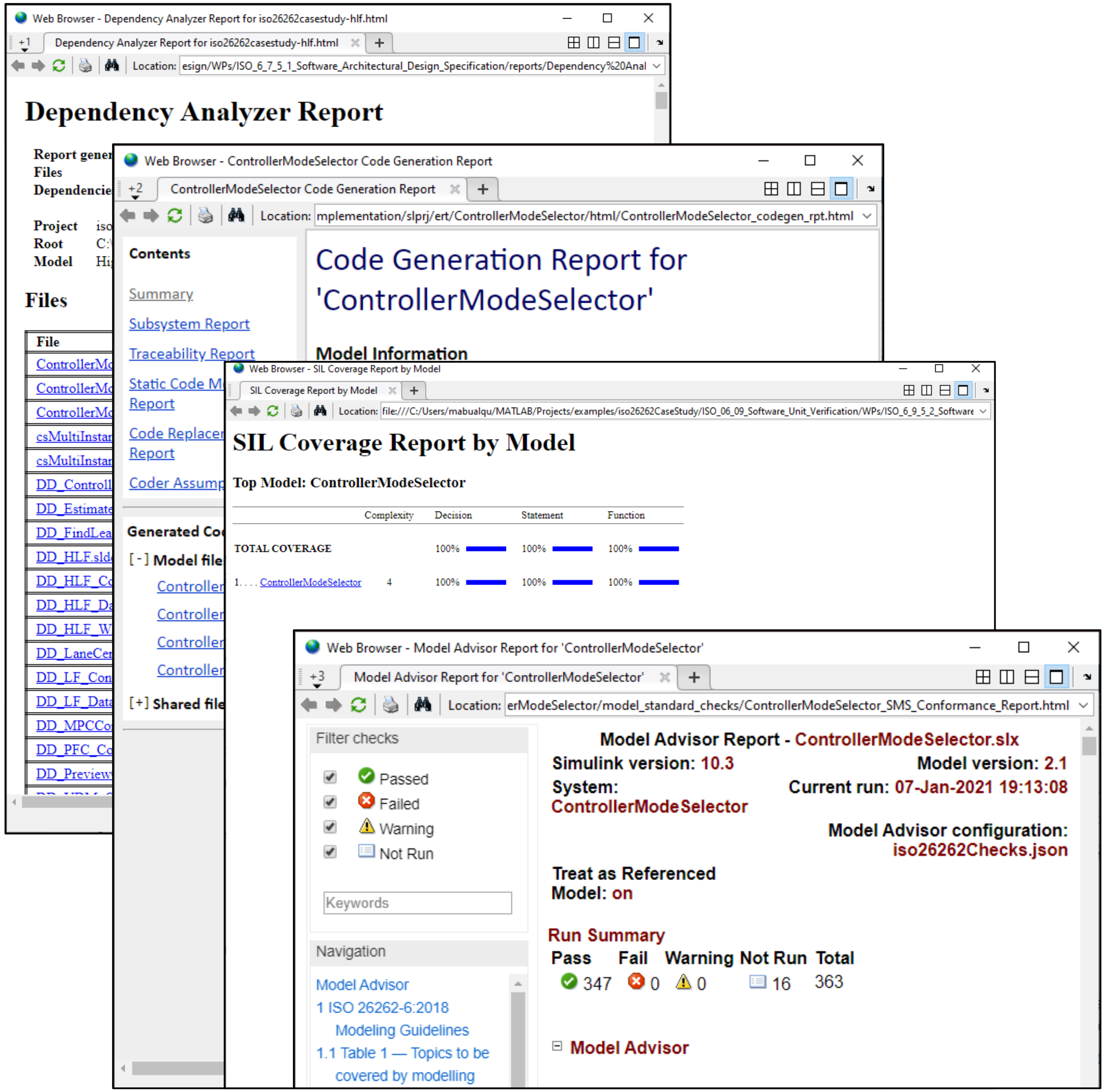
背靠背测试

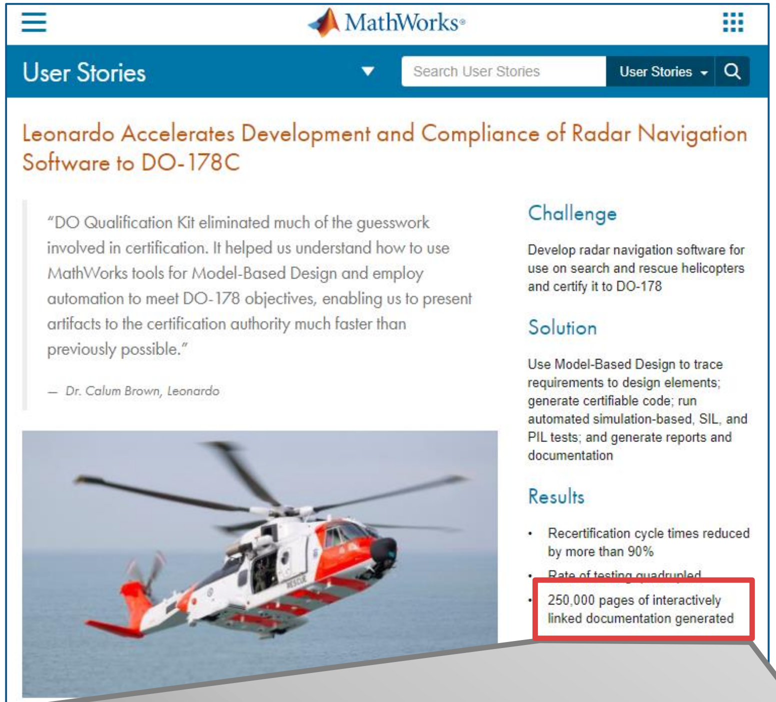

测试用例
Embedded
Coder
模型
生成代码
编译器
目标
文件
桌面仿真
(on PC)
执行目标代码
(SIL/PIL)
对比
结果
== ?
结果

使用Test Manager Wizard自动创建测试



生成报告以进行审查和审核



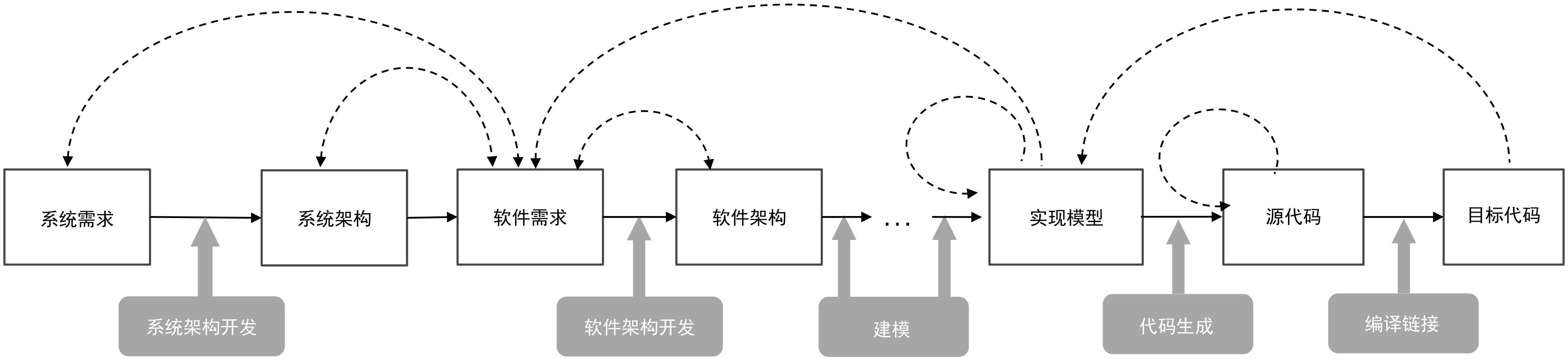


250,000 pages of interactively linked documentation generated. "Altogether, we generated the equivalent of around 250,000 pages of interactively linked test reports and other documentation using Simulink Test and Simulink Report Generator," notes Brown. "As the certification authority has *carte blanche* to review whatever they want, we felt it was easier to be able to provide evidence for everything we had done. If the DER wanted to see any particular result, it was available for them and fully linked to our model, which really built their trust."

[Leonardo User Story]

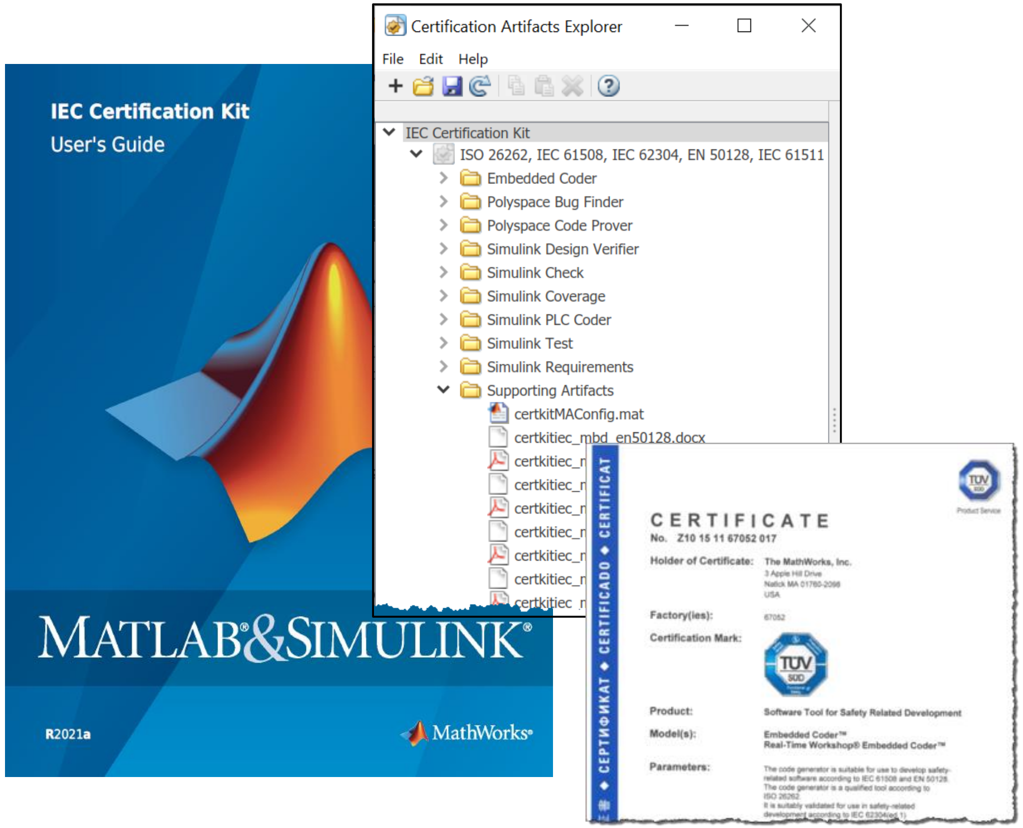
总结:基于模型设计的参考工作流

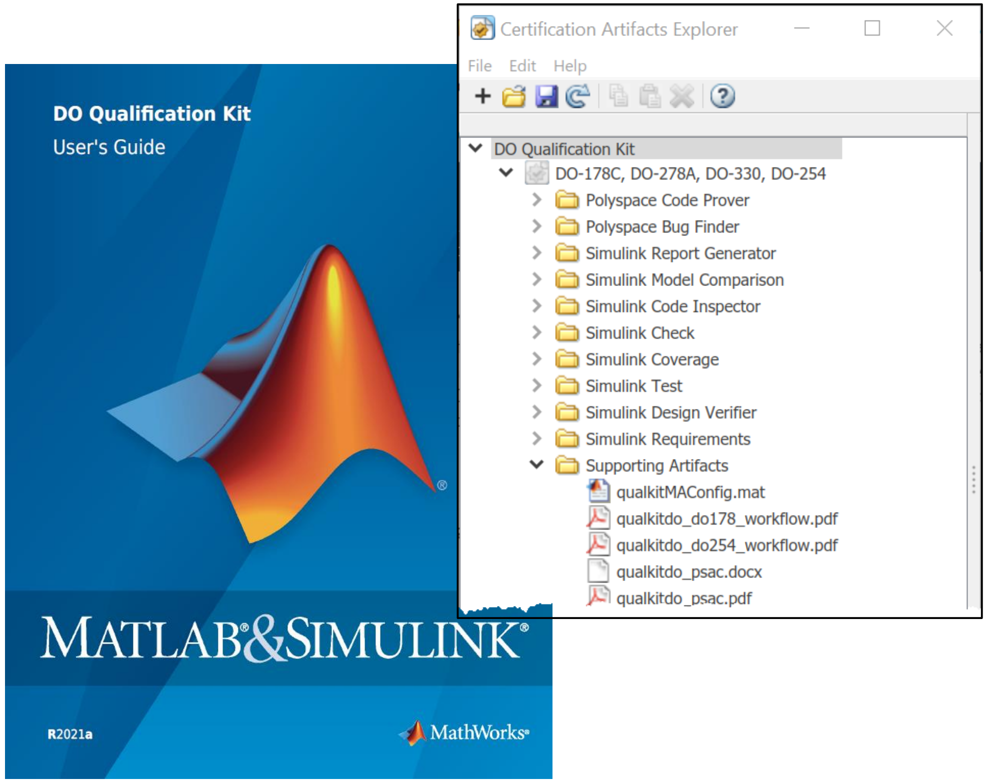
- 在需求，架构，设计，测试和代码之间建立追溯性
- 通过持续测试及早发现设计错误
- 自动代码生成
- 生成文档和报告以进行审核和审计



IEC Certification Kit 与 DO Qualification Kit

- 证明自动代码的合规性并验证产品
- 包括文档，测试用例和过程以帮助进行认证





了解更多

- [Embedded Code Generation]
- [Verification, Validation, and Test Solution Page]
- [Verifying Models and Code for High-Integrity Systems]
- [Using Model Based Design in the ISO 26262:2018 Case Study]
- [Helicopter Flight Control: A Model-Based Design Example for DO-178C and DO-331]

MATLAB EXPO

2021

感谢！

MathWorks®