

MATLAB EXPO 2018

Automating Best Practices to
Improve Design Quality

Daniel Martins



Why do 71% of Embedded Projects Fail?

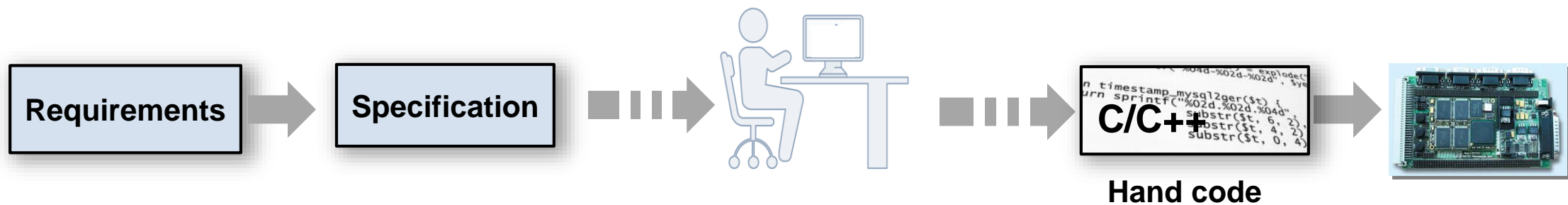
Poor Requirements Management

Sources: Christopher Lindquist, Fixing the Requirements Mess, CIO Magazine, Nov 2005

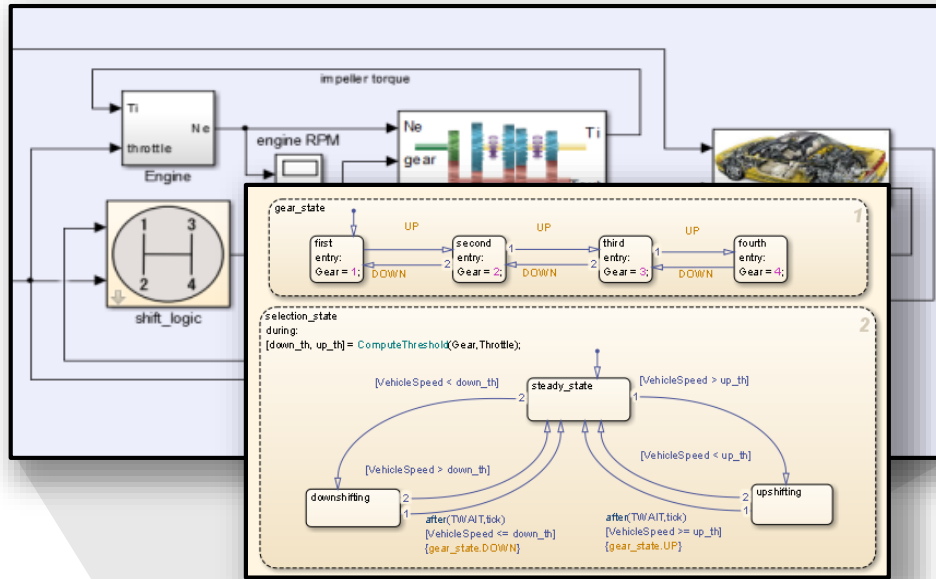
Key Takeaways

- Author, manage requirements in Simulink
- Early verification to find defects sooner
- Automate manual verification tasks
- Workflow that conforms to safety standards

Challenge with Traditional Development Process



Simulink Models for Specification



Requirements

Executable Specification



```

n timestamp mysql2ger($t) i
urn sprintf("%02d.%02d.%04d",
substr($t, 6, 2),
substr($t, 4, 2),
substr($t, 0, 4)

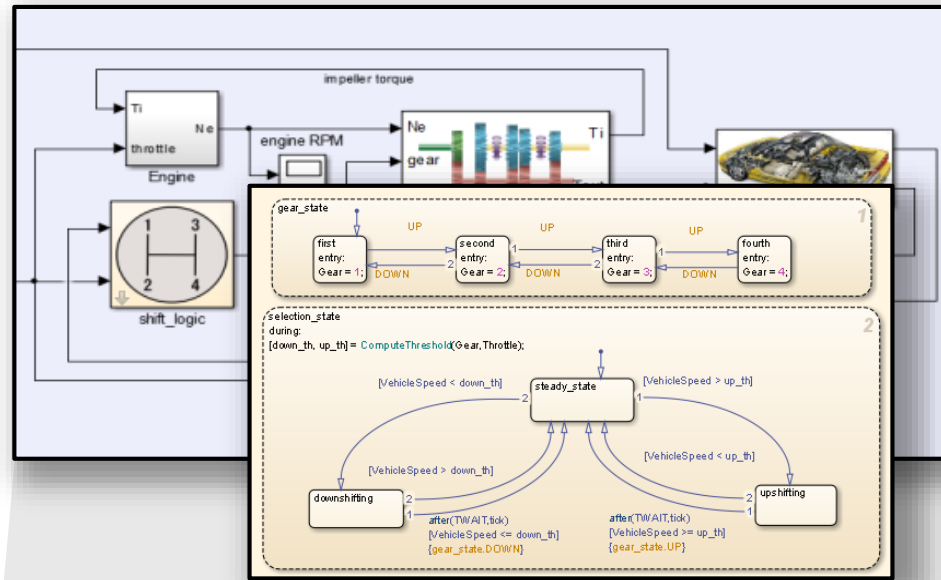
```

C/C++

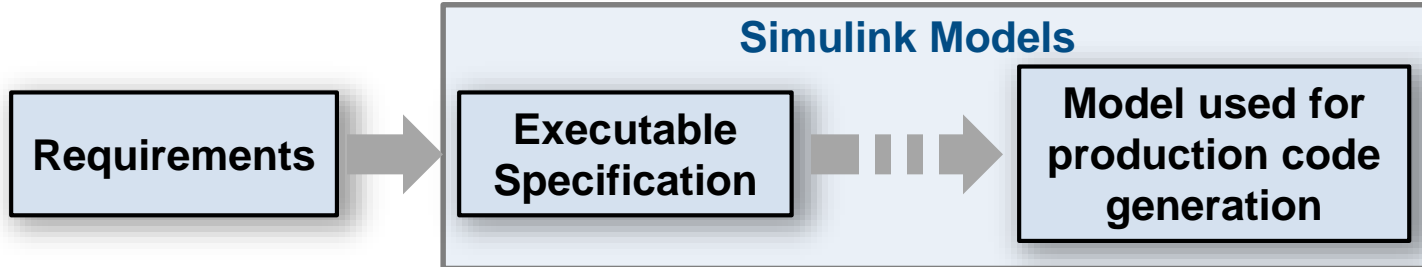
Hand code



Complete Model Based Design



Code Generation



Generated code

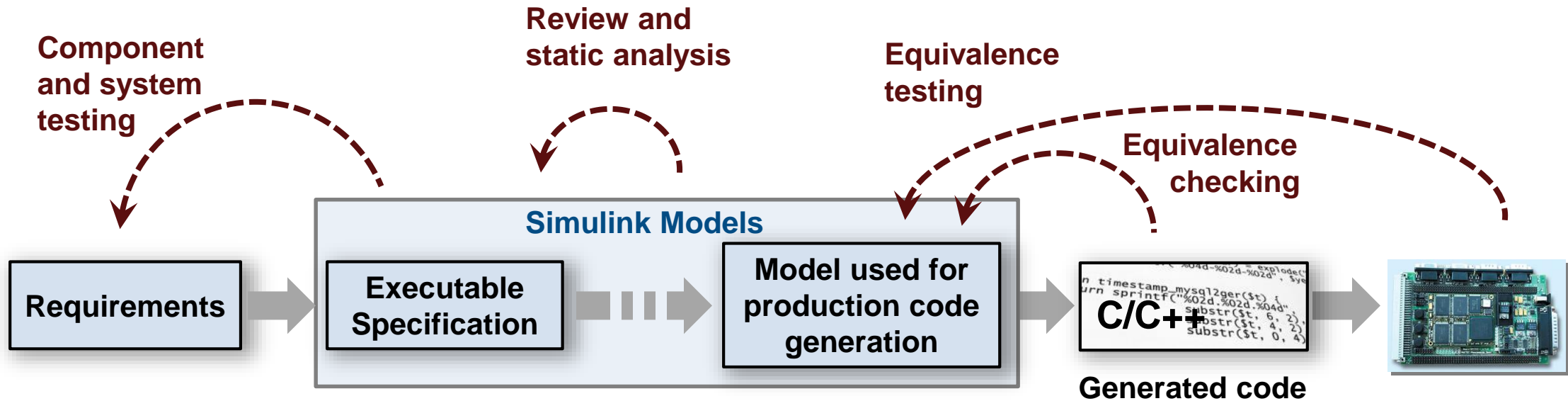
```

n timestamp mysql2ger($t) {
  urn sprintf("%02d.%02d.%04d",
    substr($t, 6, 2),
    substr($t, 4, 2),
    substr($t, 0, 4)
  )
}

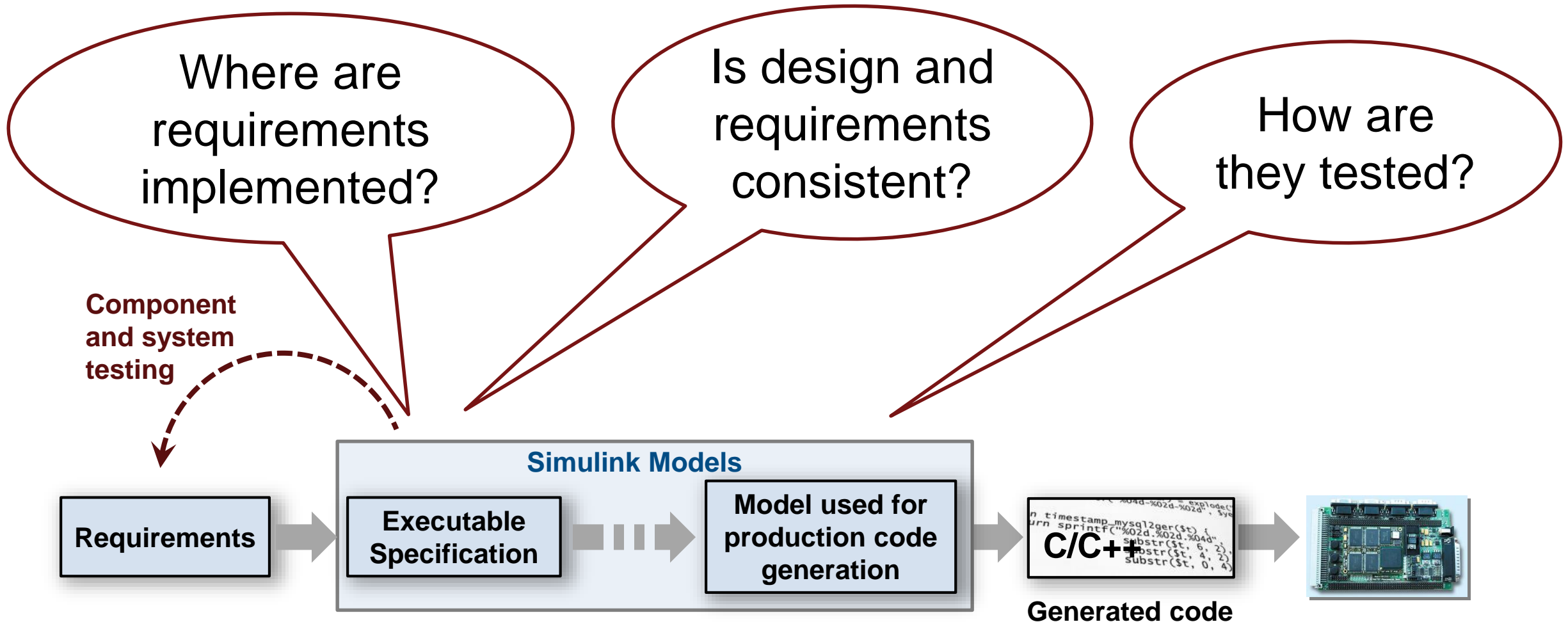
```



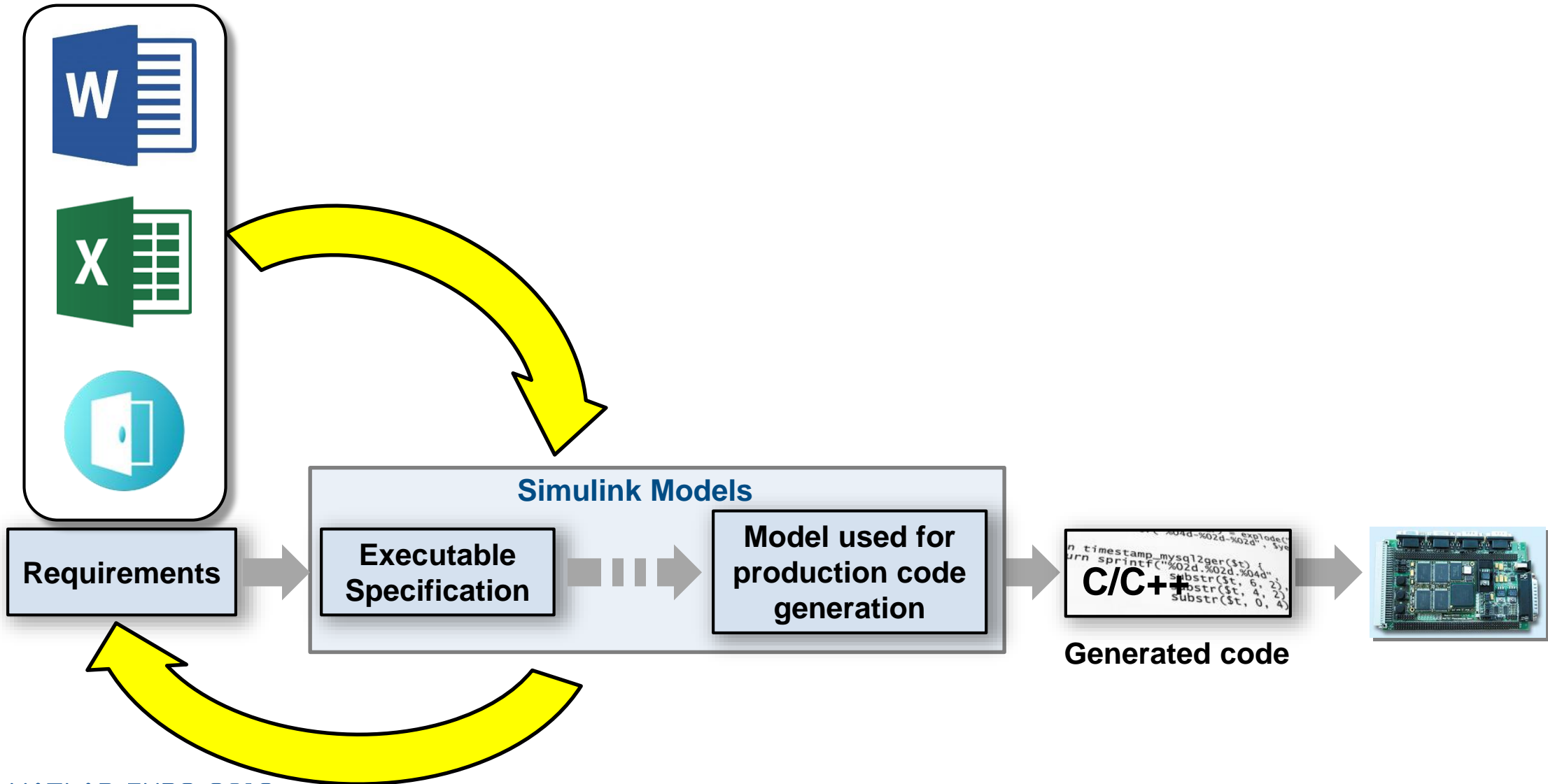
Model Based Design Verification Workflow



Challenges with Requirements



Gap Between Requirements and Design



Simulink Requirements

R2017b

Author


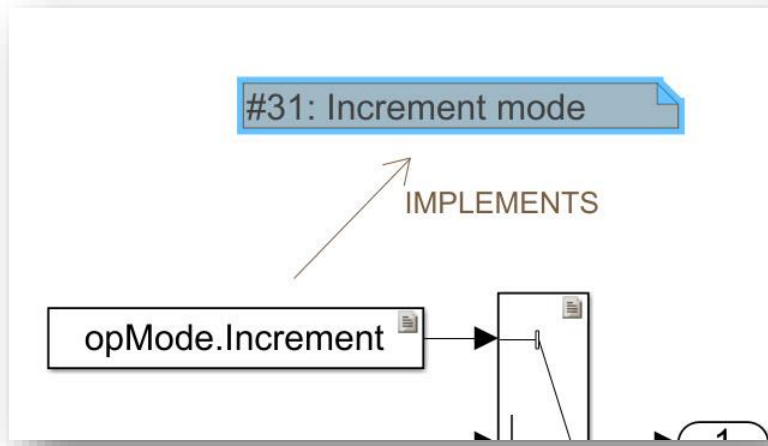
Summary: Cancel Switch Detection

Description Rationale

2 14 B I U [List Icon] [List Icon] [List Icon] [List Icon] ... >>

If the Cancel switch is pressed, the value of *reqDrv* should be set to *reqMode.Cancel*.

Dashboard image

Track

Manage

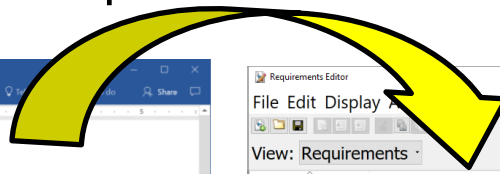
⚠ Issue: Destination Changed.

Stored:	Revision: 15
Actual:	Revision: 18

Clear Issue

Import Requirements from External Sources

Import



IBM Rational DOORS

ReqIF
Requirements Interchange Format

Microsoft Word

3 - FUNCTIONAL-REQUIREMENTS

3.1 - ENABLING-CRUISE-CONTROL

Cruise-control-is-enabled-when-the-following-conditions-are-met:

- Vehicle-speed-is-within-the-target-speed-range-(40km/h—100km/h).
- Key-position-is-ON.
- Gear-position-is-Drive.
- Cruise-button-is-pushed-while-the-cruise-control-mode-is-disabled.

Dashboard-image

3.2 - DISABLING-CRUISE-CONTROL

Cruise-control-is-disabled-when-one-or-more-of-the-following-are-met:

- Key-position-is-set-to-any-other-position-than-ON.
- When-the-vehicle-is-started.-Cruise-button-is-pushed-while-the-cruise-control-enabled-or-activated.
- Gear-position-is-not-Drive

Dashboard-image

R2018a

Simulink Requirements Editor

Requirements Editor

View: Requirements

Index	ID	Summary
crs_req		
1	crs_req	References to crs_req.docx
1.1	1 Overview	Overview This document describes a r
1.2	2 System overview	System overview
1.2.1	2.1 System inputs	System inputs
1.2.1.1	2.1.1 Cruise control buttons	Cruise control buttons Five buttons are
1.2.1.2	2.1.2 Other inputs	Other inputs Current vehicle speed Th
1.2.2	2.2 Cruise control mode indi...	Cruise control mode indicator Two indi
1.2.3	2.3 Cruise control modes	Cruise control modes There are three r
1.3	3 Functional Requirements	Functional Requirements
1.3.1	3.1 Enabling cruise control	Enabling cruise control Cruise control i
1.3.2	3.2 Disabling cruise control	Disabling cruise control Cruise control
1.3.3	3.3 Activating cruise control	Activating cruise control Cruise control
1.3.4	3.4 Deactivating cruise control	Deactivating cruise control Cruise cont
1.3.5	3.5 Target Speed Increment	Target Speed Increment While the cru
1.3.6	3.6 Target speed decrement	Target speed decrement While the cru
1.3.7	3.7 Successive Target Speed...	Successive Target Speed Increment W
1.3.8	3.8 Successive Target Speed...	Successive Target Speed Decrement W
1.3.9	3.9 Adjusting Target Speed ...	Adjusting Target Speed with Accelerat
1.3.10	3.10 Resuming cruise control	Resuming cruise control Cruise control
1.3.11	3.11 Throttle value calculation	Throttle value calculation The cruise c
1.3.12	3.12 Cruise Control SET Indi...	Cruise Control SET Indicator Light Cru
1.4	4 Interface specification	Interface specification

Properties

Index: 1.3.1
 Custom ID: 3.1 Enabling cruise control
 Summary: Enabling cruise control Cruise control is enabled when the following condi...

Description Rationale

3.1 Enabling cruise control

Cruise control is enabled when the following conditions are met:

- Vehicle speed is within the target speed range (40km/h – 100km/h).
- Key position is ON.
- Gear position is Drive.
- Cruise button is pushed while the cruise control mode is disabled.

Dashboard image

Keywords:

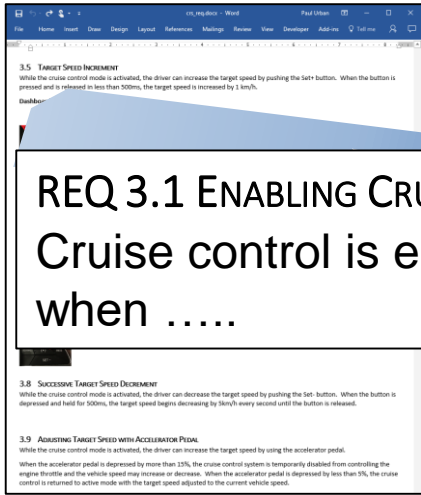
Revision information:

Links

Show in document

Show in document

Link Requirements, Designs and Tests



REQ 3.1 ENABLING CRUISE CONTROL
Cruise control is enabled when

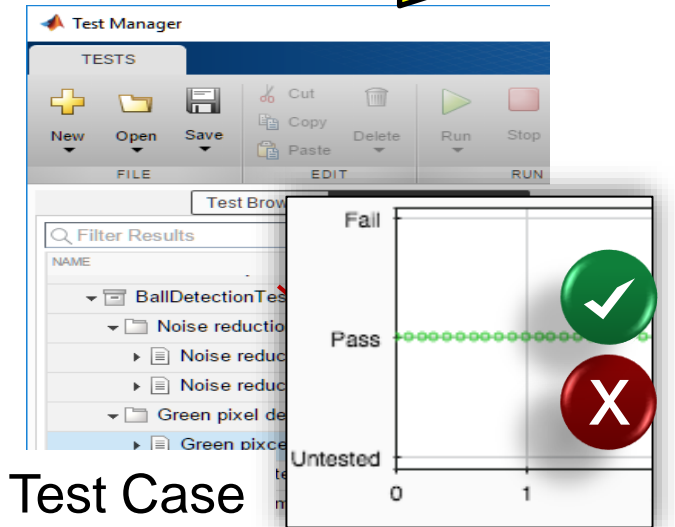
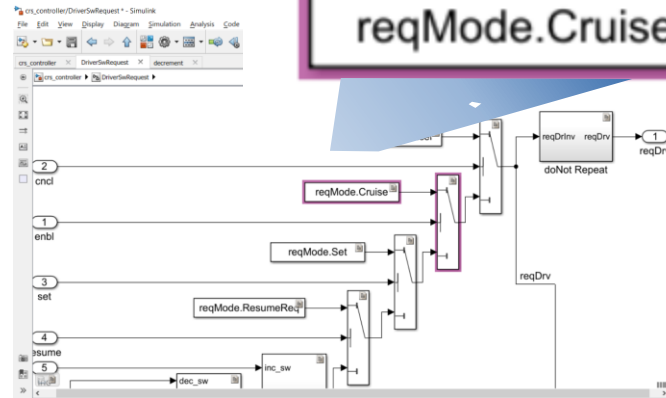
Derives

ENABLE SWITCH DETECTION
If the Enable switch is pressed

Implemented
By

Verified
By

reqMode.Cruise



Track Implementation and Verification

Requirements - crs_controller

View: Requirements

Index	ID	Summary	Implemented	Verified
crs_req_func_spec*	—	—		
> 1	#1	Driver Switch Request Handling		
> 2	#19	Cruise Control Mode		
> 2.1	#20	Disable Cruise Control system		
> 2.2	#24	Operation mode determination		

Ready

Implementation Status

- Implemented
- Justified
- Missing

Verification Status

- Passed
- Failed
- No Result
- Missing

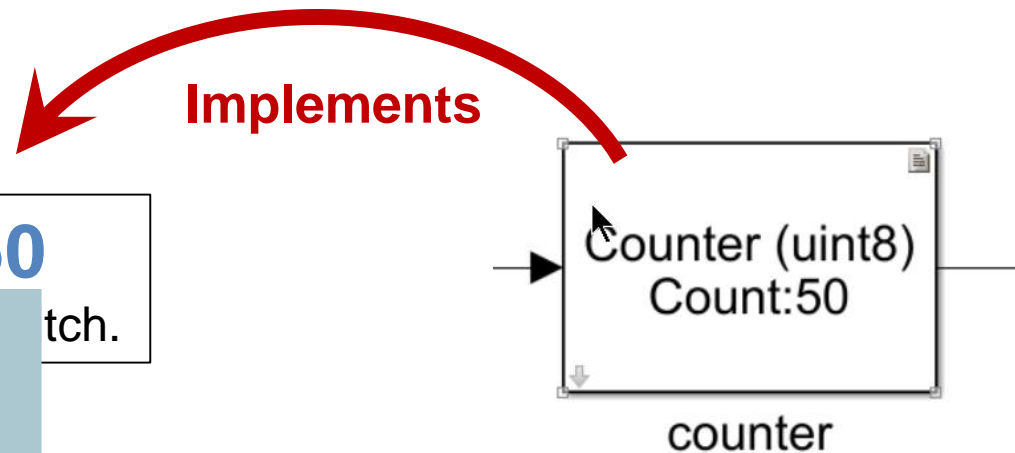
Respond to Change

Original Requirement

If the switch is pressed and the counter reaches **50** then it shall be recognized as a long press of the switch.

Updated Requirement

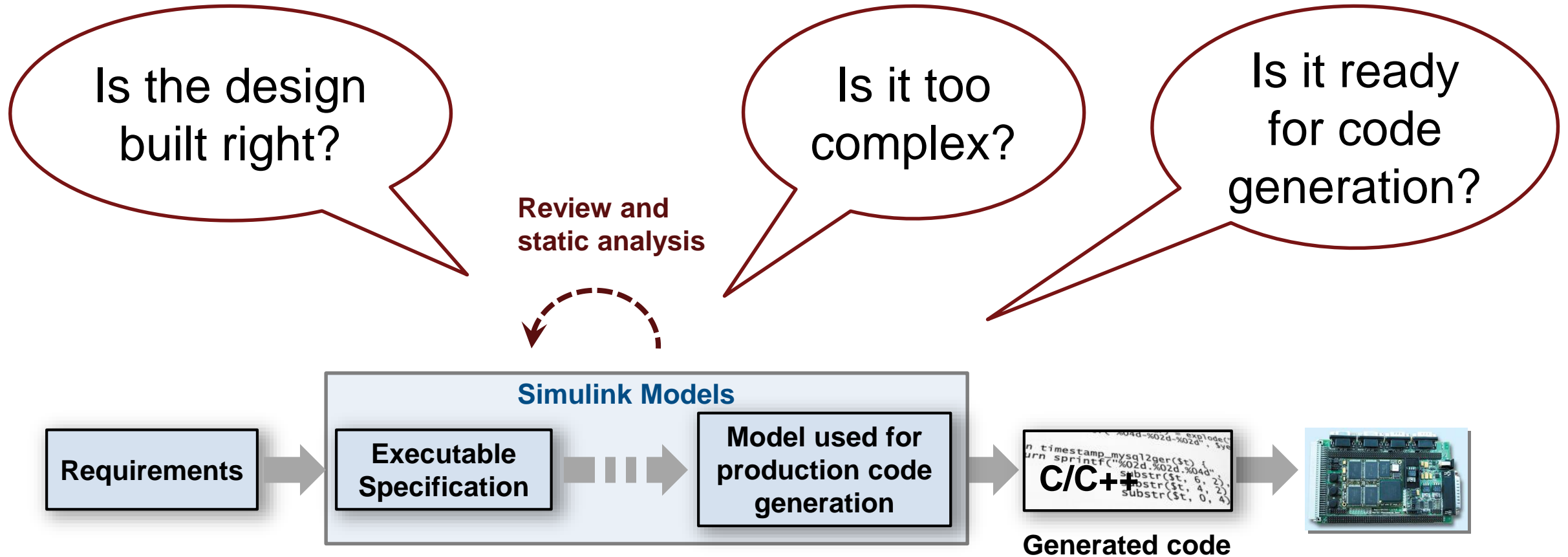
If the switch is pressed and the counter reaches **75** then it shall be recognized as a long press of the switch.



← **Implemented by:**
counter

 Issue: Destination Changed.

Verify Design to Guidelines and Standards



Automate verification with static analysis

Model Advisor Analysis

Check for blocks not recommended for C/C++ production code deployment

Analysis
Identify blocks not supported by code generation or not recommended for C/C++ production code deployment.

Run This Check

Result: **Warning**
Identify blocks not supported by code generation or not recommended for C/C++ production code deployment.

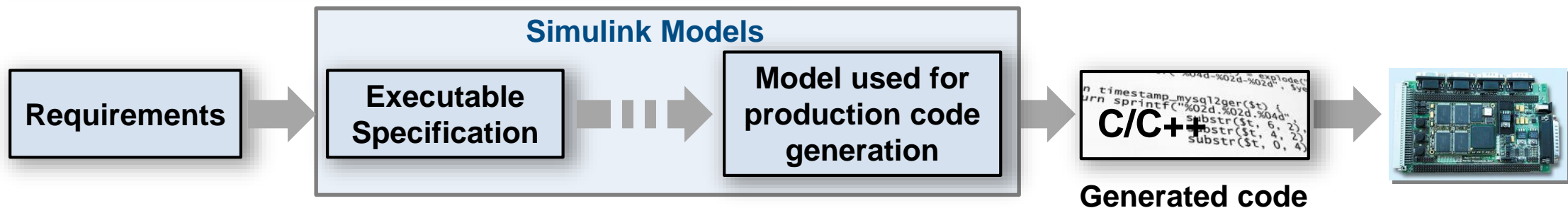
Warning
The following blocks are not supported or not recommended for C/C++ production code deployment:

Block	Block Type	Code generation support	Recommendation for C/C++ production code deployment
.../Intake Manifold/p0 = 0.589 bar	Integrator	Yes ^{1,2}	No
sldemo_fuelsys/Throttle Command	Repeating table	Yes ³	No

Recommended Action
Although Embedded Coder supports these blocks, they are not recommended for C/C++ production code deployment. Review the support notes for these blocks and follow the given advice.

Check for:

- Readability and Semantics
- Performance and Efficiency
- Clones
- And more.....



Generate reports for reviews and documentation

Model Advisor Analysis

Check for blocks not recommended for C/C++ production code deployment

Analysis
Identify blocks not supported by code generation or not recommended for C/C++ production code deployment.

Result: **Warning**
Identify blocks not supported by code generation or not recommended for C/C++ production code deployment.

Warning
The following blocks are not supported or not recommended for C/C++ production code deployment:

Block	Block Type	Code generation support	Recommendation for C/C++ production code deployment
.../Intake Manifold/p0 = 0.589 bar	Integrator	Yes ^{1,2}	No
sldemo_fuelsys/Throttle Command	Repeating table	Yes ³	No

Recommended Action
Although Embedded Coder supports these blocks, they are not recommended for C/C++ production code deployment. Review the support notes for these blocks and follow the given advice.

Model Advisor Reports

Simulink version: 9.1
System: sldemo_fuelsys
Treat as Referenced Model: off

Model version: 1.749
Current run: 11-Mar-2018 13:31:16

Run Summary

Pass	Fail	Warning	Not Run	Total
203	0	215	196	614

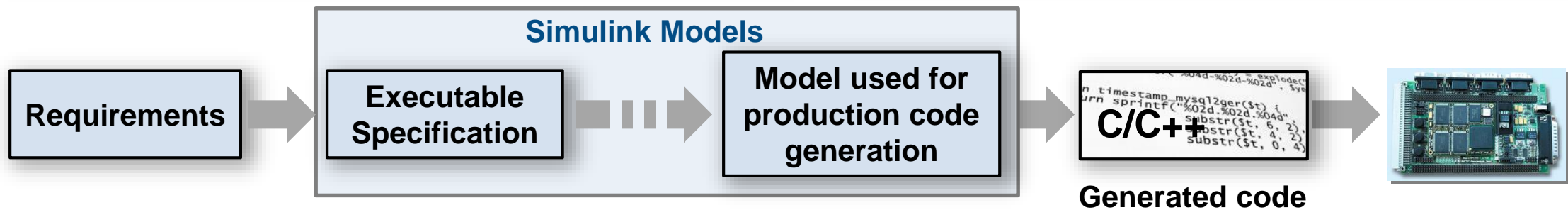
By Task

- 1 Code Generation Efficiency 3 0 3 3

Check optimization settings
Check for optimizations that can lead to non-optimal code generation and simulation.

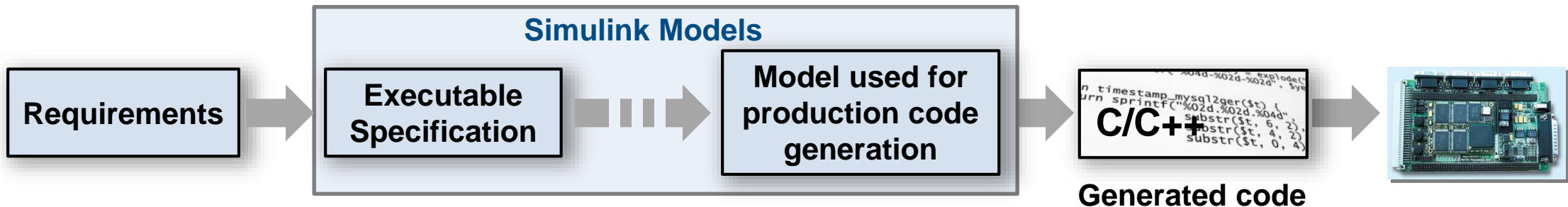
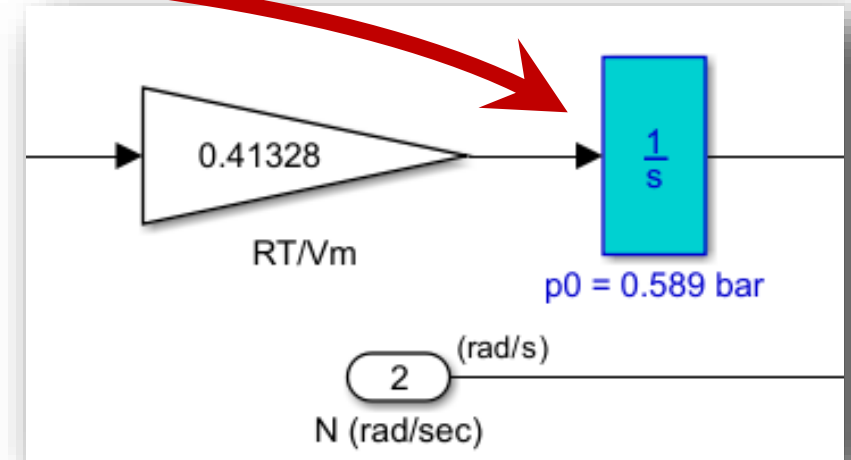
Warning

Parameter	Current Value	Recommended Values
Use bitsets for storing state configuration (StateBitsets)	off	on
Use bitsets for storing Boolean data (DataBitsets)	off	on



Navigate to Problematic Blocks

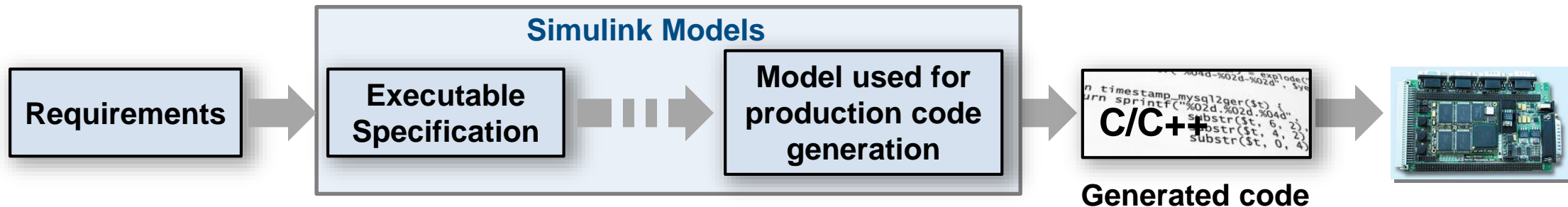
Block	Block Type	Code generation support	Recommendation for C/C++ production code deployment
.../Intake Manifold/p0 = 0.589 bar	Integrator	Yes ^{1, 2}	No
sldemo_fuelsys/Throttle Command	Repeating table	Yes ³	No



Guidance Provided to Address Issues or Automatically Correct

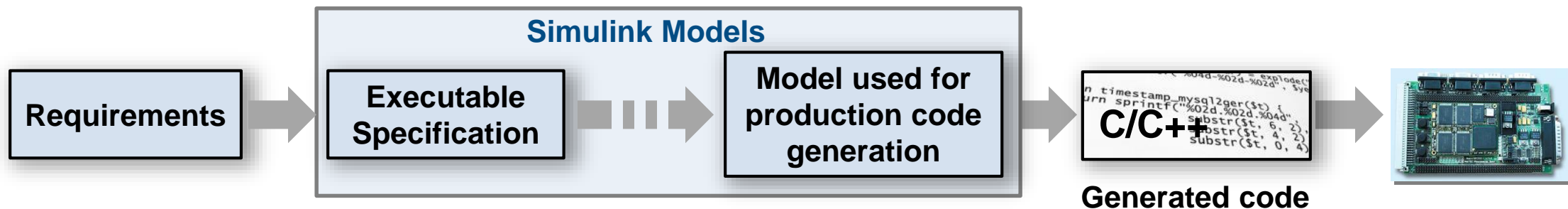
Recommended Action

Although Embedded Coder supports these blocks, they are not recommended for C/C++ production code deployment. Review the support notes for these blocks and follow the given advice.

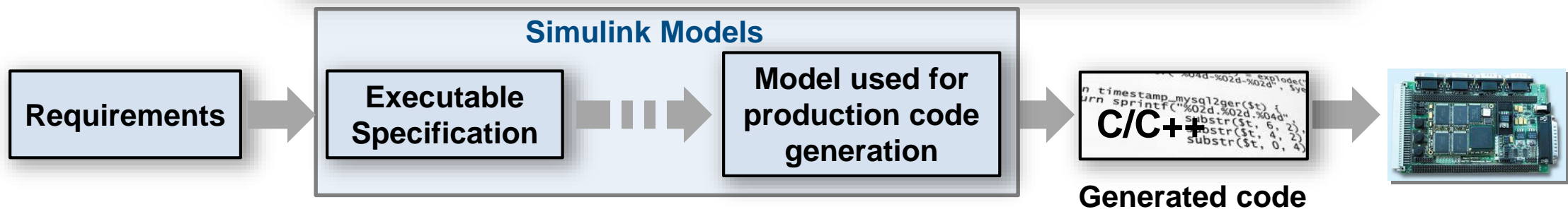


Built in checks for industry standards and guidelines

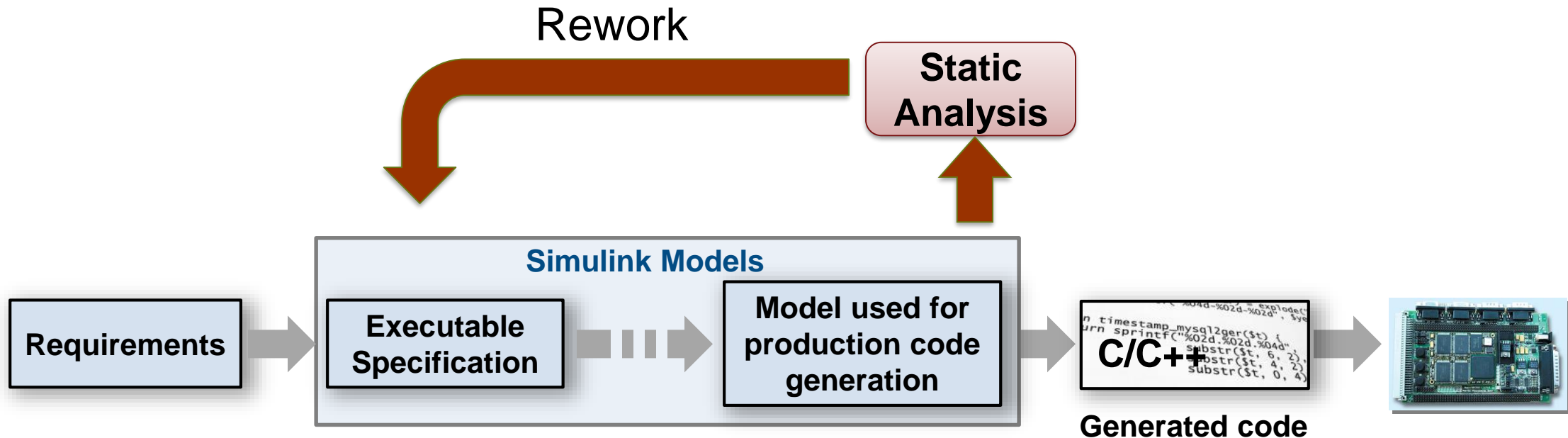
- DO-178/DO-331
- MISRA C:2012
- ISO 26262
- CERT C, CWE, ISO/IEC TS 17961
- IEC 61508
- MAAB (MathWorks Automotive Advisory Board)
- IEC 62304
- JMAAB (Japan MATLAB Automotive Advisory Board)
- EN 50128



Configure and customize analysis

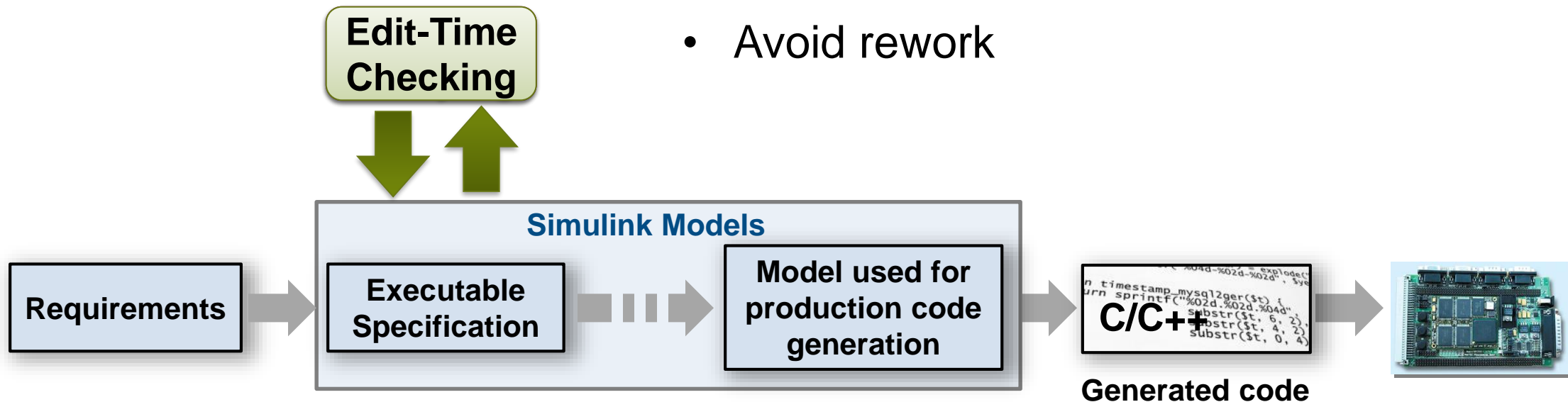


Checks for standards and guidelines are often performed late

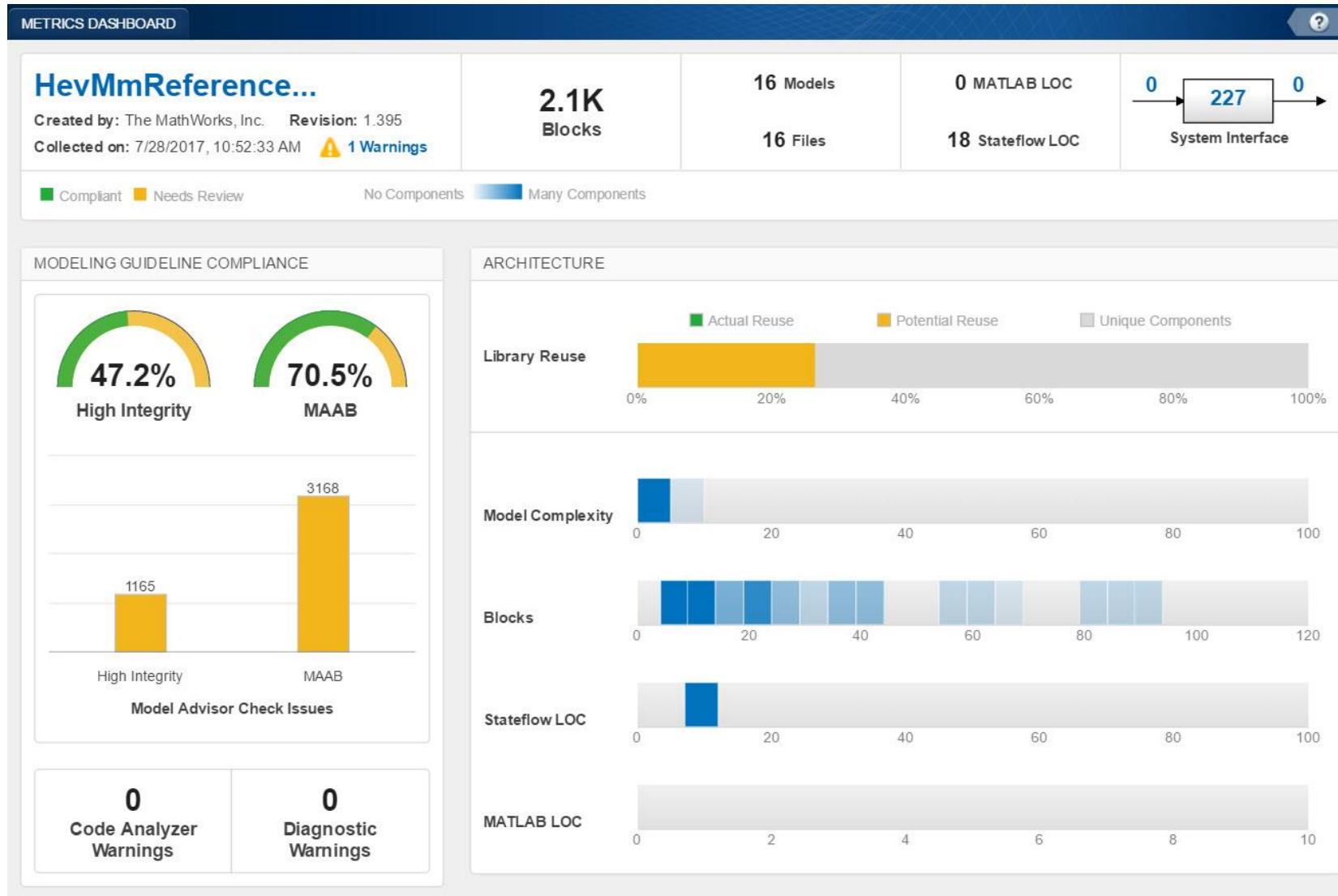


Shift Verification Earlier With Edit-Time Checking

- Highlight violations as you edit
- Fix issues earlier
- Avoid rework



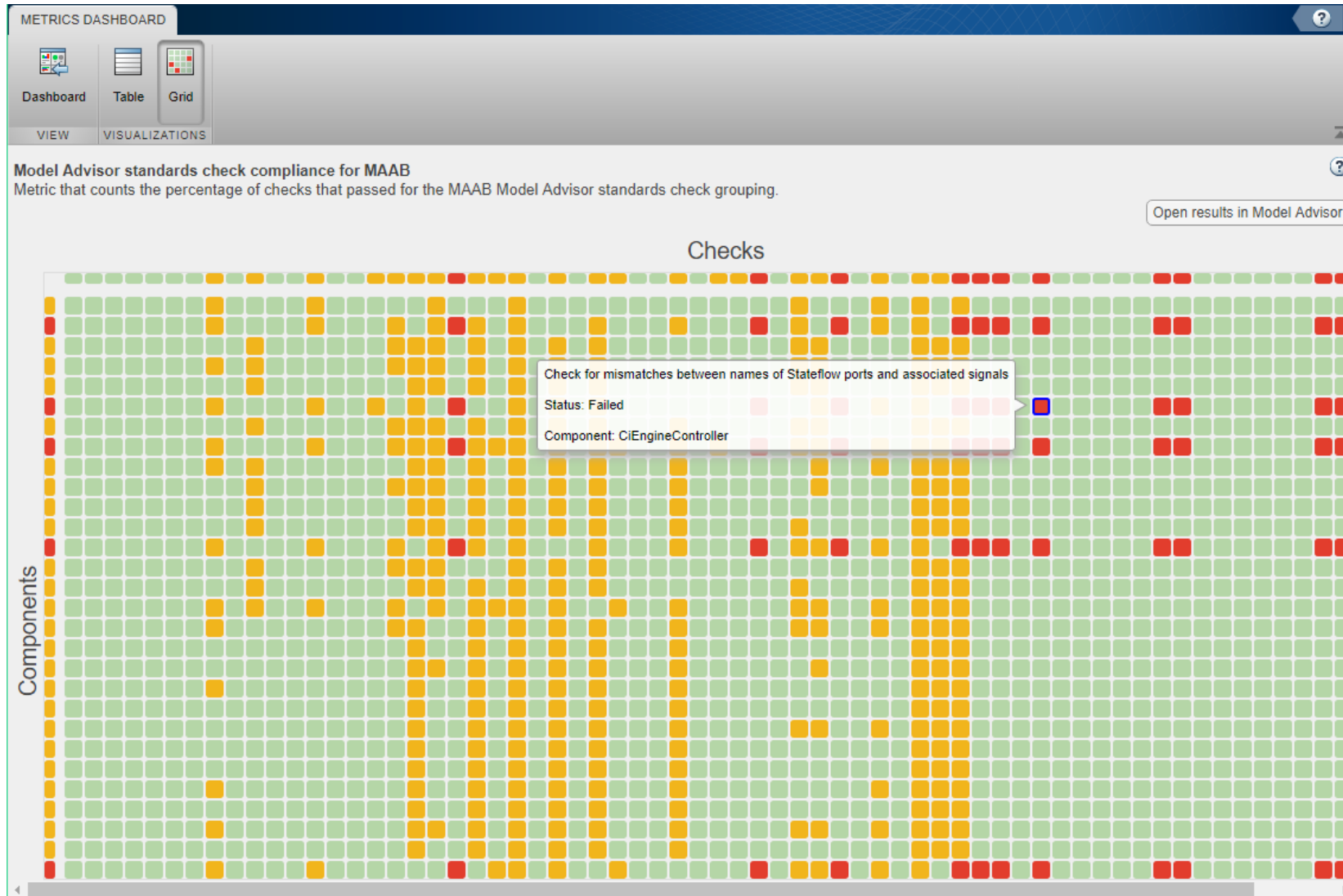
Assess Quality with Metrics Dashboard



- Consolidated view of metrics
 - Size
 - Compliance
 - Complexity
- Identify where problem areas may be

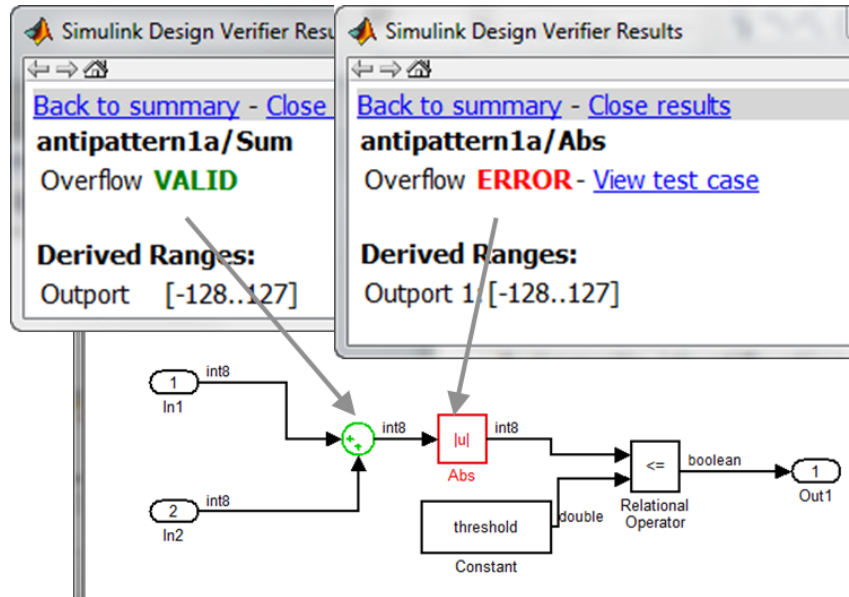
Grid Visualization for Metrics

R2018a



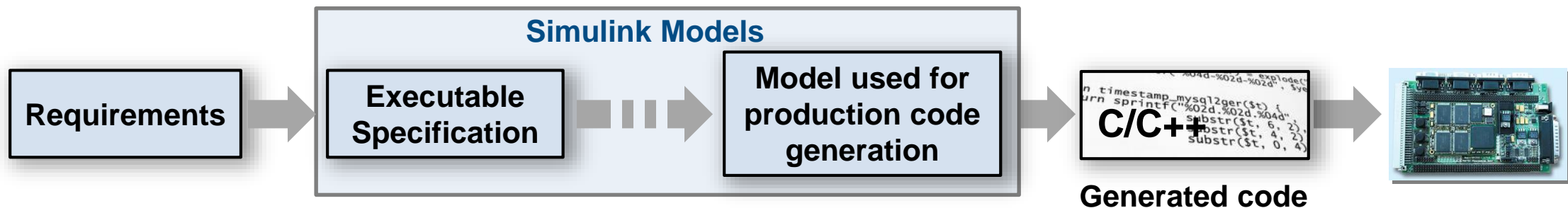
- Visualize Standards Check Compliance
 - Find Issues
 - Identify patterns
 - See hot spots

Detect Design Errors with Formal Methods

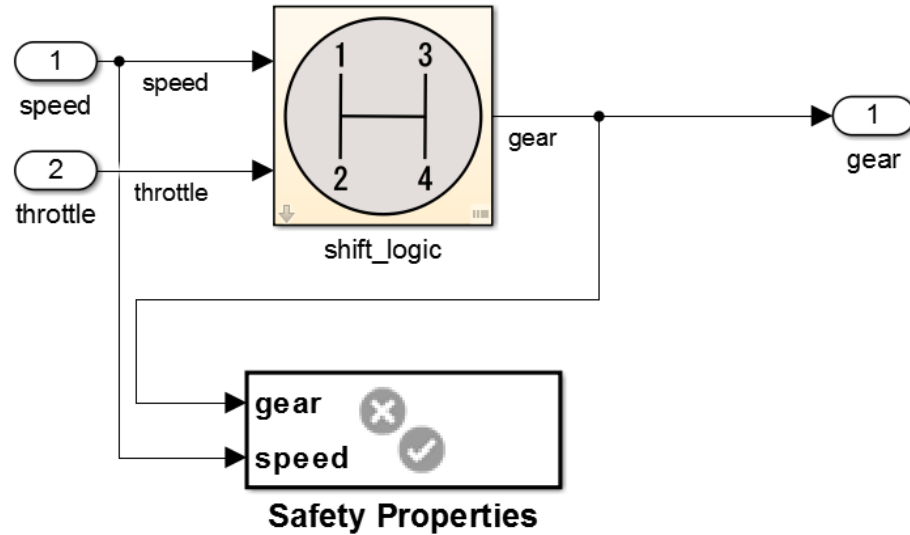


- Find run-time design errors:
 - Integer overflow
 - Dead Logic
 - Division by zero
 - Array out-of-bounds
 - Range violations

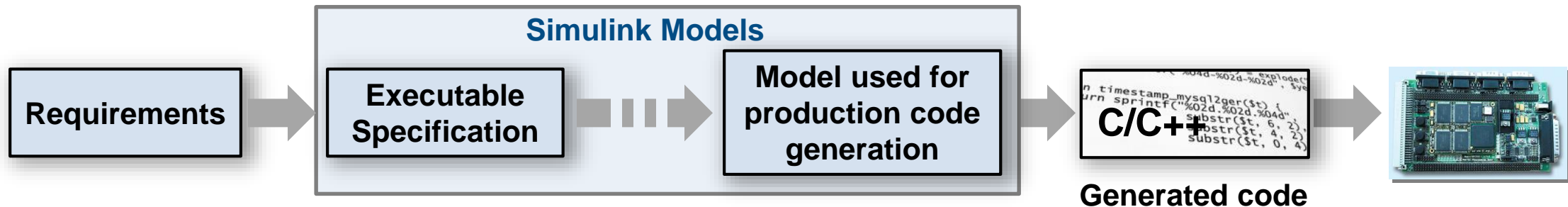
- Generate counter example to reproduce error



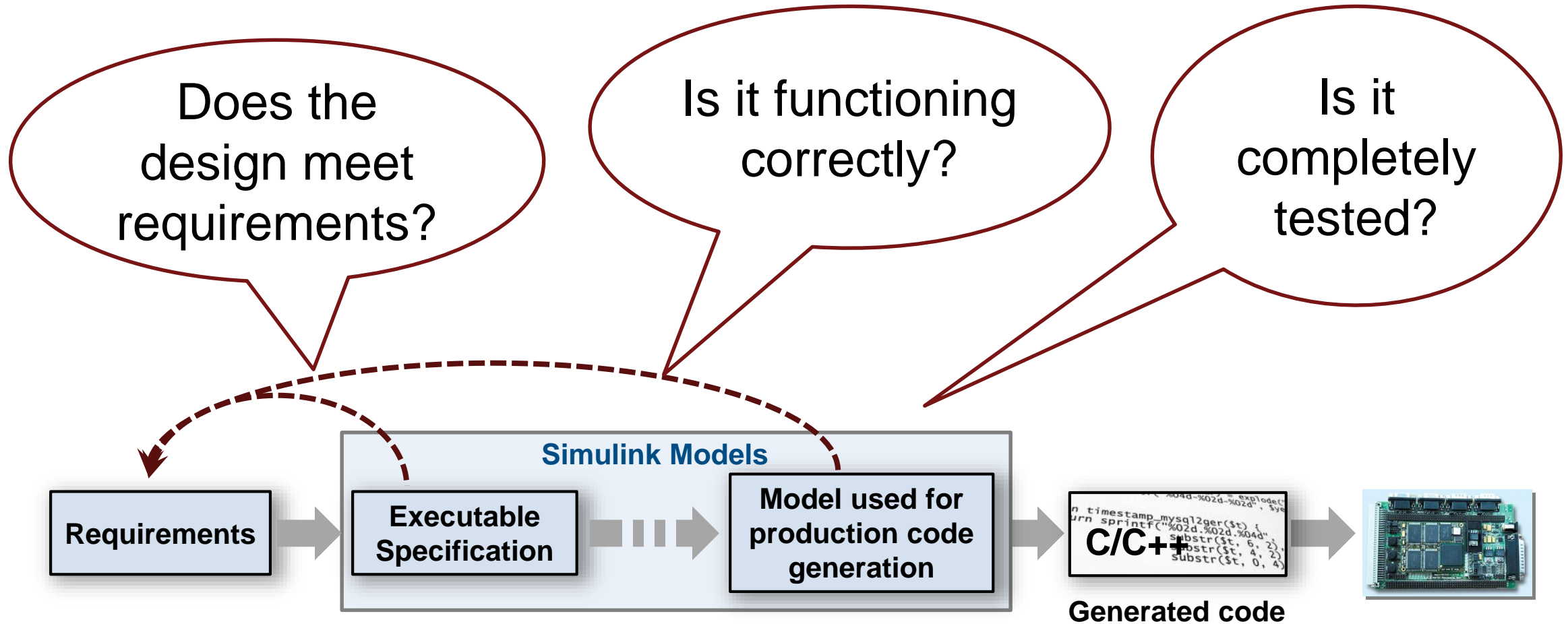
Prove That Design Meets Requirements



- Prove design properties using formal requirement models
- Model functional and safety requirements
- Generates counter example for analysis and debugging



Functional Testing



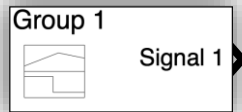
Systematic Functional Testing

Test Case

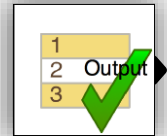
Inputs



MAT file (input)



Signal Builder

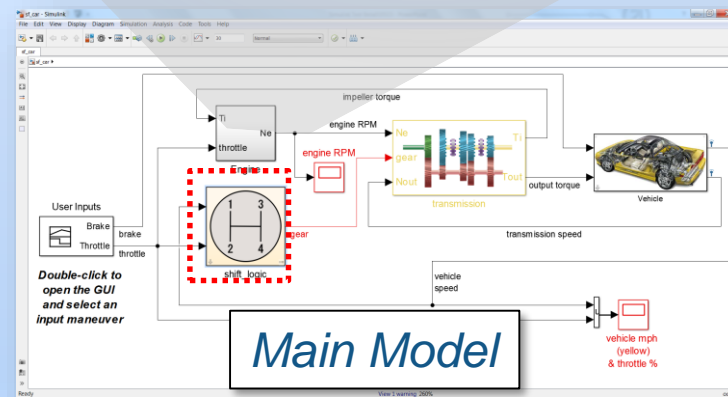
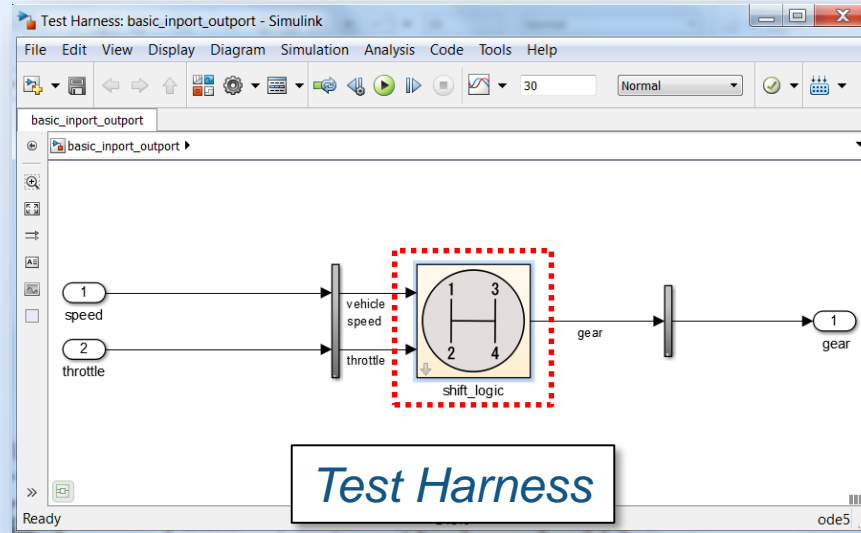


Test Sequence



Excel file (input)

R2017b



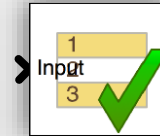
Assessments



MAT file (baseline)

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

MATLAB Unit Test



Test Assessment



Excel file (baseline)

R2017b

Manage Testing and Test Results

Test Manager

TESTS

File Edit Run Results Resources

Test Browser Results and Artifacts

Start Page x Slow Accel x

Filter Tests

- ComponentTesting
 - General Performance Test
 - Functional and Regression tests
 - Signal Builder Baseline examples
 - Slow Accel
 - Fast Accel
 - Decel
 - ExcelDrivenExamples
 - Software-in-the-loop Testing
 - SystemTesting
 - ExampleBaselineTesting

Slow Accel

ComponentTesting > Functional and Regression tests > Signal Builder Baseline examples > Slow Accel

Baseline Test

DESCRIPTION

REQUIREMENTS

SYSTEM UNDER TEST

PARAMETER OVERRIDES

CALLBACKS

INPUTS

OUTPUTS

CONFIGURATION SETTINGS OVERRIDES

BASELINE CRITERIA

SIGNAL NAME	ABS TOL	REL TOL
SlowAccelbaselineCheckpoint1.mat	0	0.00 %

PROPERTY VALUE

Name	Slow Accel
Type	Baseline Test
Location	C:\Users\monelli\Desktop...
Enabled	<input checked="" type="checkbox"/>
Hierarchy	ComponentTesting > Fu...
Model	st_car
Simulation Mode	[Model Settings]
Harness Name	SigBdriven

Test Manager

TESTS VISUALIZE FORMAT

Clear Plot Data Cursors Highlight in Model Send to Figure

EDIT ZOOM & PAN MEASURE & TRACE SHARE

Test Browser Results and Artifacts

Start Page x Slow Accel x Comparison x

Filter Results

NAME	STATUS
Results : 2015-Jan-12 17:35:31	2 <input checked="" type="checkbox"/> 1 <input checked="" type="checkbox"/>
Signal Builder Baseline examples	2 <input checked="" type="checkbox"/> 1 <input checked="" type="checkbox"/>
Slow Accel	<input checked="" type="checkbox"/>
Fast Accel	<input checked="" type="checkbox"/>
Baseline Criteria Result	<input checked="" type="checkbox"/>
gear	<input checked="" type="checkbox"/>
throttle	<input checked="" type="checkbox"/>
vehicle speed	<input checked="" type="checkbox"/>
Sim Output (sf_car : normal)	<input checked="" type="checkbox"/>
Decel	<input checked="" type="checkbox"/>

PROPERTY VALUE

Name	gear
Status	<input checked="" type="checkbox"/>
Absolute Tolerance	0
Relative Tolerance	0.00 %
Block Path	SigBdriven/shift_logic

Comparison

Baseline Compare To

fourth
third
second
first
None

0 2 4 6 8 10 12 14 16 18 20 22 24 26 28 30

Tolerance Difference

1.0
0.8
0.6
0.4
0.2
0

0 2 4 6 8 10 12 14 16 18 20 22 24 26 28 30

Coverage Analysis to Measure Testing

- Identify testing gaps
- Missing requirements
- Unintended Functionality
- Design Errors

Simulink

Stateflow

Generated Code

```

46 /*
47 */
48 rtb_inputGElower = (rtb_input >= slvndemo_counter_U.lower);
49
50 /* Switch: '<Root>/Switch' incorporates:
51 * Import: '<Root>/upper'

```

Coverage: sf_car

Transition "UP" from "third" UP was never true.

[speed < up_th]

Decisions analyzed:

!((slvndemo_counter_U.upper >= rtb_input) && rtb_inputGElower)	50%
false	51/51
true	0/51

Conditions analyzed:

Description:	True	False
slvndemo_counter_U.upper >= rtb_input	51	0
rtb_inputGElower	51	0

MC/DC analysis (combinations in parentheses did not occur)

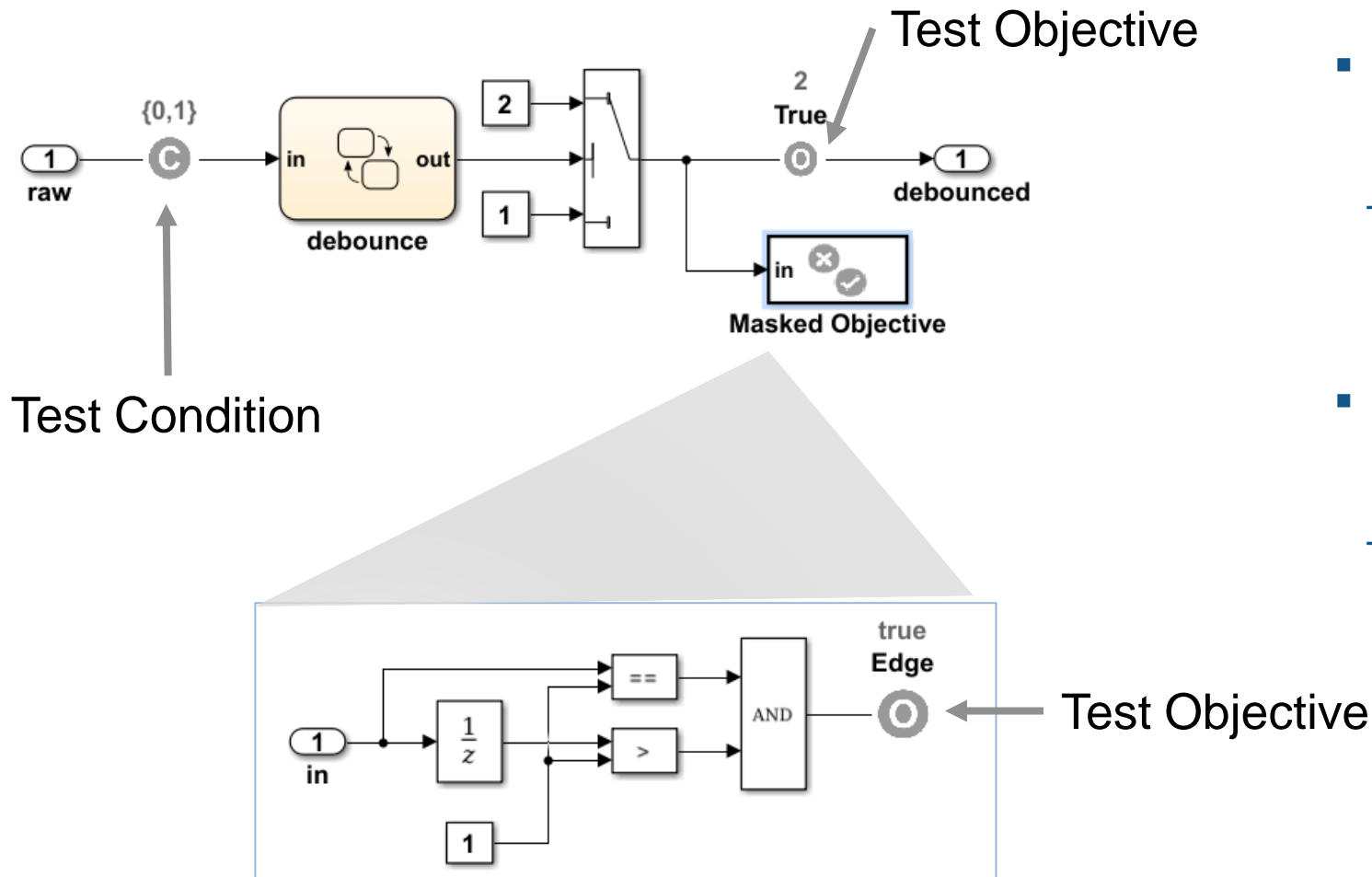
decision outcomes:	True	False
	Out	Out

Summary

Coverage Reports

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%

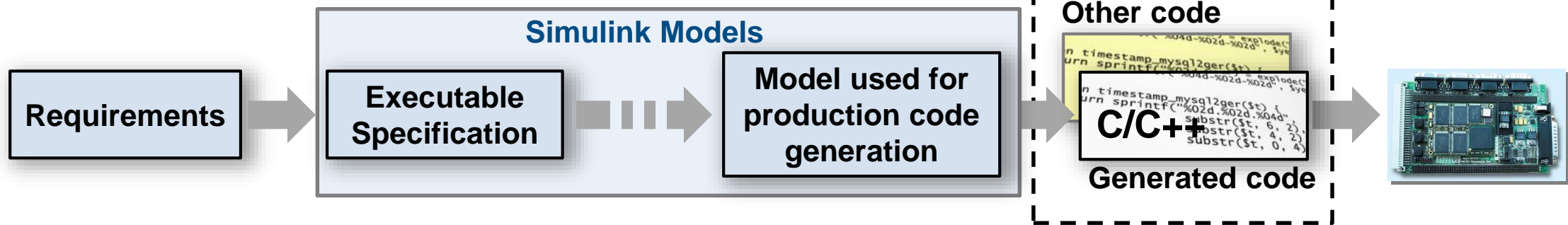
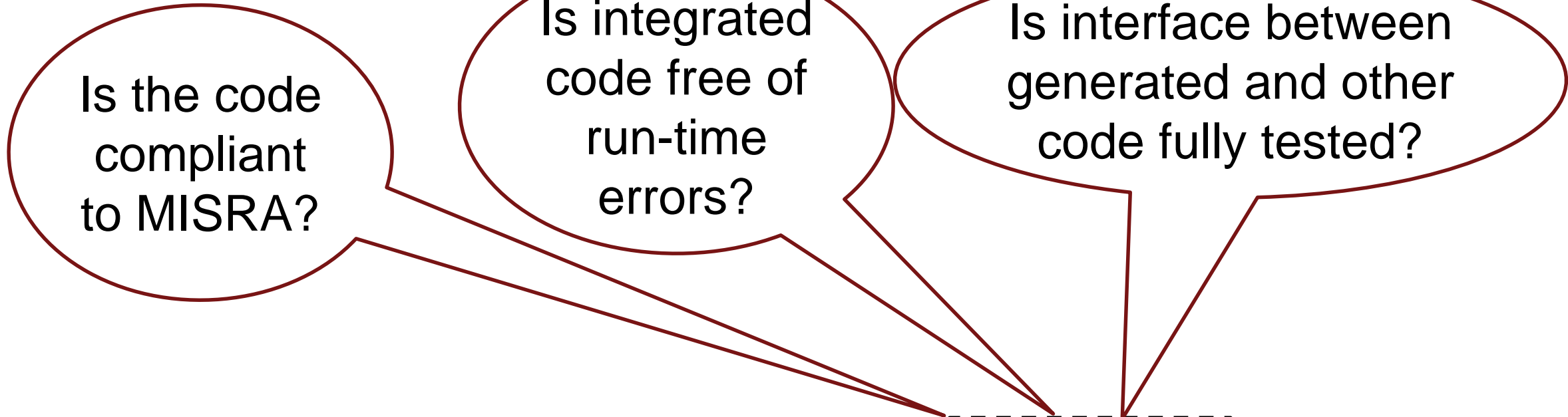
Test Case Generation for Functional Testing



- Specify functional test objectives
 - Define custom objectives that signals must satisfy in test cases

- Specify functional test conditions
 - Define constraints on signal values to constrain test generator

Static Code Analysis

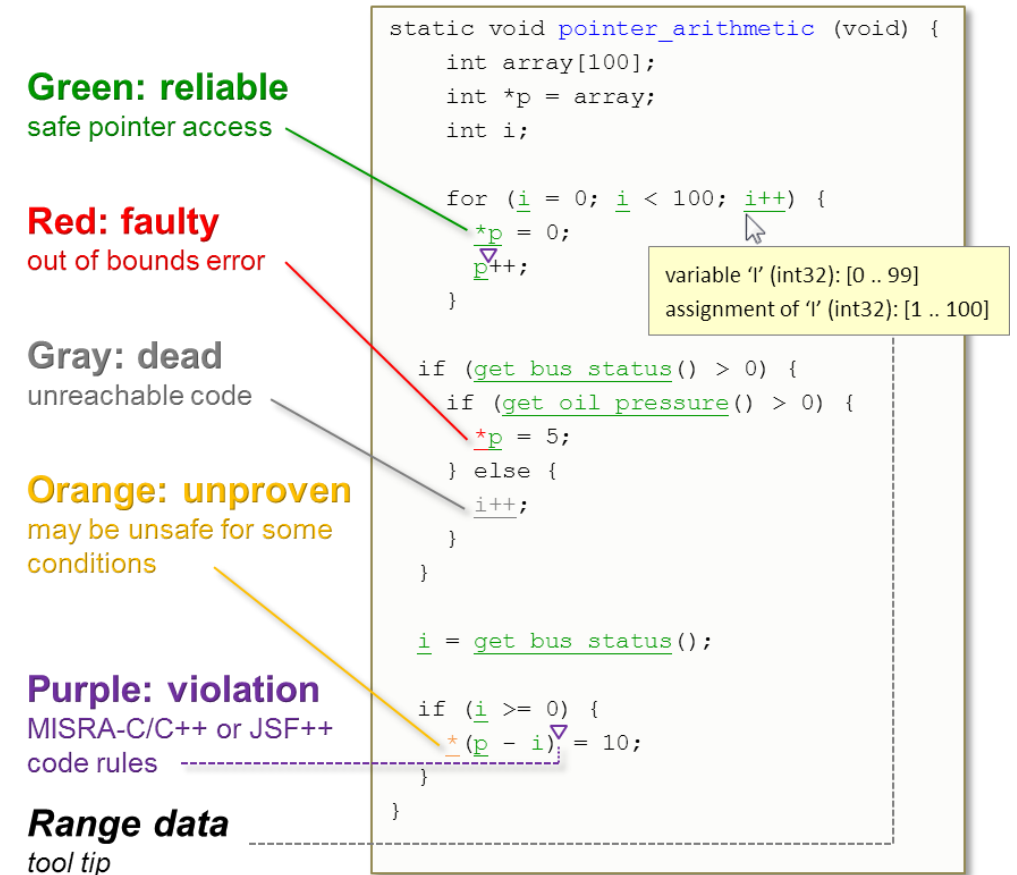


The Generated Code is integrated with Other Code (Handwritten)

Static Code Analysis with Polyspace

- Code metrics and standards
 - Comment density, cyclomatic complexity,...
 - MISRA and Cybersecurity standards
 - Support for DO-178, ISO 26262,

- Bug finding and code proving
 - Check data and control flow of software
 - Detect bugs and security vulnerabilities
 - Prove absence of runtime errors



Green: reliable
safe pointer access

Red: faulty
out of bounds error

Gray: dead
unreachable code

Orange: unproven
may be unsafe for some conditions

Purple: violation
MISRA-C/C++ or JSF++
code rules

Range data
tool tip

```

static void pointer_arithmetic (void) {
    int array[100];
    int *p = array;
    int i;

    for (i = 0; i < 100; i++) {
        *p = 0;
        p++;
    }

    if (get_bus_status() > 0) {
        if (get_oil_pressure() > 0) {
            *p = 5;
        } else {
            i++;
        }
    }

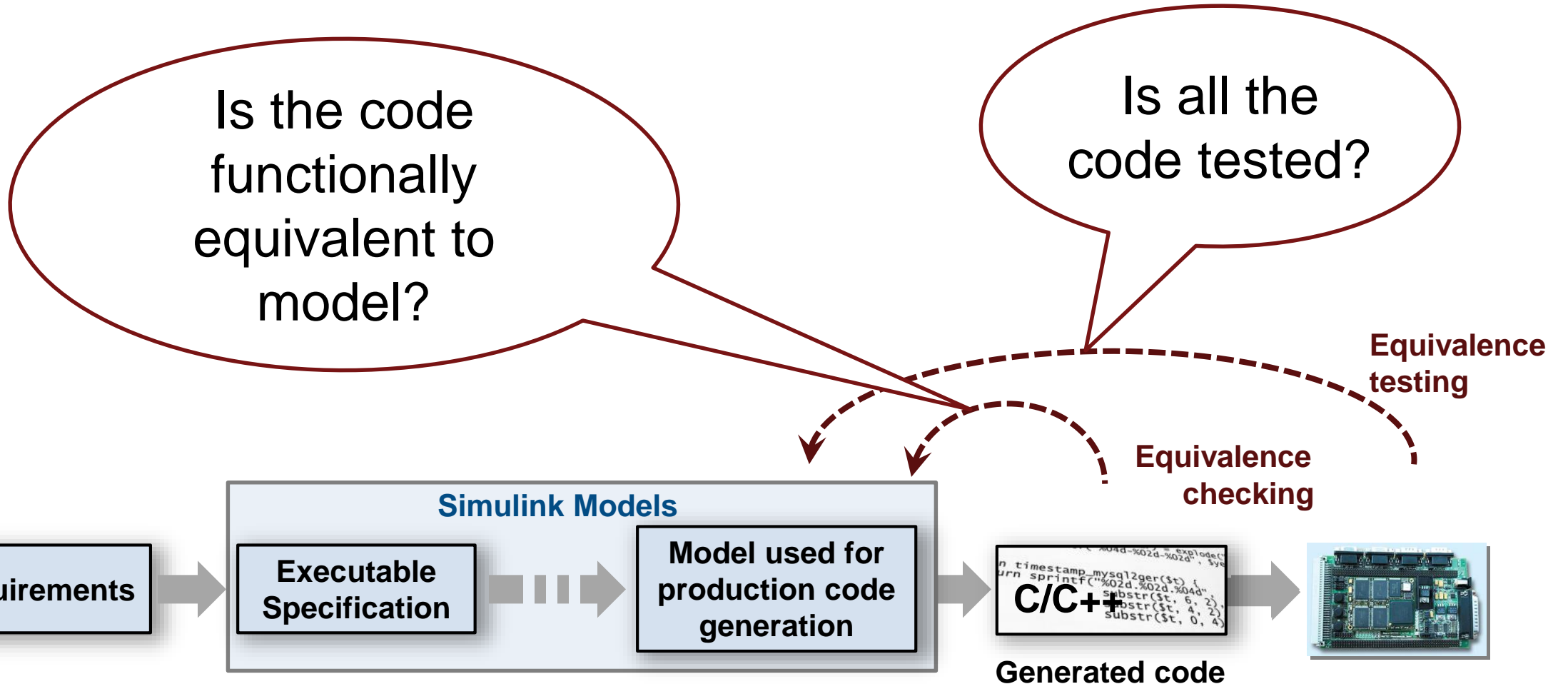
    i = get_bus_status();

    if (i >= 0) {
        *(p - i) = 10;
    }
}
    
```

variable 'i' (int32): [0 .. 99]
assignment of 'i' (int32): [1 .. 100]

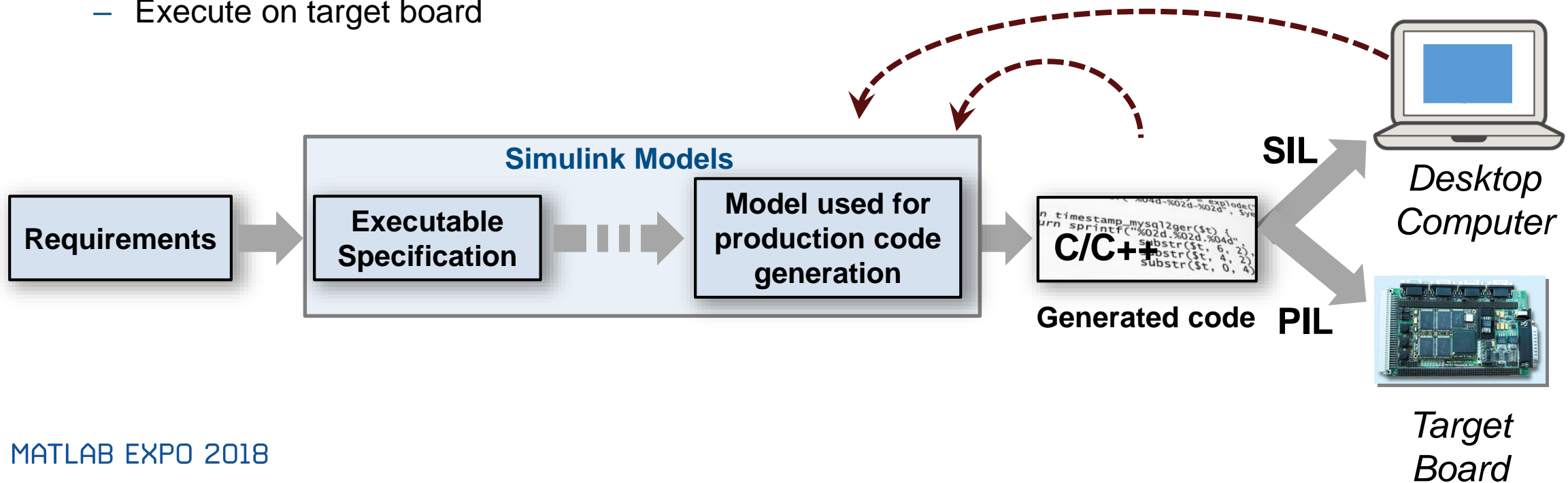
Results from Polyspace Code Prover

Equivalence Testing



Equivalence Testing

- Software in the Loop (SIL)
 - Show functional equivalence, model to code
 - Execute on desktop / laptop computer
- Processor in the Loop (PIL)
 - Numerical equivalence, model to target code
 - Execute on target board
- Re-use tests developed for model to test code
- Collect code coverage



Qualify tools with IEC Certification Kit and DO Qualification Kit

- Qualify code generation and verification products
- Includes documentation, test cases and procedures

KOSTAL Asia R&D Center Receives ISO 26262 ASIL D Certification for Automotive Software Developed with Model-Based Design



Kostal's electronic steering column lock module.

BAE Systems Delivers DO-178B Level A Flight Software on Schedule with Model-Based Design



Primary flight control computers from BAE Systems.

Lear Delivers Quality Body Control Electronics Faster Using Model-Based Design

Challenge

Design, verify, and implement high-quality automotive body control electronics

Solution

Use Model-Based Design to enable early and continuous verification via simulation, SIL, and HIL testing

Results

- Requirements validated early. Over 95% of issues fixed before implementation, versus 30% previously
- Development time cut by 40%. 700,000 lines of code generated and test cases reused throughout the development cycle
- Zero warranty issues reported



Lear automotive body electronic control unit.

"We adopted Model-Based Design not only to deliver better-quality systems faster, but because we believe it is a smart choice. Recently we won a project that several of our competitors declined to bid on because of its tight time constraints. Using Model-Based Design, we met the original delivery date with no problem."

- Jason Bauman, Lear Corporation

Customer References and Applications



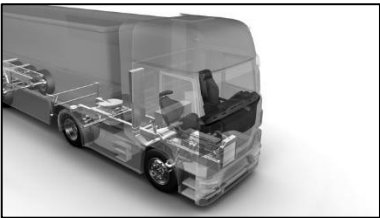
Airbus Helicopters Accelerates Development of DO-178B Certified Software with Model-Based Design

Software testing time cut by two-thirds



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

Specification errors detected early



Continental Develops Electronically Controlled Air Suspension for Heavy-Duty Trucks

Verification time cut by up to 50 percent

More User Stories: www.mathworks.com/company/user_stories.html

Summary

1. Author and manage requirements within Simulink
2. Find defects earlier
3. Automate manual verification tasks
4. Reference workflow that conforms to safety standards

