

# MATLAB EXPO 2019

Simplifying Requirements Based  
Verification with Model-Based  
Design

简化基于需求的测试

Hongfei Liu

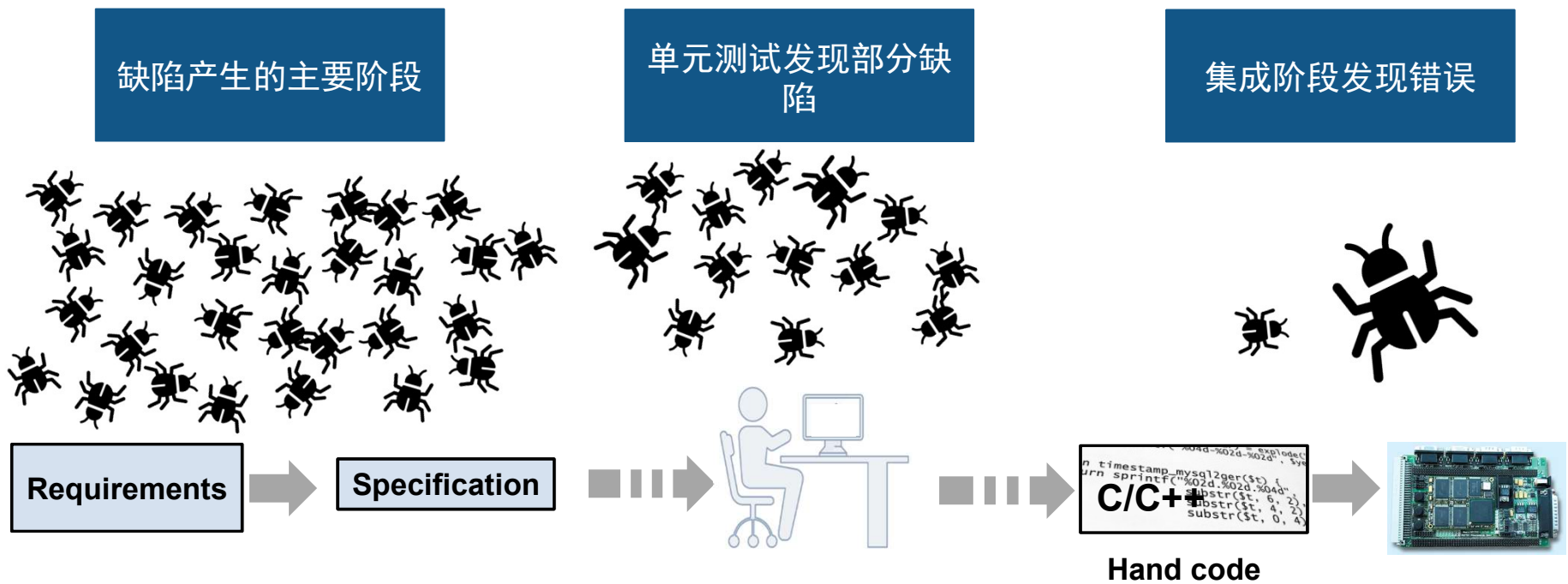


## Key takeaways

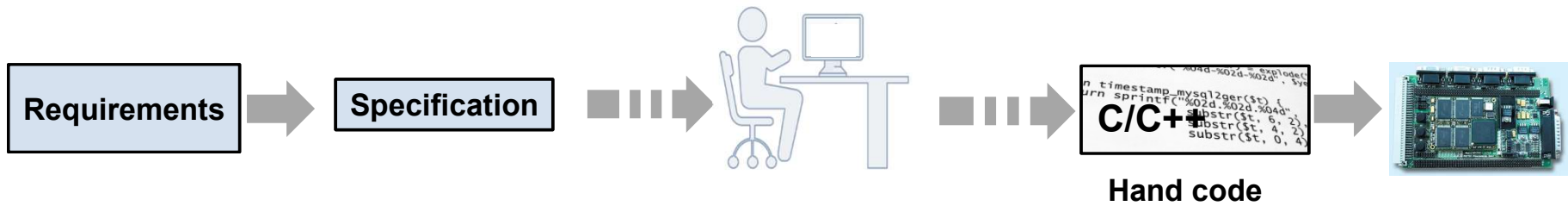
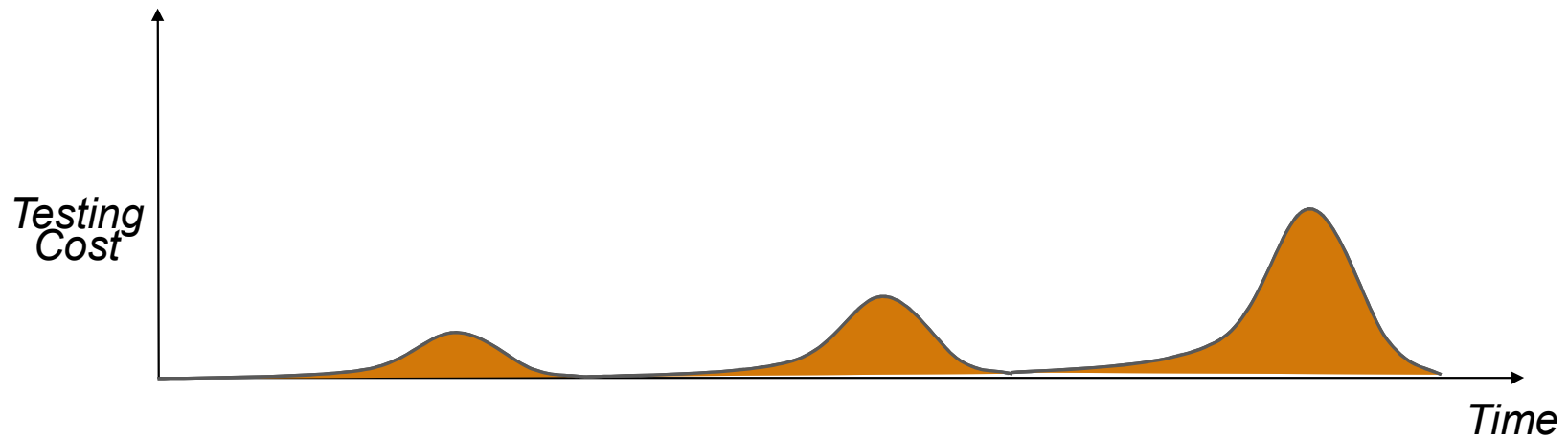
- Verify and validate requirements earlier 及早介入
- Identify inconsistencies in requirements by using unambiguous assessments 发现需求里的一致
- Traceability from requirements to design and test 完整的追溯关系

*“By enabling us to **analyze** requirements quickly, **reuse** designs from previous products, and **eliminate** manual coding errors, Model-Based Design has reduced development times and enabled us to **shorten** schedules to **meet** the needs of our customers.”*  
- MyoungSuk Ko, LS Automotive

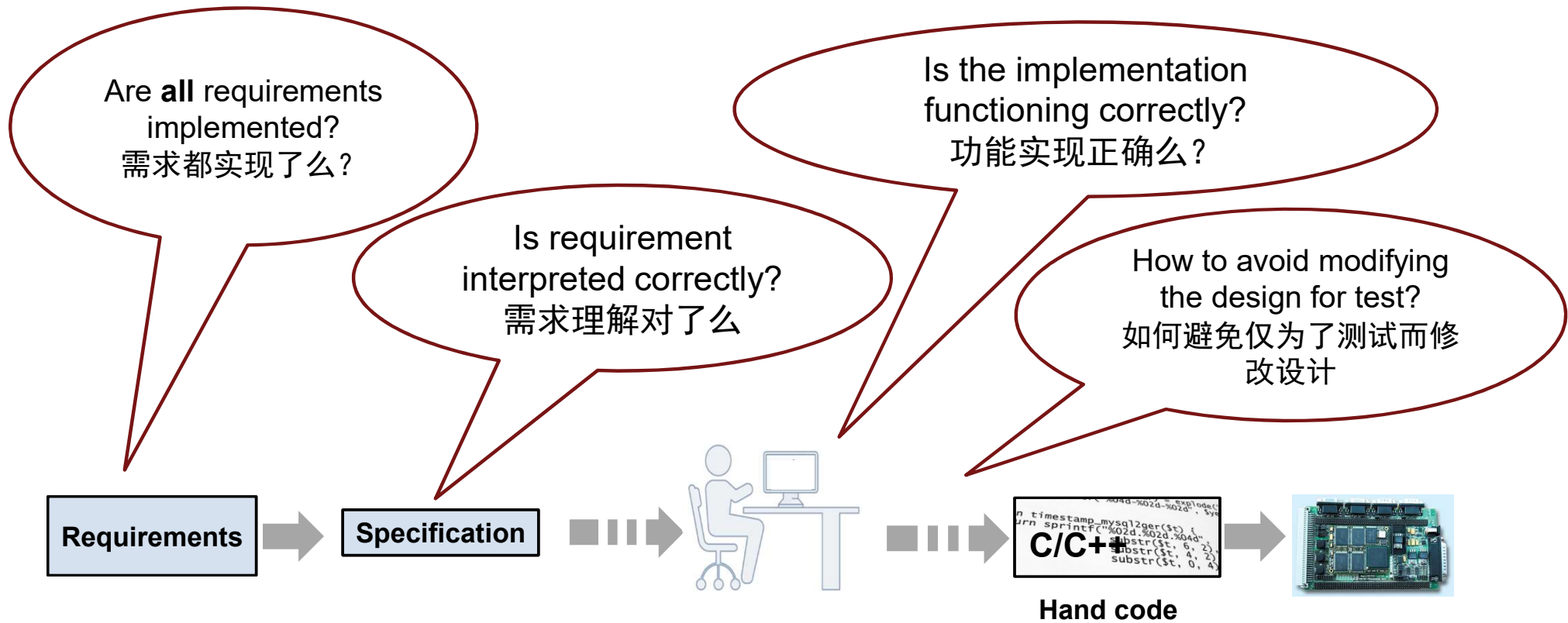
# 传统方式的挑战：缺陷发现的太晚



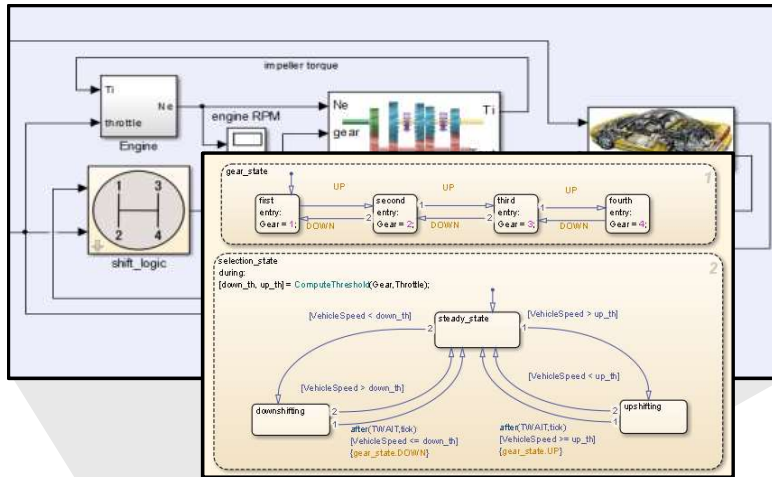
# Cost of finding errors increases over time 成本暴涨



## 基于需求的测试面临的挑战



# 基于模型的设计



## Model-Based Design enables:

- *Early testing to increase confidence in your design* 早期测试 增强信心
- *Delivery of higher quality software throughout the workflow* 贯穿每个开发环节的高质量保证

Requirements

Design Model

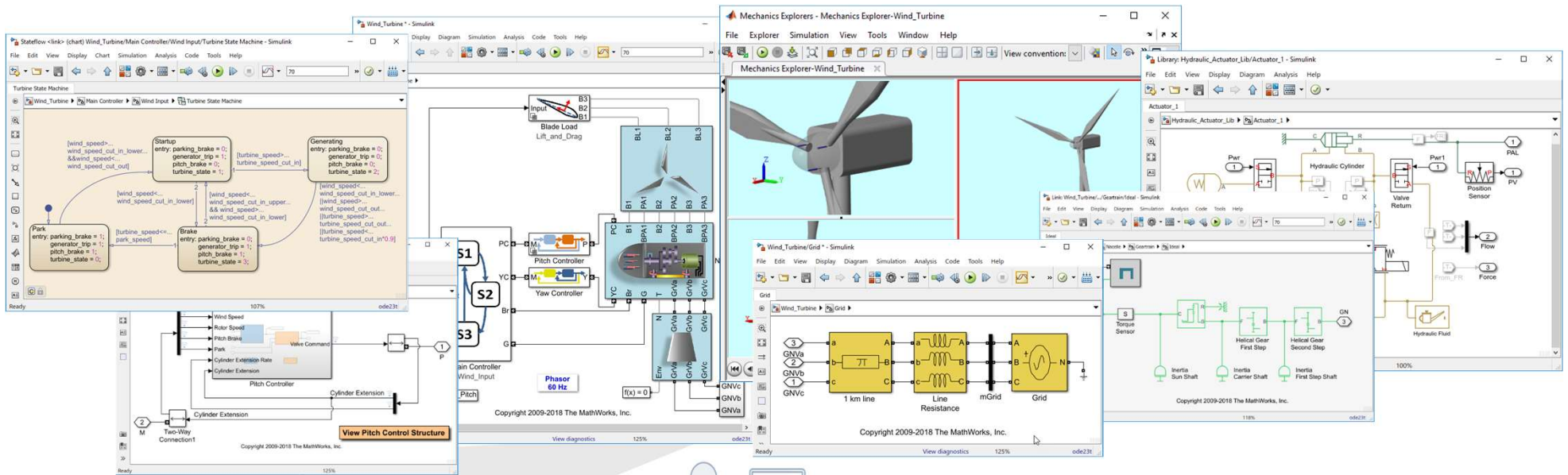


C/C++

Hand code



# Multiple languages to describe complex systems 复杂系统的多样描述



Requirements

Design Model



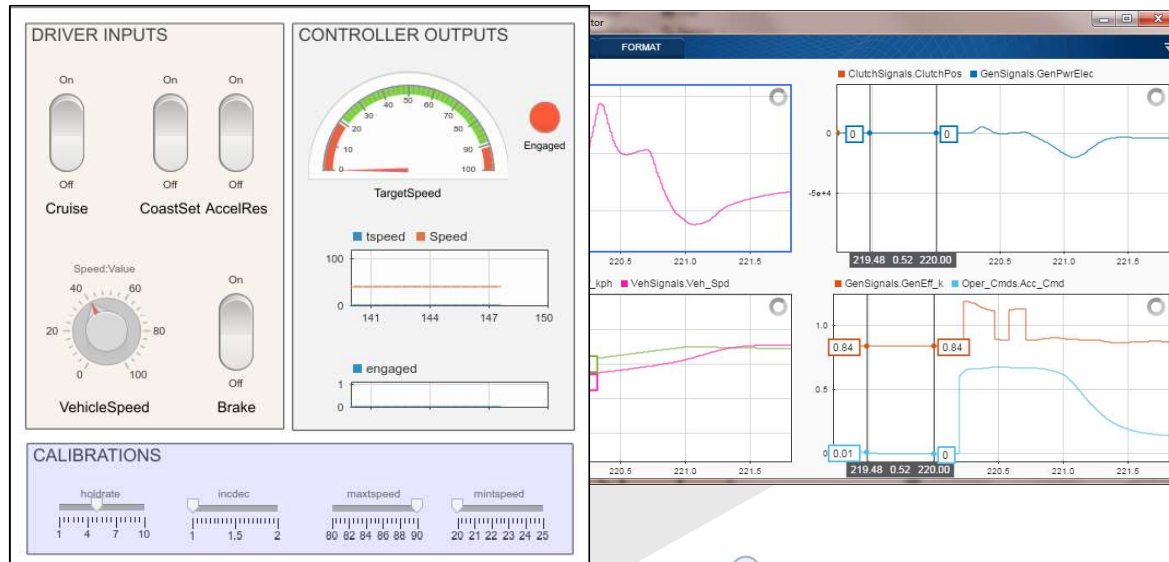
```

n timestamp_mysql2ger($s)
    printf("%02d.%02d.%04d",
        substr($t, 6, 2),
        substr($t, 4, 2),
        substr($t, 0, 4)
    )
    
```

Hand code

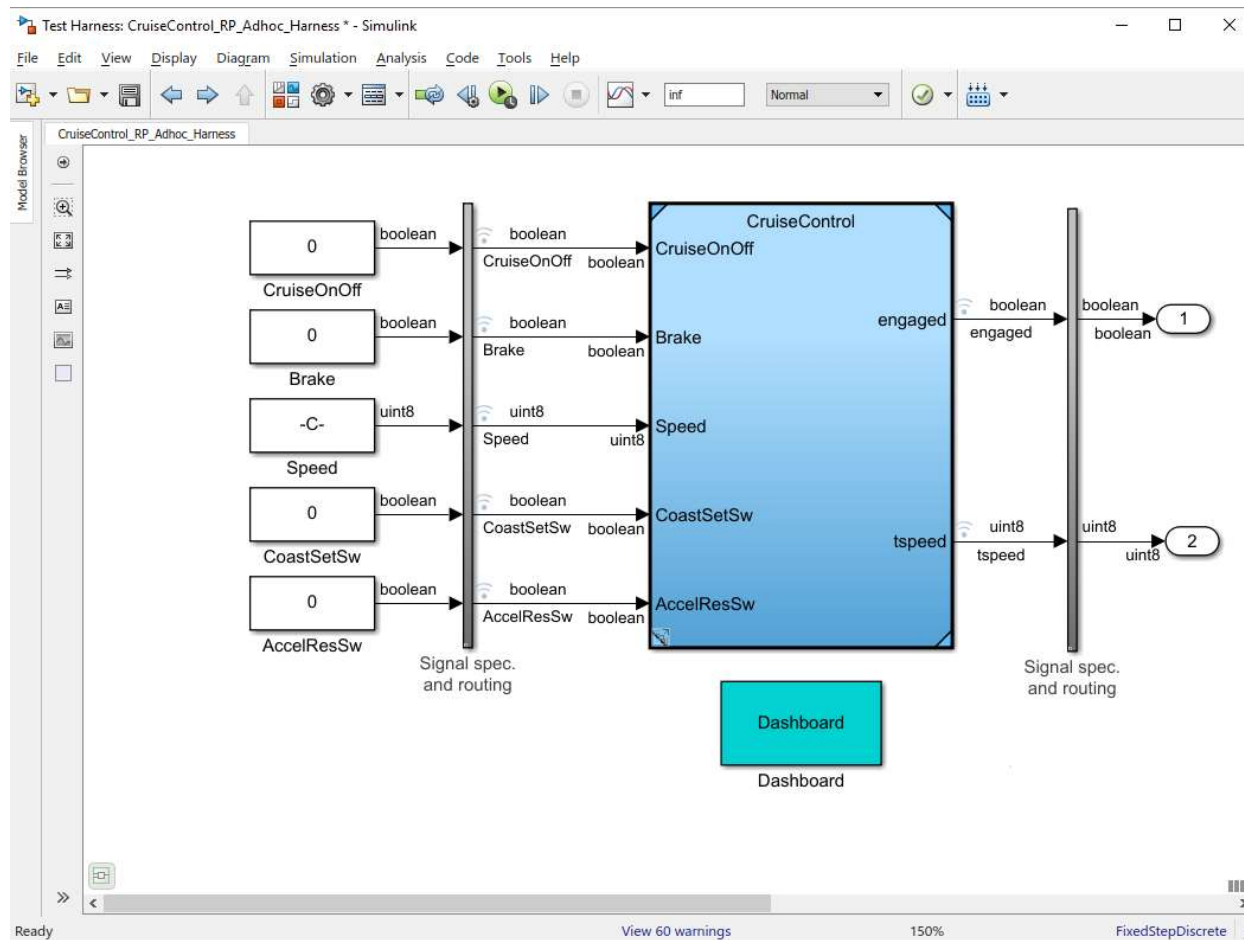


# Ad-Hoc Testing: Explore behavior and design alternatives 探索更多设计可选项

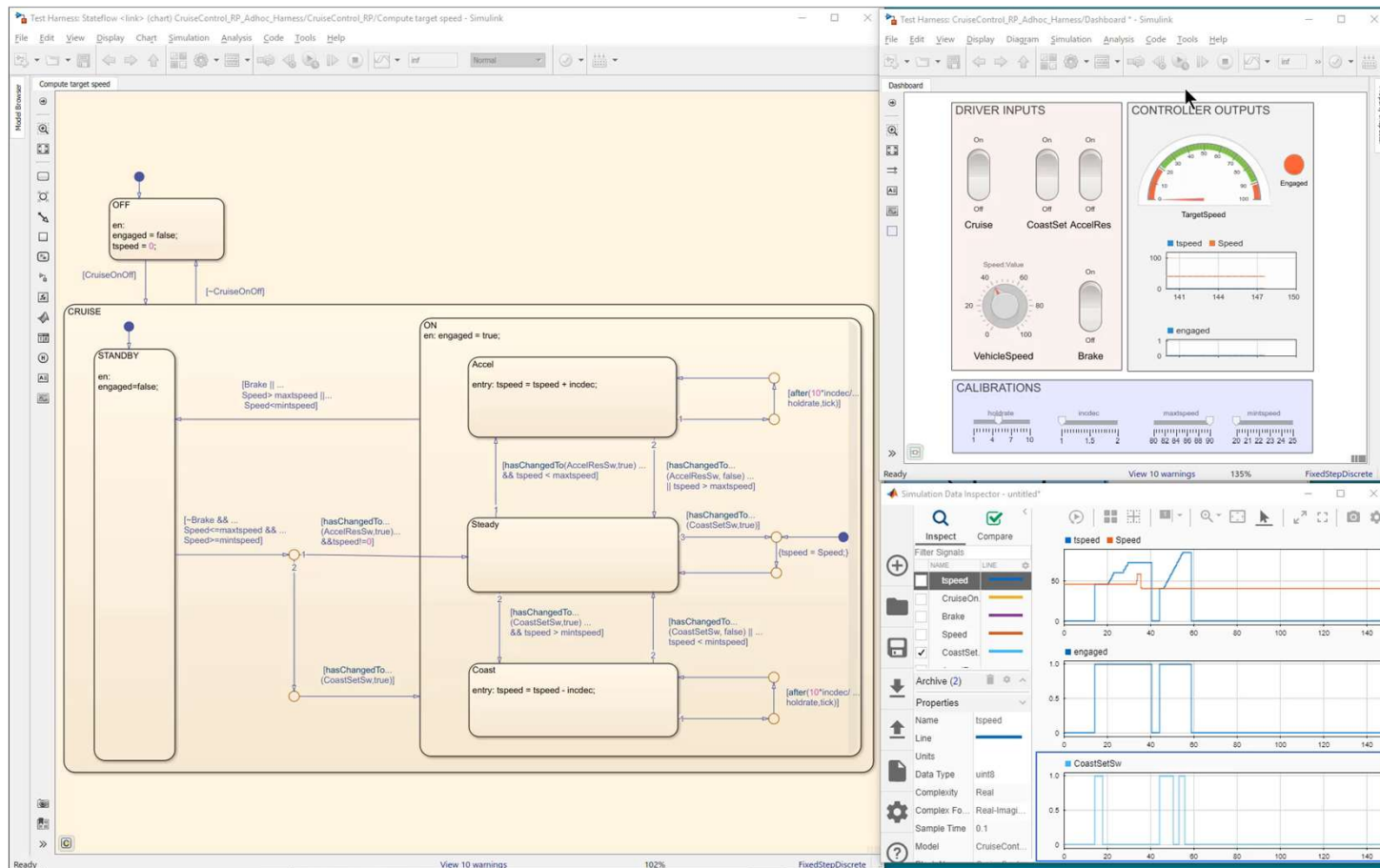




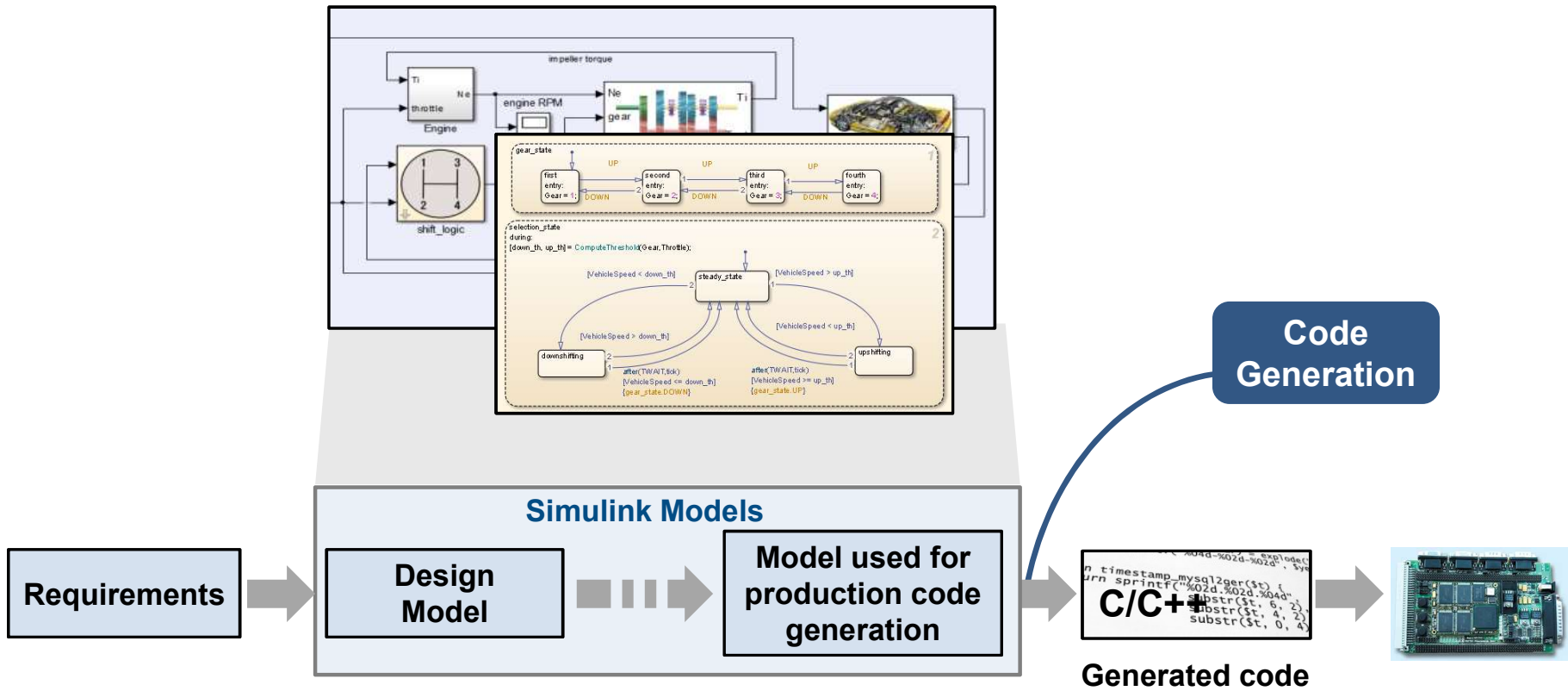
# Validate behavior earlier with simulation 早期验证



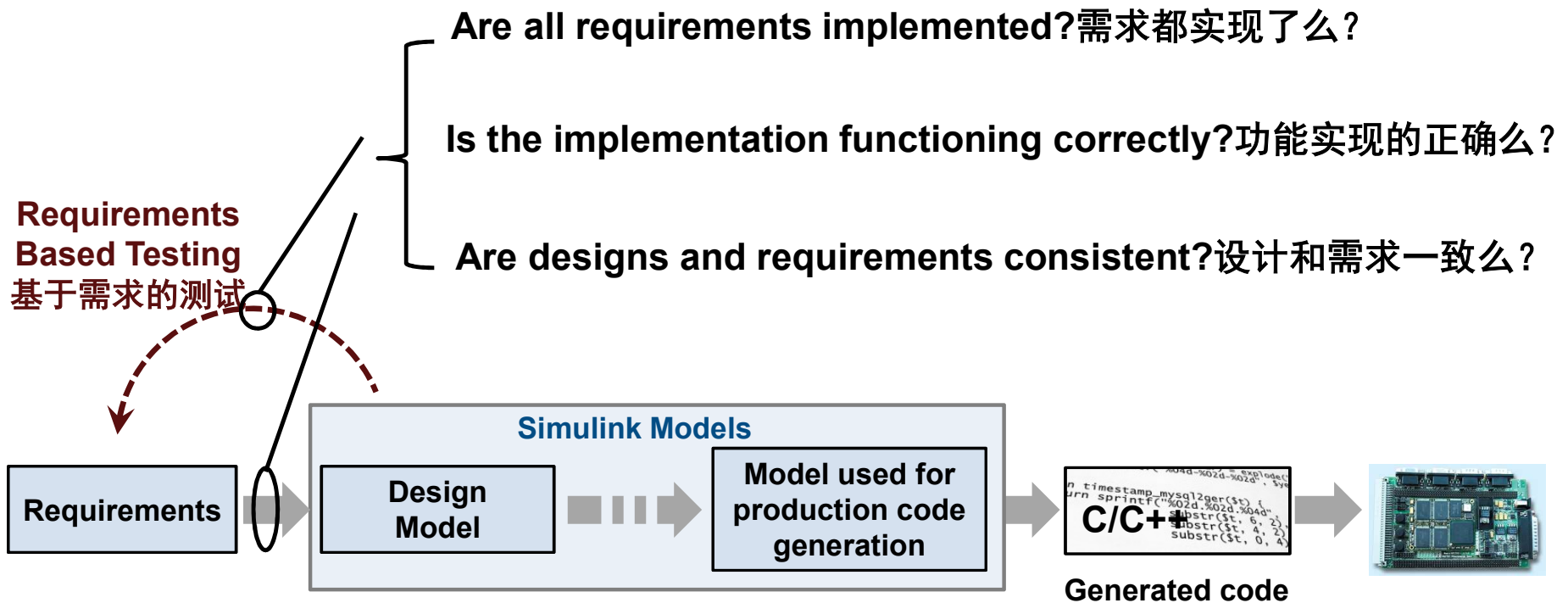
# Validate Behavior Earlier with Simulation 早期验证



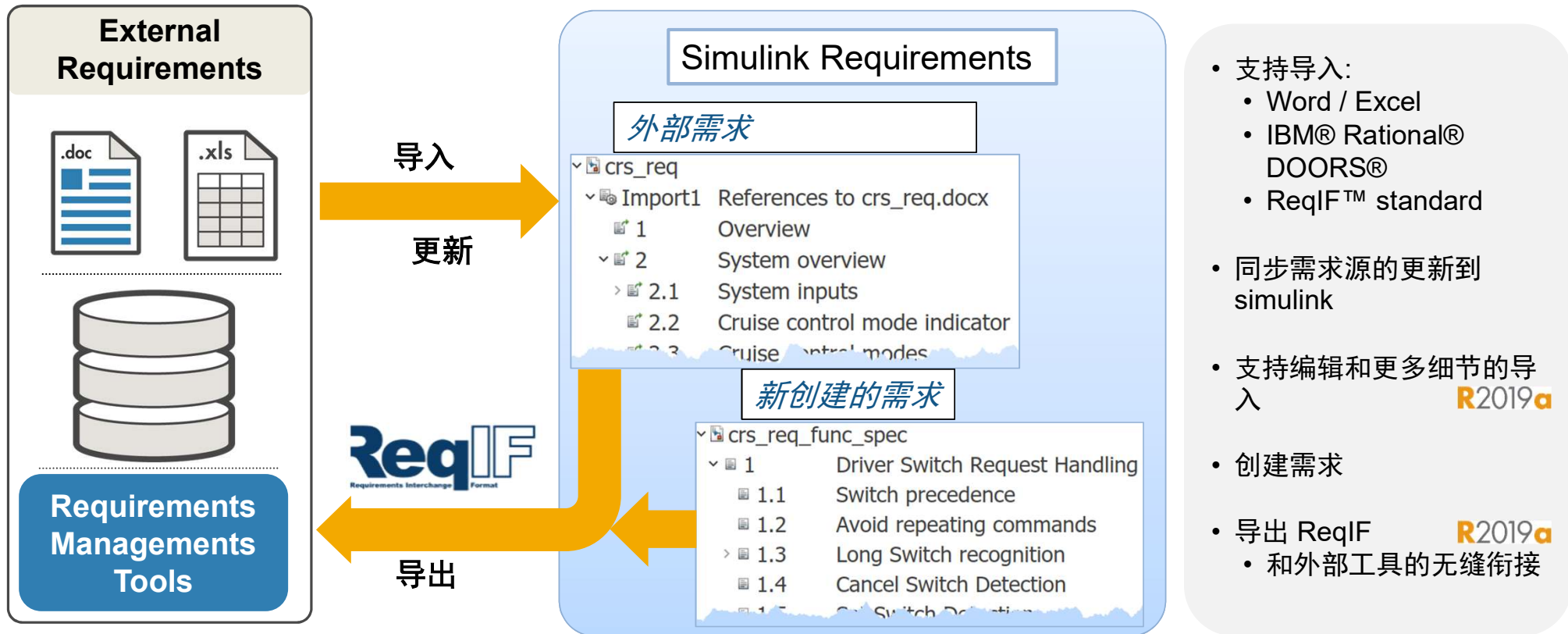
# 完整的MBD流程



# Systematically verify requirements 系统地验证需求



## Integrate with requirements tools and author requirements 和需求工具的集成



# 和外部需求管理工具的衔接

**External Requirements**

**Requirements Managements Tools**



**Simulink Requirements**

**外部的需求**

- crs\_req
  - Import1 References to crs\_req.docx
    - 1 Overview
    - 2 System overview
      - 2.1 System inputs
      - 2.2 Cruise control mode indicator
      - 2.3 Cruise control modes

**新创建的需求**

- crs\_req\_func\_spec
  - 1 Driver Switch Request Handling
    - 1.1 Switch precedence
    - 1.2 Avoid repeating commands
    - 1.3 Long Switch recognition
    - 1.4 Cancel Switch Detection
    - 1.5 Cancel Switch Detection

- 支持导入:
  - Word / Excel
  - IBM® Rational® DOORS®
  - ReqIF™ standard
- 同步需求源的更新到 simulink
- 支持编辑和更多细节的导入 **R2019a**
- 创建需求
- 导出 ReqIF **R2019a**
  - 和外部工具的无缝衔接

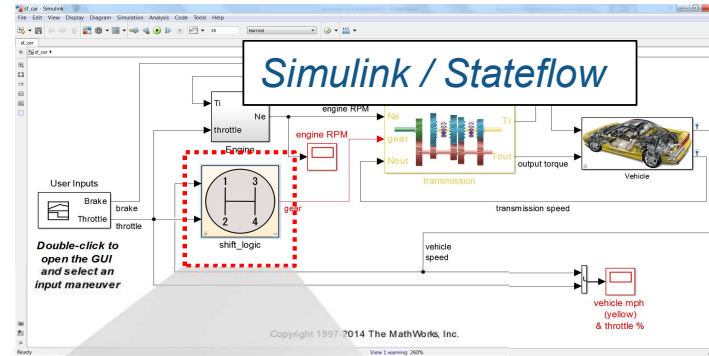
# Simulink中的需求测试

## Requirements

- TransmissionReq
  - 1 Transmission Operating Modes
    - 1.1 Reverse cannot be entered from drive
    - 1.2 Engine only starts in Park

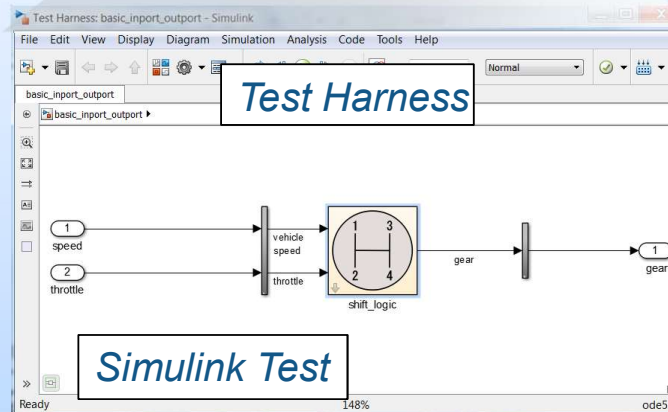
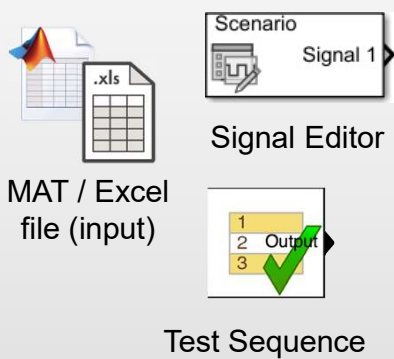
实现于

测试于

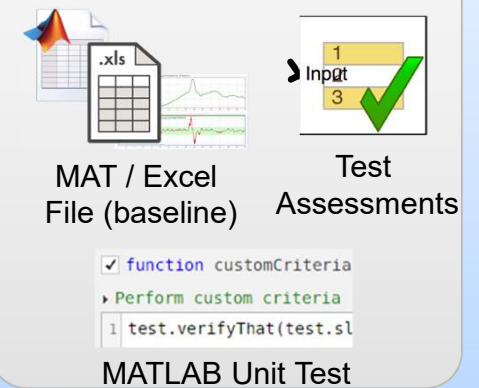


## Test Case

### Inputs



### Assessments



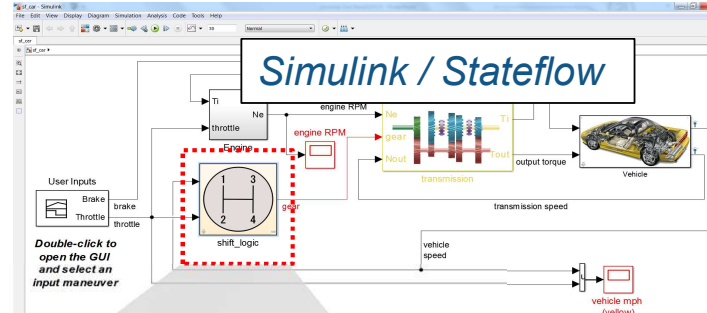
# Simulink中的需求测试

## Requirements

- crs\_req\_func\_spec
  - 1 Driver Switch Request Handling
    - 1.1 Switch precedence
    - 1.2 Avoid repeating commands

实现于

测试于

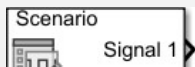


## Test Case

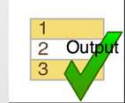
### Inputs



MAT / Excel file (input)



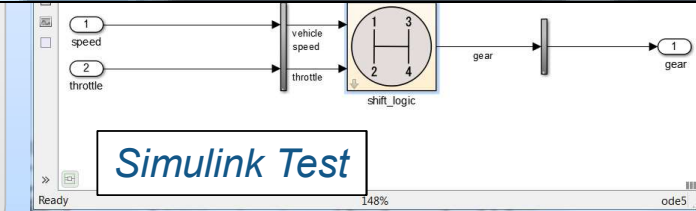
Signal Editor



Test Sequence

## Implemented

## Verified



MAT / Excel File (baseline)

Test Assessments

```

function customCriteria
Perform custom criteria
1 test.verifyThat(test.sl
    
```

MATLAB Unit Test



# 举例: 测试 Heat Pump Controller Requirements

**1 Requirements for the basic Heatpump Controller**  
Temperature difference is defined as the difference between the room and the set temperature. The controller shall turn the fan on when the temperature difference has reached a certain level, to circulate the air. The controller shall turn the heatpump on when the temperature difference has reached another level, to heat or cool the space.

**1.1 Idle when Temperature in Range**  
If the temperature difference is less than 1 degrees, the system shall be idle with all signals off.

**1.2 Activate Fan**  
The fan shall activate when the temperature difference is greater than or equal to 1 degrees.

**1.3 Activate Heat Pump**  
The pump shall activate when the temperature difference is greater than or equal to 2 degrees for more than 2 seconds and stay active for at least 2 seconds.

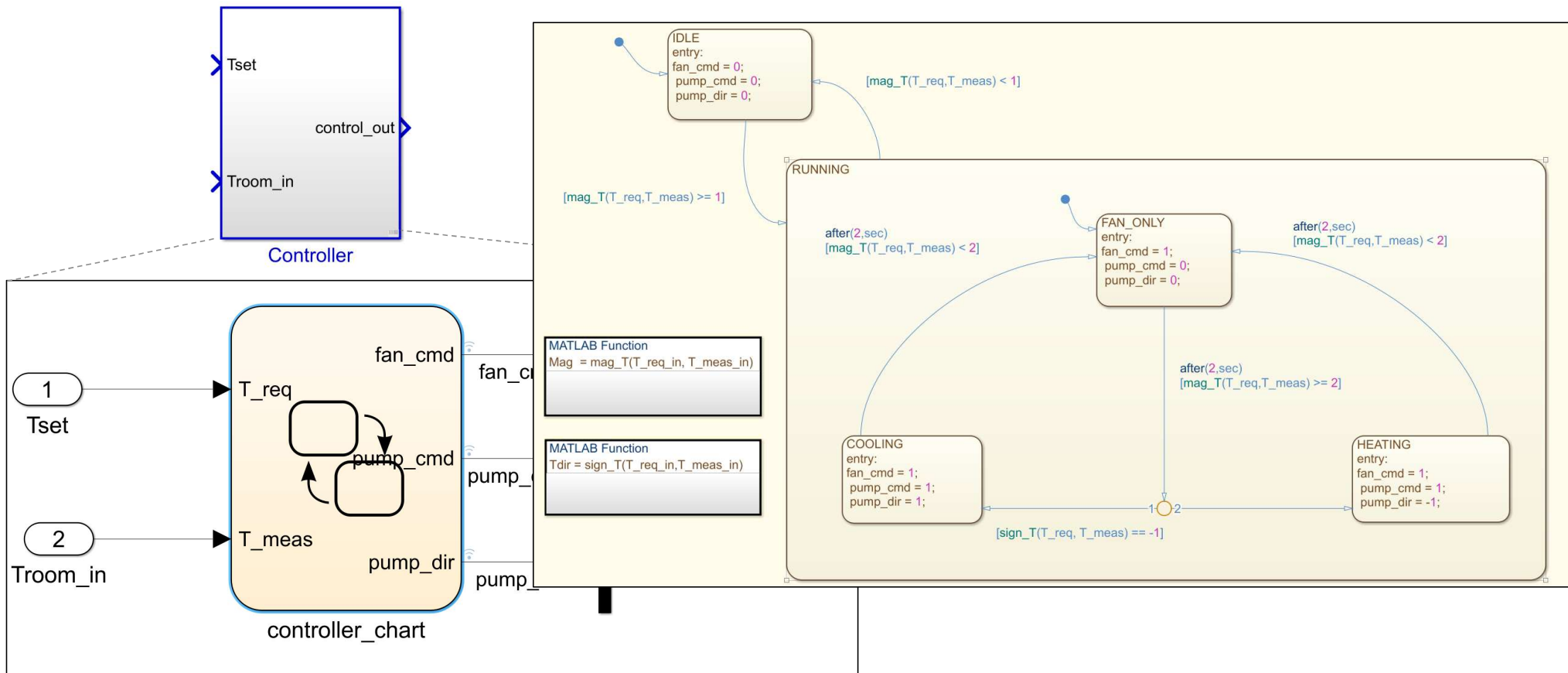
**1.3.1 Cool Mode**  
If the room temperature is greater than the set temperature, the system shall cool the space.

**1.3.2 Heat Mode**  
If the room temperature is less than the set temperature, the system shall heat the space.

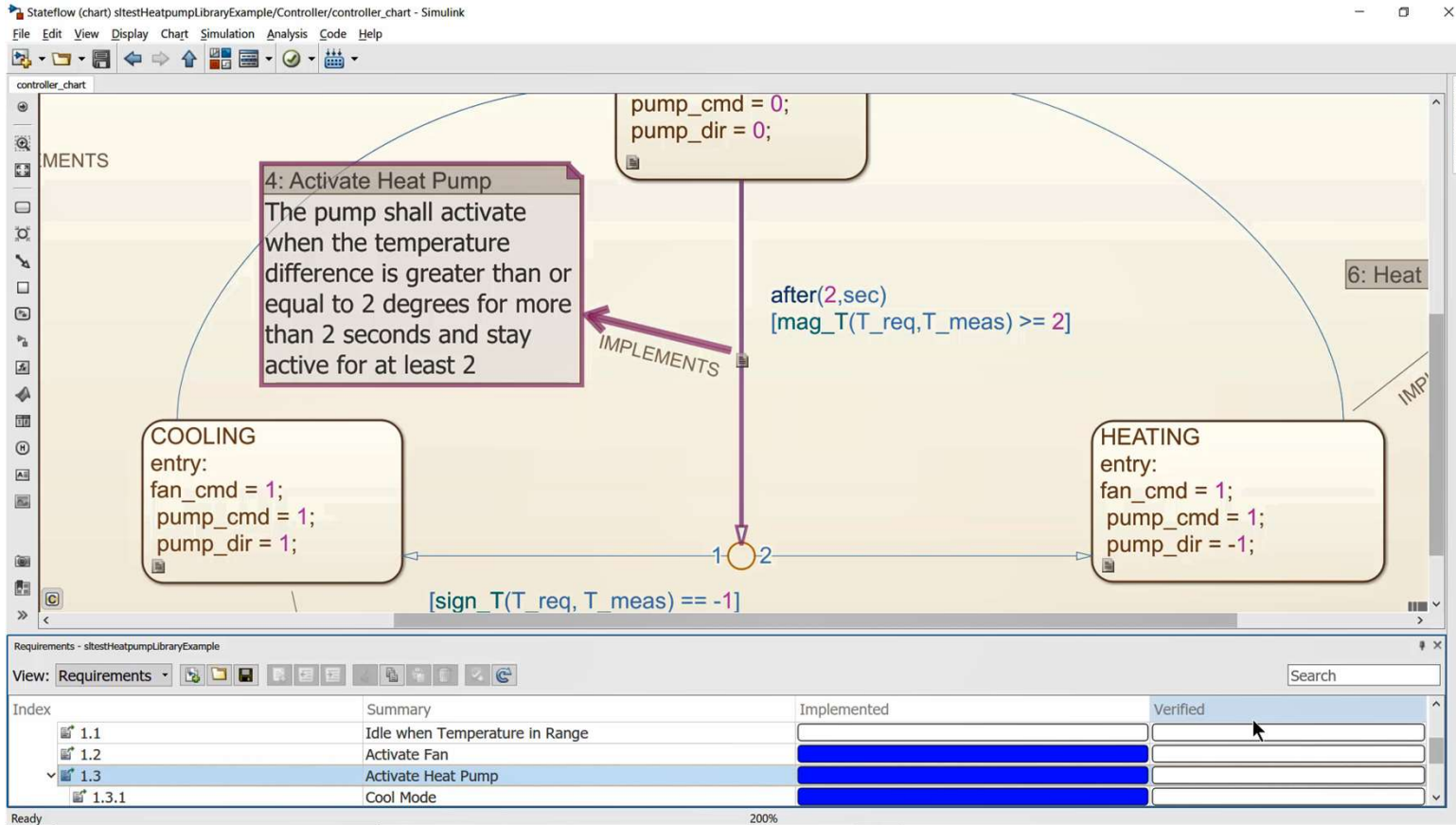
**1.4 Max Temperature**  
The difference between the room temperature and the set temperature should never exceed 6 degrees

*Requirements in DOORS*

# 举例: Heat Pump Controller 实现



# 在模型中建立到需求的追溯



# 从需求的角度审查实现的模型

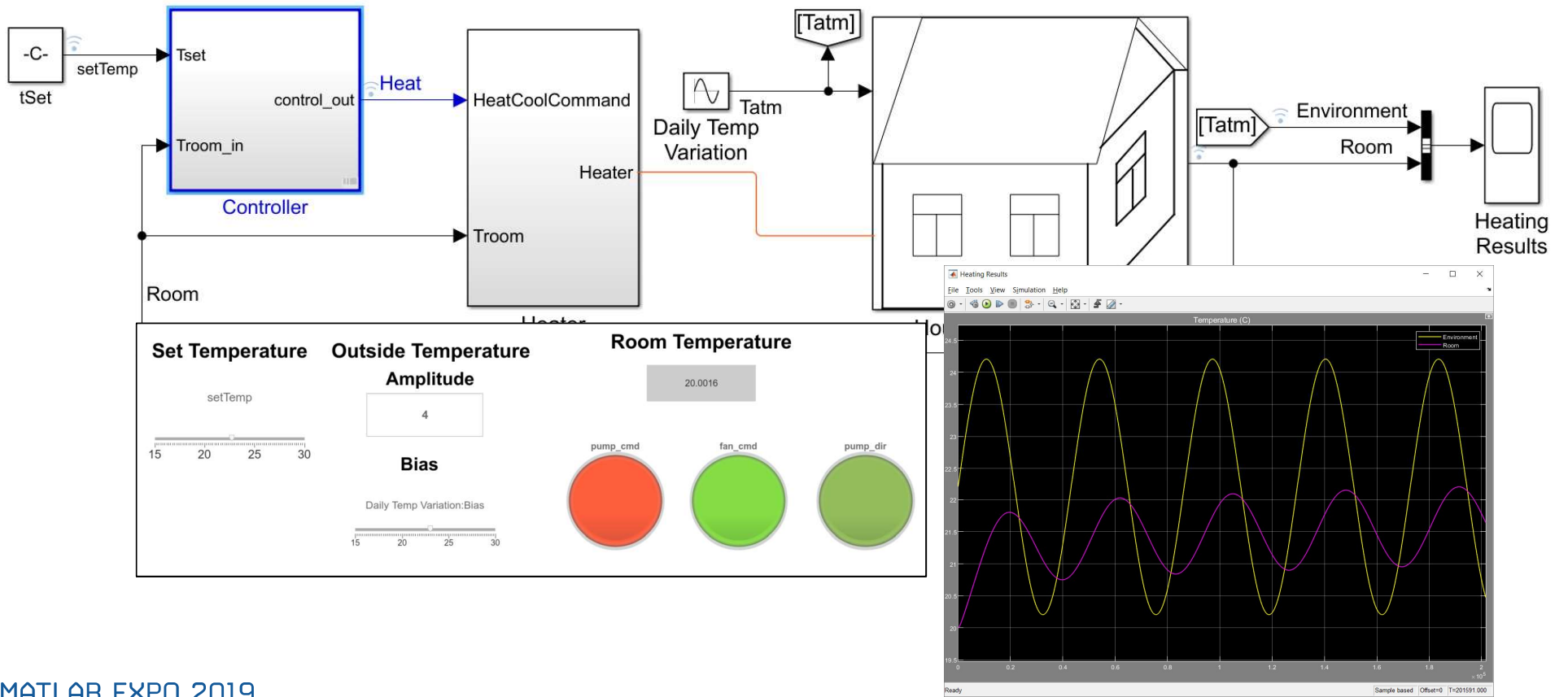
The screenshot displays a Stateflow model for a heating system controller. Key elements include:

- Requirement Annotations:** A box labeled "Requirement Annotations" points to requirement 3: "3: Activate Fan. The fan shall activate when the temperature difference is greater than or equal to 1 degrees." This requirement is circled in red.
- Badges:** A box labeled "Badges" points to requirement 4: "4: Activate Heat Pump. The pump shall activate when the temperature difference is greater than or equal to 2 degrees for more than 2 seconds and stay active for at least 2." This requirement is also circled in red.
- Implementation and Verification Status:** A box labeled "Implementation and Verification Status" points to a table at the bottom of the interface. The table shows the status of requirements 1.3, 1.3.1, and 1.3.2.
- Property Inspector:** A box labeled "Property Inspector" points to the right-hand panel, which shows details for Requirement 4, including its type (Functional), index (1.3), and summary (Activate Heat Pump). It also lists the implementation and verification links.

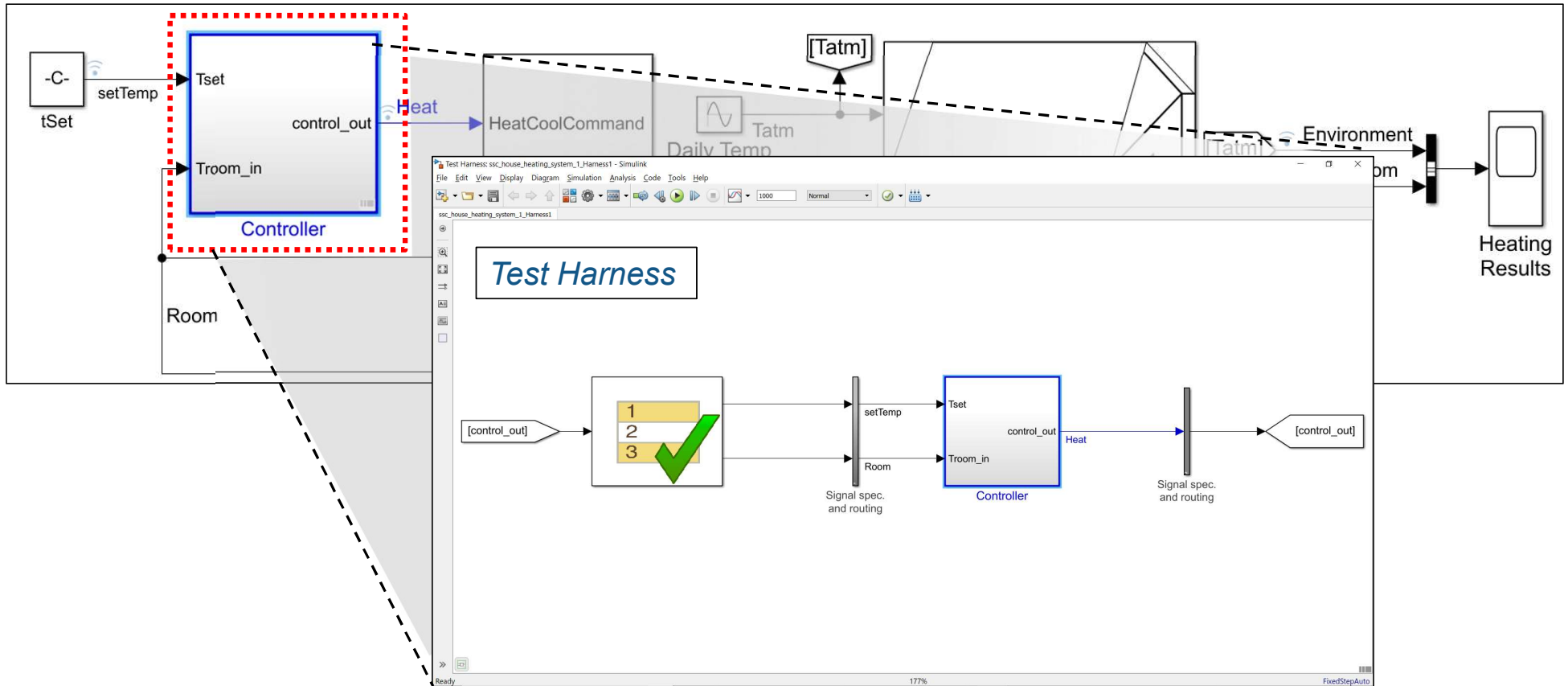
Index	Summary	Implemented	Verified
1.3	Activate Heat Pump	<div style="width: 100%; height: 10px; background-color: blue;"></div>	<div style="width: 25%; height: 10px; background-color: green;"></div>
1.3.1	Cool Mode	<div style="width: 100%; height: 10px; background-color: blue;"></div>	<div style="width: 0%; height: 10px; background-color: green;"></div>
1.3.2	Heat Mode	<div style="width: 100%; height: 10px; background-color: blue;"></div>	<div style="width: 0%; height: 10px; background-color: green;"></div>

Browser

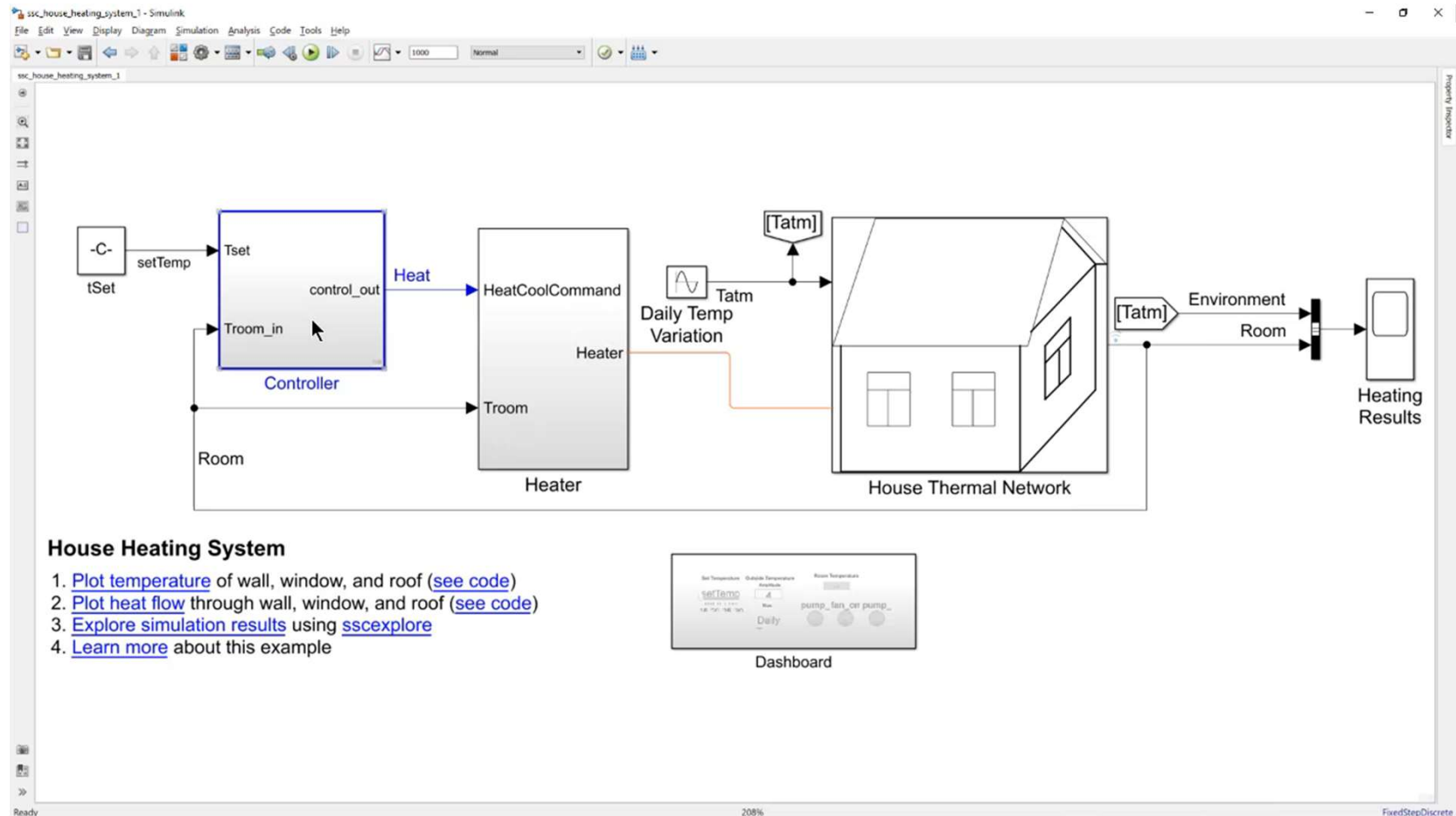
# 使用被控对象模型进行仿真



# 用Test Harness隔离测试的模型



# 用Test Harness隔离测试的模型



# Test Sequence Block: 分步设计测试顺序

The screenshot shows the Test Sequence Editor window with the following components:

- Toolbar:** Contains icons for file operations (save, undo, redo, up, down), test execution (run, stop, refresh), and other utility functions.
- Left Panel:**
  - Symbols:**
    - Input:** 1. control\_out
    - Output:** 1. Tset, 2. Troom\_in
    - Local:**
    - Constant:**
    - Parameter:**
    - Data Store Memory:**
  - Step Hierarchy:**
    - Initialize
    - Cold\_Outside
    - Hot\_Outside
- Main Table:**

Step	Transition	Next Step
<b>Initialize</b> %% Initialize data inputs. Tset = 23; Troom_in = 23;	1. true	Cold_Outside ▼
<b>Cold_Outside</b> %% Check heating mode Troom_in = 23 - ramp(et*0.2);	1. Troom_in <= 15	Hot_Outside ▼
<b>Hot_Outside</b> %% Check cooling mode Troom_in = 23 + ramp(et*0.2);	1. Troom_in >= 27	Return_Idle ▼
<b>Return_Idle</b> %% Return to idle mode Troom_in = Troom_in - ramp(et*0.2);	1. Troom_in <= 22	End ▼
<b>End</b> Troom_in = 22		



## Test Assessments: 将需求变成公式?

### Activate Heat Pump

If the temperature difference exceeds 2 degrees for more than 2 seconds, then the pump shall activate for at least 2 seconds

When <condition 1> is true,  
Then <condition 2> must be true for some time

Simple concept

$$(|x_1 - x_2| \geq x_3)^{\varepsilon} \wedge \square_{[0,t_1)}(|x_1 - x_2| \geq x_3) \rightarrow \square_{[0,t_2)}x_4$$

Hard to formalize

MTL logic

# 用填表的方式创建assessments

R2019a

Simulation Test

Select releases for simulation: Current

Create Test Case from External File

REQUIREMENTS

SYSTEM UNDER TEST\*

Model: ssc\_house\_heating\_system\_1

TEST HARNESS\*

Harness: ssc\_house\_heating\_system\_1\_Harness1

SIMULATION SETTINGS OVERRIDES

INPUTS

SIMULATION OUTPUTS

LOGICAL AND TEMPORAL ASSESSMENTS\*

EN	NAME	ASSESSMENT	REQUIREMENTS	VISUAL REPRESENTATION
<input checked="" type="checkbox"/>	Activate_Pump	At any point of time, if $\text{abs}(\text{roomTemperature} - \text{setTemperature}) \geq \text{threshold}$ becomes true and stays true for at least 2 seconds then, starting from rising edge of trigger, with a delay of at most 2 seconds, $\text{pumpCmd}$ must stay true for at least 2 seconds	None	

SYMBOLS

- roomTemperature
- setTemperature
- threshold
- pumpCmd

+ Add Assessment - Delete + Add Symbol - Delete

COVERAGE SETTINGS\*

# 执行assessments 来验证需求

**Activate\_Pump**

ASSESSMENT

► At any point of time, if  $\text{abs}(\text{roomTemperature} - \text{setTemperature}) \geq \text{threshold}$  becomes true and stays true for at least 2 seconds then, starting from rising edge of trigger, with a delay of at most 2 seconds, **pumpCmd** must stay true for at least 2 seconds

SYMBOLS

- roomTemperature
- setTemperature
- threshold

**Expected Behavior**

**Actual Result**

**Explanation**

Assessment 'Activate\_Pump' failed when triggered at 11 s.

- Trigger condition ' $\text{abs}(\text{roomTemperature} - \text{setTemperature}) \geq \text{threshold}$ ' is true at 11 s.
- Expected response condition to be true within 11 s and 13 s.
  - i.e. with a delay of at most 2 s after 'rising edge' of trigger.
- Expected 'pumpCmd' to be true at 13 s for at least 2 s, actual value at 13 s is false.

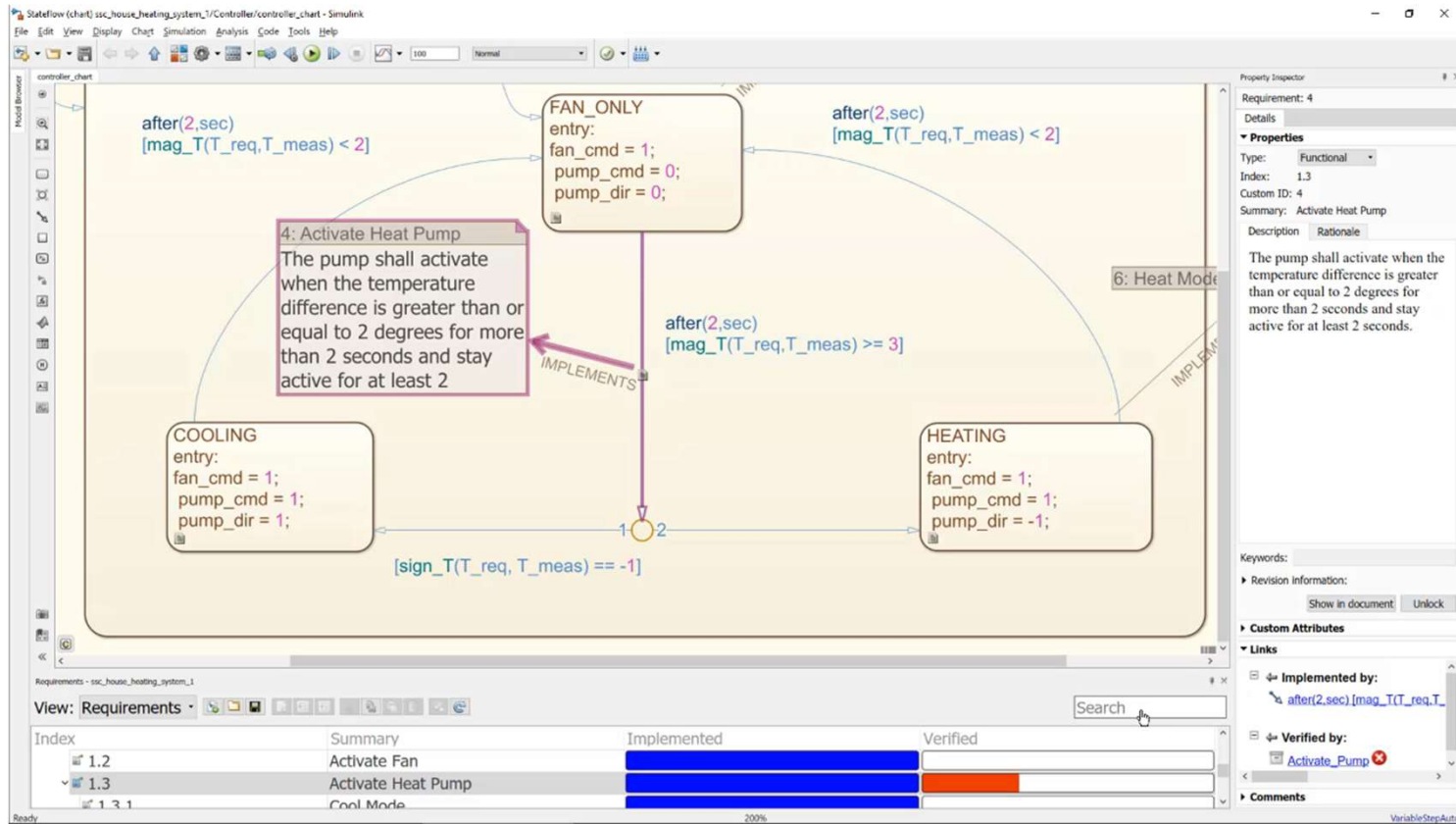
EXPRESSION TREE

▼ Activate\_Pump: At any point in time, if  $\text{abs}(\text{roomTemperature} - \text{setTemperature}) \geq \text{threshold}$  becomes true and stays true for at least 2 seconds then, starting from rising edge of trigger, with a delay of at most 2 seconds, **pumpCmd** must stay true for at least 2 seconds

▼ if  $\text{abs}(\text{roomTemperature} - \text{setTemperature}) \geq \text{threshold}$  becomes true and stays true for at least 2 seconds then, starting from rising edge of trigger, with a delay of at most 2 seconds, **pumpCmd** must stay true for at least 2 seconds

PLOTS

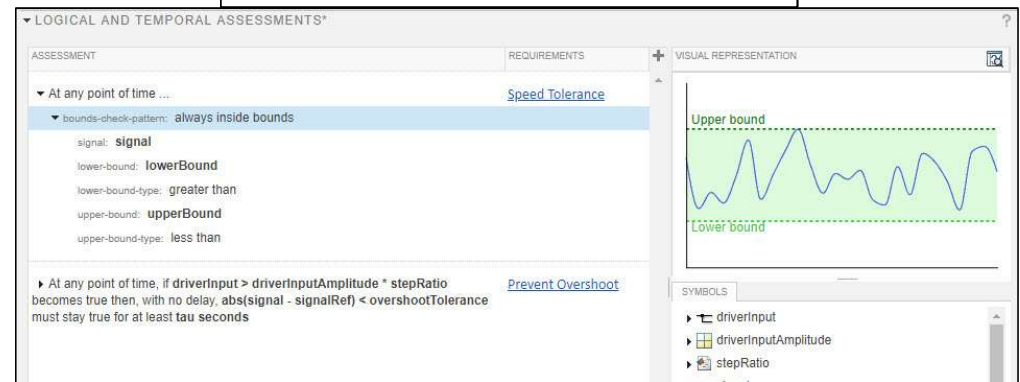
# 利用追溯审查模型和需求



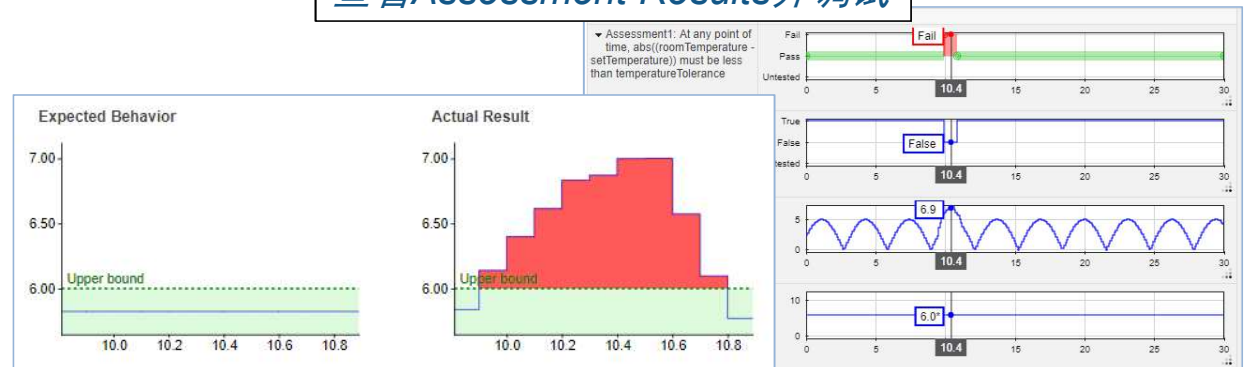
# 将文字的需求转换成非二义性的Temporal Assessments

- 使用表格编辑器创建 assessments
- 每个assessments 都是一个容易理解的句子
- 审查temporal assessment 结果并调试
- 和需求有追溯关系

Temporal Assessment 编辑器



查看Assessment Results并调试



# 跟踪开发和测试的状态

Requirements - crs\_controller

View: Requirements

Index	ID	Summary	Implemented	Verified
crs_req_func_spec	-	-		
1	#1	Driver Switch Request Handling		
1.1	#2	Switch precedence		
1.2	#3	Avoid repeating commands		
1.3	#4	Long Switch recognition		
1.4	#7	Cancel Switch Detection		
1.5	#8	Set Switch Detection		
1.6	#9	Enable Switch Detection		

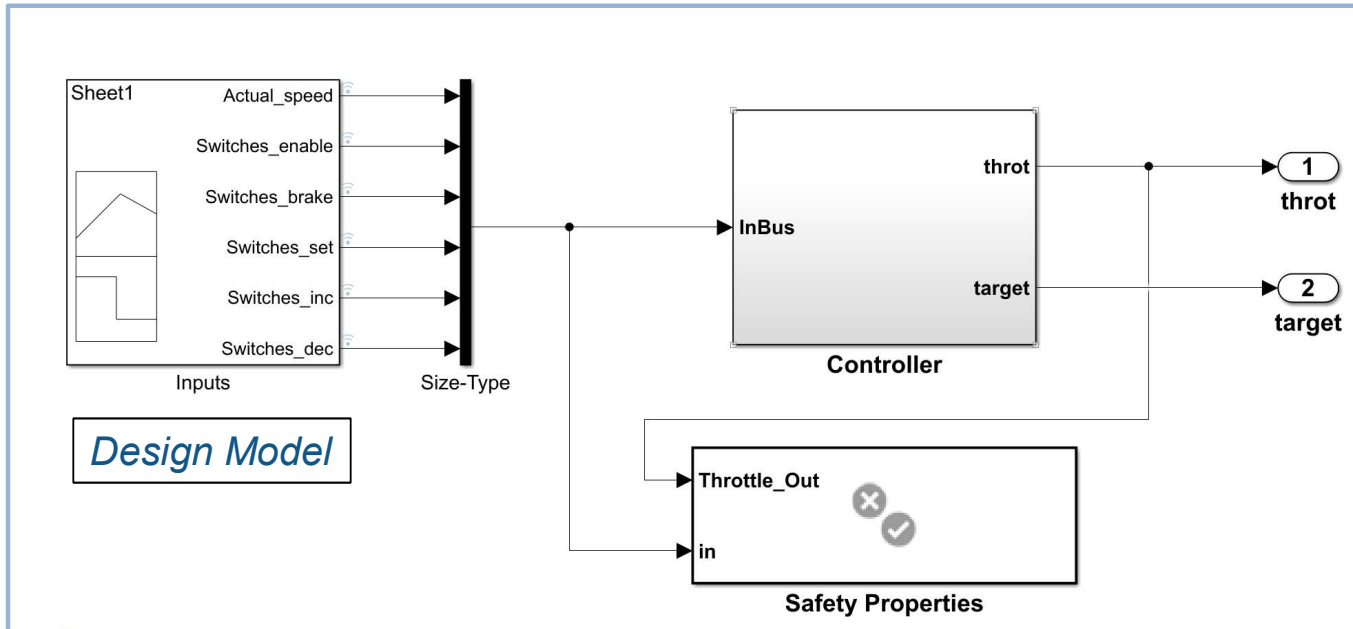
**Implementation Status**

- Implemented
- Justified
- Missing

**Verification Status**

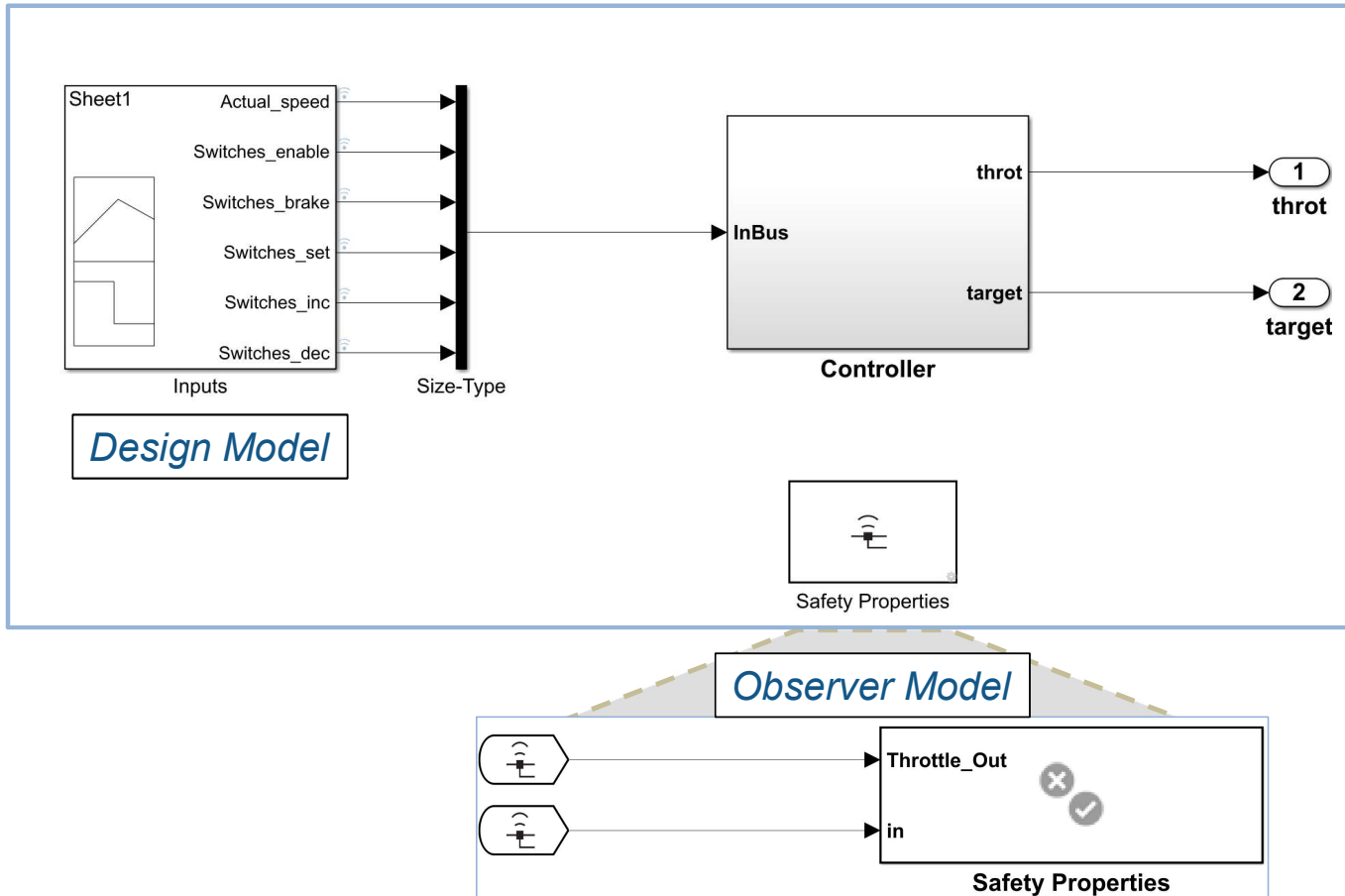
- Passed
- Failed
- Unexecuted
- Missing

## Observers: 将测试从开发中分离



- 不需要信号线或者改变动态响应就能访问深层嵌套重的信号量
- 避免为了测试更改接口
- 不增加额外的信号线，简化设计和测试。

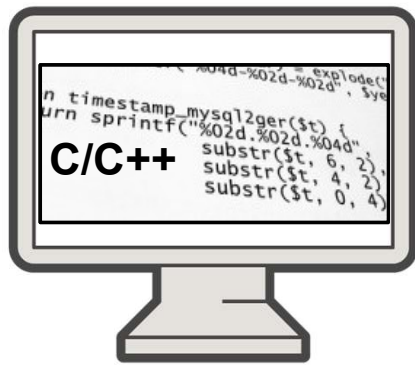
# Observers:将测试从开发中分离



- 不需要信号线或者改变动态响应就能访问深层嵌套重的信号量
- 避免为了测试更改接口
- 不增加额外的信号线，简化设计和测试。



# 重用模型的测试用例到代码的测试



**软件在环  
(SIL)**

- 功能一致性，模型到代码
- 桌面测试



**处理器在环  
(PIL)**

- 数值一致性 (Numerical equivalence)，模型到目标码
- 目标板测试



**硬件在环  
(HIL)**

- 检查设计和代码的实时性.
- 使用Simulink Real-Time在Speedgoat 目标计算机执行

# LS Automotive Reduces Development Time for Automotive Component Software with Model-Based Design

## Challenge

Shorten development times for embedded control software used in automotive switches and components

## Solution

Use Model-Based Design to model controller designs, run simulations, verify customer specifications, and generate error-free production code

## Results

- Specification errors detected early **早期发现设计缺陷**
- Proven development approach established
- **80% Coding errors eliminated**



An LS Automotive door area unit.

*“By enabling us to analyze requirements quickly, reuse designs from previous products, and eliminate manual coding errors, Model-Based Design has reduced development times and enabled us to shorten schedules to meet the needs of our customers.”*

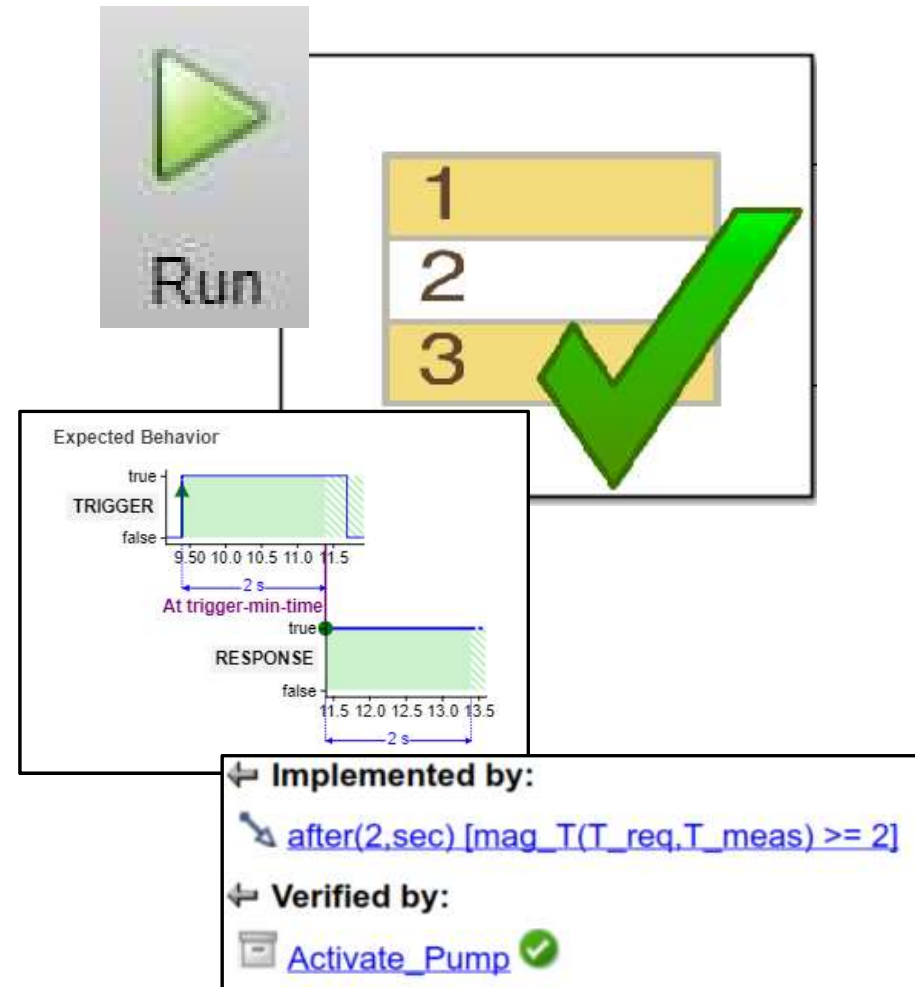
*- MyoungSuk Ko, LS Automotive*

[Link to user story](#)

MATLAB EXPO 2019

# 总结

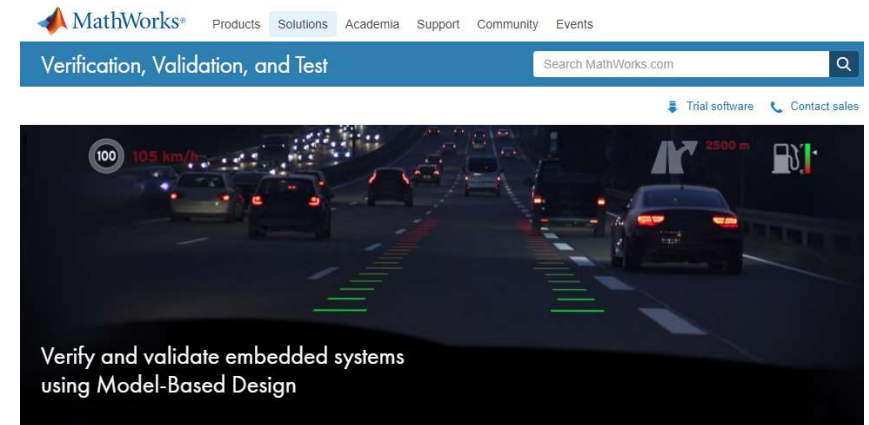
- Verify and validate requirements earlier  
早期介入
- Identify inconsistencies in requirements by using unambiguous assessments 发现需求的不一致性
- Traceability from requirements to design and test 追溯关系



## Learn More

Key products covered in this presentation:

- [Simulink Requirements](#)
- [Simulink Test](#)
- [Simulink Real-Time](#)



Learn more at Verification, Validation and Test Solution Page:

[mathworks.com/solutions/verification-validation.html](https://mathworks.com/solutions/verification-validation.html)

**Thanks!**