

Simulink Verification and Validation Product Update

September 2017

R2017b

New! Simulink Requirements

Work with requirements and design together

- Author, edit, and organize requirements

The screenshot displays the Simulink Requirements Perspective interface. The top window shows a requirements diagram with nodes like 'reqMode Cancel', 'reqMode Cruise', and 'reqMode Set' connected to a 'doNot Repeat' block. Two callout boxes provide details for requirements #7 and #8. Requirement #7 is 'Cancel Switch Detection' and requirement #8 is 'Avoid repeating commands'. The bottom window shows a table of requirements with columns for ID, Summary, Implemented, and Verified.

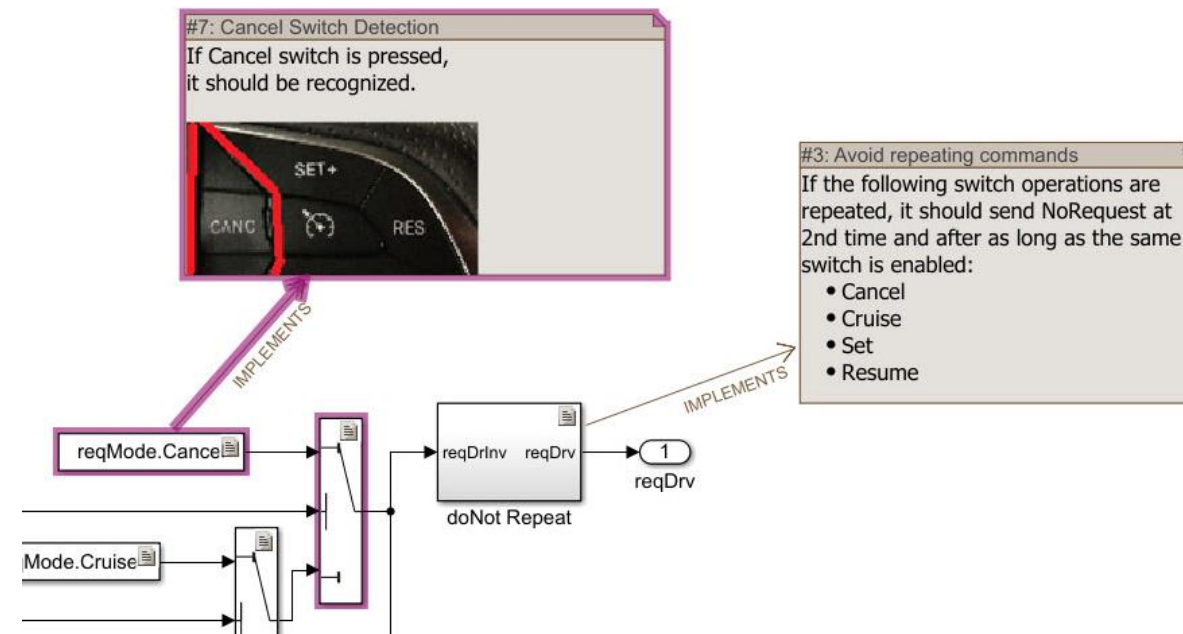
ID	Summary	Implemented	Verified
#1	Driver Switch Request Handling	Implemented	Verified
#2	Switch precedence	Implemented	Verified
#3	Avoid repeating commands	Implemented	Verified
#4	Long Switch recognition	Implemented	Verified
#5	Waiting state for Long Increment switch detection	Implemented	Verified
#6	Waiting state for Long Decrement switch detection	Implemented	Verified
#7	Cancel Switch Detection	Implemented	Verified
#8	Set Switch Detection	Implemented	Verified
#9	Enable Switch Detection	Implemented	Verified
#10	Resume Switch Detection	Implemented	Verified
#11	Transmitter Switch Detection	Implemented	Verified

Requirements Perspective

New! Simulink Requirements

Work with requirements and design together

- Author, edit, and organize requirements
- View and link requirements within the Simulink graphical editor



Requirement Badges and Annotations on Diagram

New! Simulink Requirements

Work with requirements and design together

- Author, edit, and organize requirements
- View and link requirements within the Simulink graphical editor
- Track status and manage requirement changes

Index	Summary	Implemented	Verified
crs_req_func_spec	-	<div style="width: 100%; height: 10px; background-color: blue;"></div>	<div style="width: 100%; height: 10px; background-color: green;"></div>
1	Driver Switch Request Handling	<div style="width: 100%; height: 10px; background-color: blue;"></div>	<div style="width: 100%; height: 10px; background-color: green;"></div>
1.1	Switch precedence	Implemented: 16, Justified: 0, Total: 19	
1.2	Avoid repeating commands	<div style="width: 100%; height: 10px; background-color: blue;"></div>	
1.3	Long Switch recognition	<div style="width: 100%; height: 10px; background-color: blue;"></div>	
1.4	Cancel Switch Detection	<div style="width: 100%; height: 10px; background-color: blue;"></div>	<div style="width: 100%; height: 10px; background-color: red;"></div>
1.5	Set Switch Detection	<div style="width: 100%; height: 10px; background-color: blue;"></div>	
1.6	Enable Switch Detection	<div style="width: 100%; height: 10px; background-color: blue;"></div>	<div style="width: 100%; height: 10px; background-color: green;"></div>
1.7	Resume Switch Detection	<div style="width: 100%; height: 10px; background-color: blue;"></div>	
1.8	Increment Switch Detection	<div style="width: 100%; height: 10px; background-color: blue;"></div>	<div style="width: 100%; height: 10px; background-color: green;"></div>
1.8.1	Increment Short Switch Detection	<div style="width: 100%; height: 10px; background-color: blue;"></div>	<div style="width: 100%; height: 10px; background-color: green;"></div>
1.8.2			

Implementation and Verification Status

Source	Destination	Change Information	
Changed source: 3/12	Changed destination: 4/12	Source: Revision: 1 (Time Stamp: 25-Jul-2017 11:34:04)	
Enabling cruise control	#9 Enable Switch Detection	Issue: Destination Changed.	
Disabling cruise control	#7 Cancel Switch Detection	Stored: Revision: 15 (Time Stamp: 20-May-2017)	Actual: Revision: 18 (Time Stamp: 20-May-2017)
Activating cruise control	#8 Set Switch Detection	<input type="button" value="Clear Issue"/>	
Deactivating cruise control	#8 Set Switch Detection		
Target Speed Increment	#11 Increment Switch Detection		
Target speed decrement	#15 Decrement Switch Detection		
Target speed decrement	#16 Decrement Short Switch Detection		
Target Speed Increment	#12 Increment Short Switch Detection		
Successive Target Speed Increment	#13 Increment Long Switch Detection		
Successive Target Speed Increment	#14 Intermediate state		
Successive Target Speed Decrement	#17 Decrement Long Switch Detection		
Successive Target Speed Decrement	#18 Intermediate state		

Notification of Requirement Changes

New! Simulink Requirements

Work with requirements and design together

- Author, edit, and organize requirements
- View and link requirements within the Simulink graphical editor
- Track status and manage requirement changes
- Trace requirements to models, generated code, and test cases

The screenshot illustrates the integration of requirements with design and testing in Simulink R2017b. It is divided into three main sections:

- Model View:** A Simulink block diagram with a 'Switch' block highlighted in a pink box.
- Generated Code:** A window titled 'Highlight code for block: '<S2>/Switch'' showing code snippets. Line 124 is highlighted: `/* Switch: '<S2>/Switch' incorporates:`. Subsequent lines list constants: `* Constant: '<S74>/Constant'`, `* Constant: '<S83>/Constant'`, `* Constant: '<S84>/Constant'`, `* Constant: '<S95>/Constant'`, and `* Constant: '<S96>/Constant'`.
- Links:** A 'Links' panel showing relationships:
 - Implemented by:** 'Switch' (file icon), 'Enumerated Constant' (file icon).
 - Verified by:** 'Cancel button' (file icon with a red 'X').
 - Derived from:** 'Disabling cruise control' (file icon).
- Tests and Test Results:** A 'Simulink Test' window showing a table of test results:

NAME	STATUS
Results: 2017-Aug-01 18:08:34	7 ✔ 1 ✘
Unit test for DriverSwRequest	7 ✔ 1 ✘
Enable button	✔
Cancel button	✘
Set button	✔
Resume button	✔
Increment button short	✔

 A green checkmark icon is next to the 'Unit test for DriverSwRequest' row, and a red 'X' icon is next to the 'Cancel button' row.

New! Simulink Requirements

Work with requirements and design together

- Author, edit, and organize requirements
- View and link requirements within the Simulink graphical editor
- Track status and manage requirement changes
- Trace requirements to models, generated code, and test cases

The screenshot displays the Simulink Requirements Perspective. The top window shows a Simulink model with requirements like '#7: Cancel Switch Detection' and '#8: Avoid repeating commands' linked to model blocks. The bottom window shows the Requirements Inspector, which includes a table of requirements and a 'Links' section.

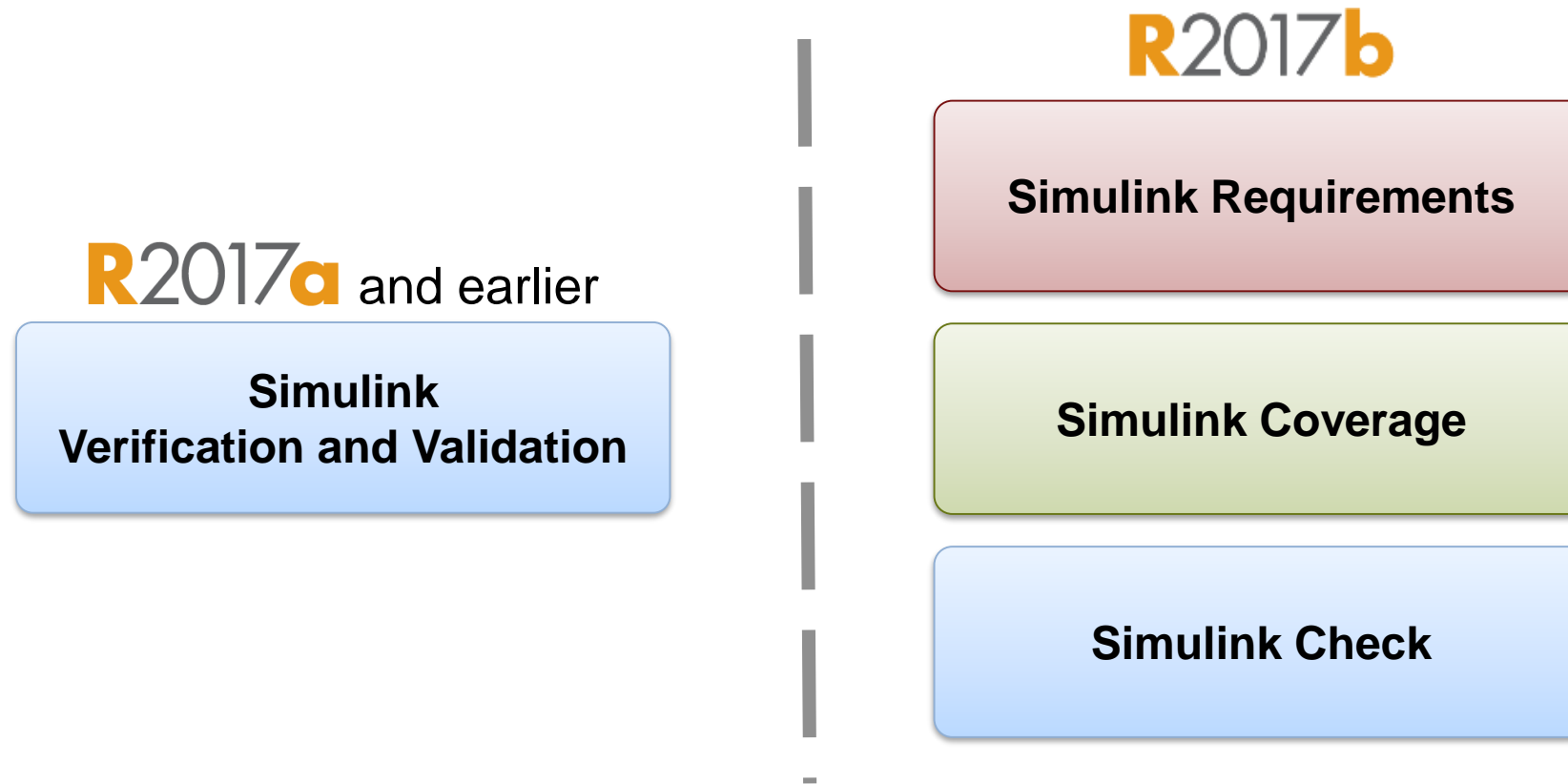
ID	Summary	Implemented	Verified
#1	Driver Switch Request Handling	Implemented	Verified
#2	Switch precedence	Implemented	Verified
#3	Avoid repeating commands	Implemented	Verified
#4	Long Switch recognition	Implemented	Verified
#5	Waiting state for Long Increment switch detection	Implemented	Verified
#6	Waiting state for Long Decrement switch detection	Implemented	Verified
#7	Cancel Switch Detection	Implemented	Verified
#8	Set Switch Detection	Implemented	Verified
#9	Enable Switch Detection	Implemented	Verified
#10	Resume Switch Detection	Implemented	Verified
#11	Increment Switch Detection	Implemented	Verified

The 'Links' section in the Requirements Inspector shows the following links for requirement #7:

- Implemented by: Switch, Enumerated Constant
- Verified by: Cancel button
- Derived from: 3.2_Disablingcruisecont Disabling

Requirements Perspective

New Verification and Validation Products

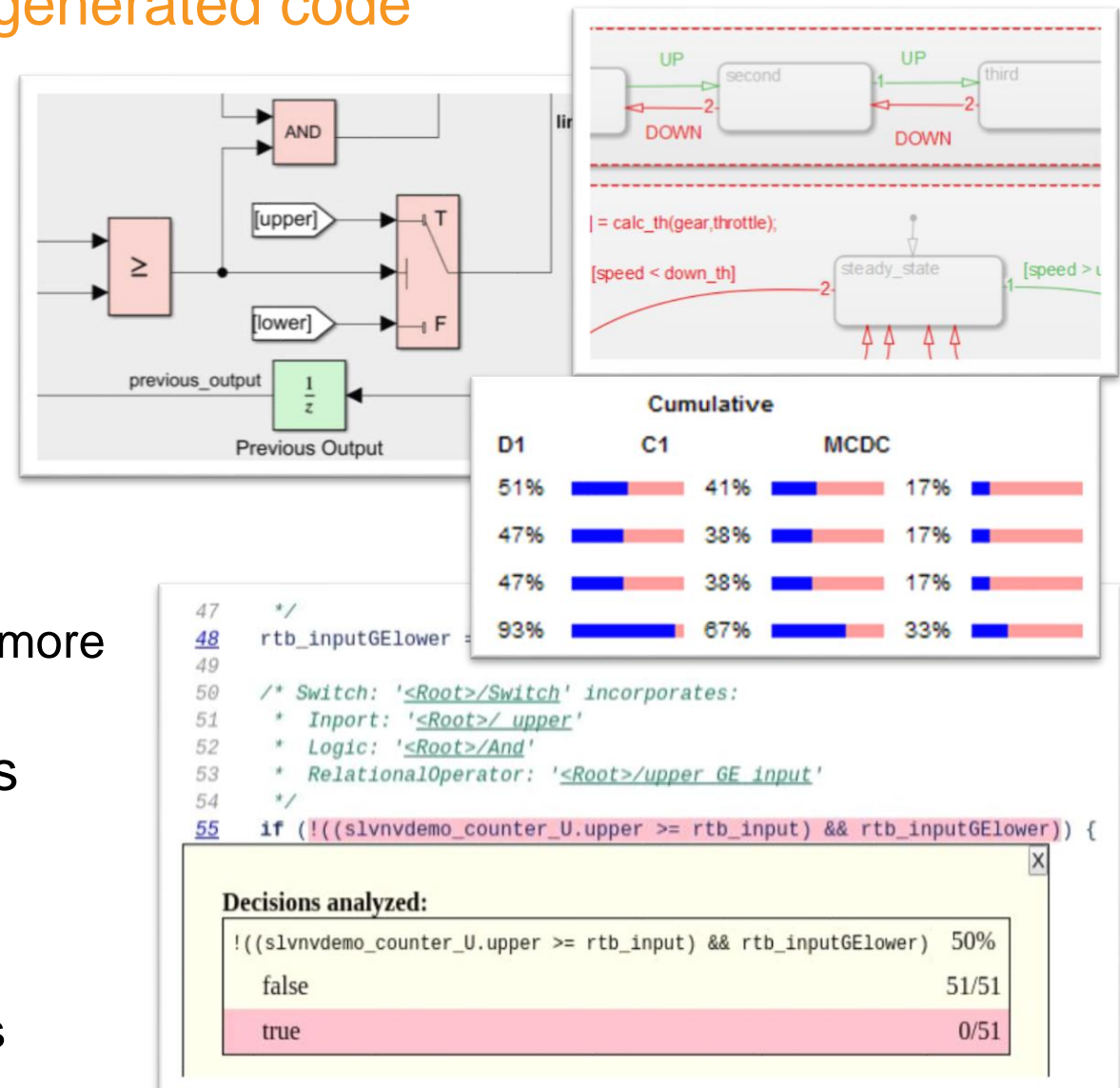


- Simulink Requirements – requirements authoring, editing, trace, management
- Simulink Coverage – model and code coverage analysis
- Simulink Check – static checking, metrics, clone detection

Simulink Coverage

Measure test coverage in models and generated code

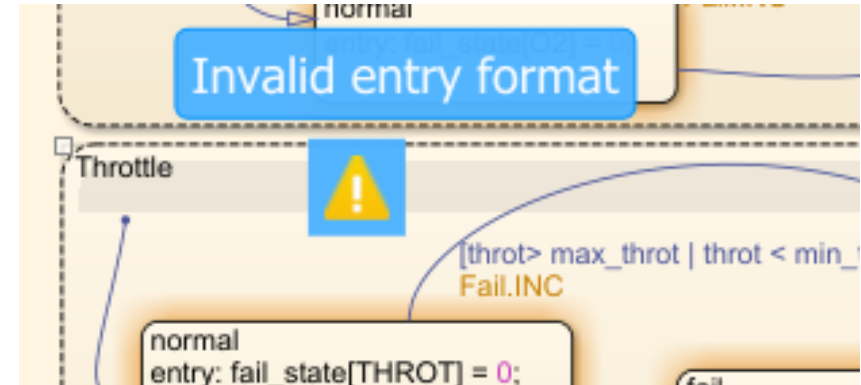
- Model coverage analysis
 - Including C/C++ S-functions
- Generated code coverage analysis
 - C/C++ generated by Embedded Coder
- Comprehensive coverage metrics
 - Decision, condition, MC/DC, relational boundary, signal range, function call, and more
- Model highlighting and coverage reports
 - Locate where test coverage is missing
- Tool qualification support
 - DO Qualification and IEC Certification Kits



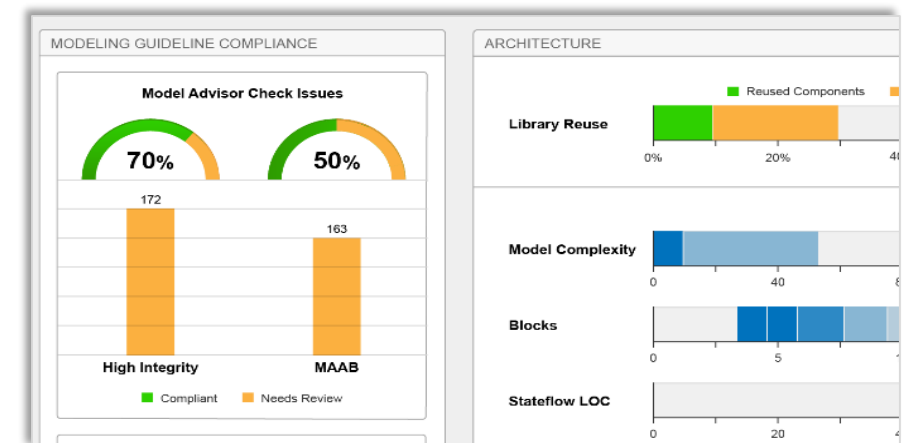
Simulink Check

Verify compliance with guidelines and standards

- Compliance checking for industry and security standards
 - DO-178, ISO 26262, IEC 61508, IEC 62304, MAAB, MISRA
 - Security: CERT C, CWE, ISO/IEC TS 17961
- Edit-time checking to fix violations earlier
- Metrics dashboard to assess design quality
- Refactoring of model clones and model transformations
- Tool qualification support with DO Qualification and IEC Certification Kits

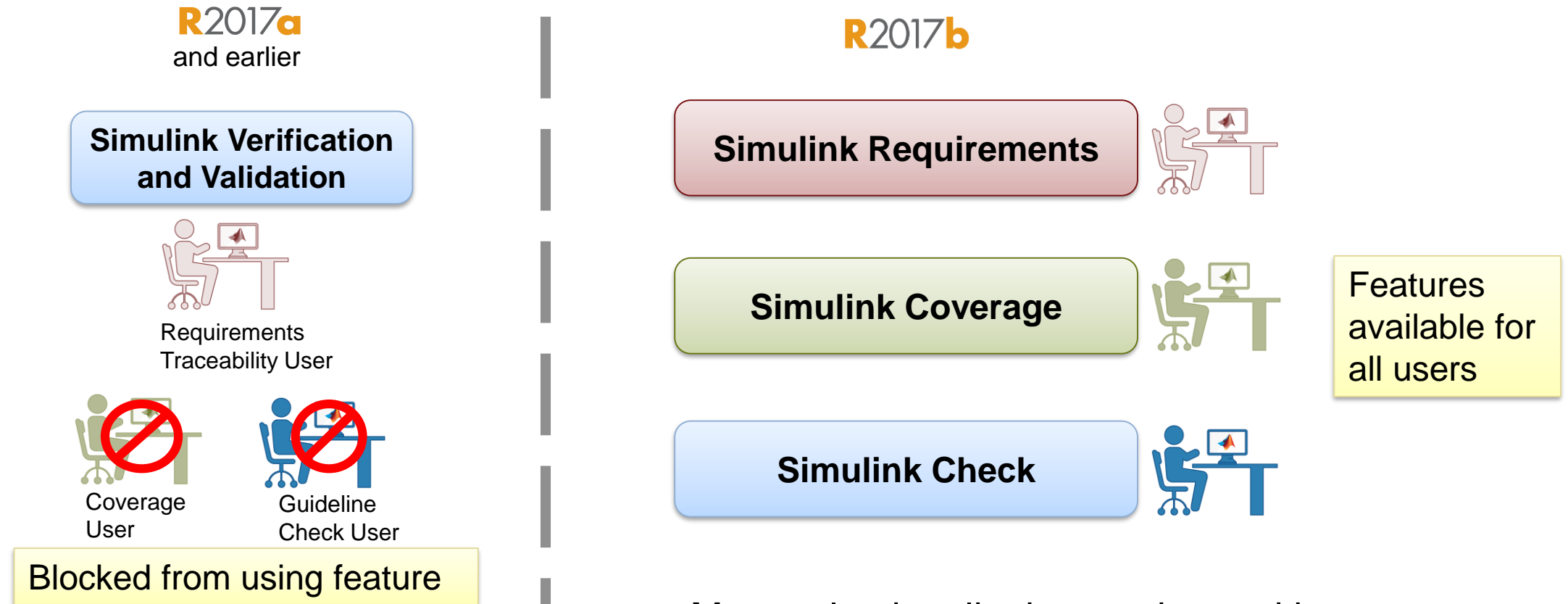


Edit-Time Checks



Metrics Dashboard

Flexibility to Align Products Based on Usage



- More value by aligning products with usage
- Easier to understand focus of products

License Information for Existing Users

Effective as of R2017b

- Licenses of Simulink Verification and Validation automatically updated at no initial cost with:
 - Simulink Requirements
 - Simulink Coverage
 - Simulink Check

- New products each require MATLAB and Simulink
 - No dependencies on each other

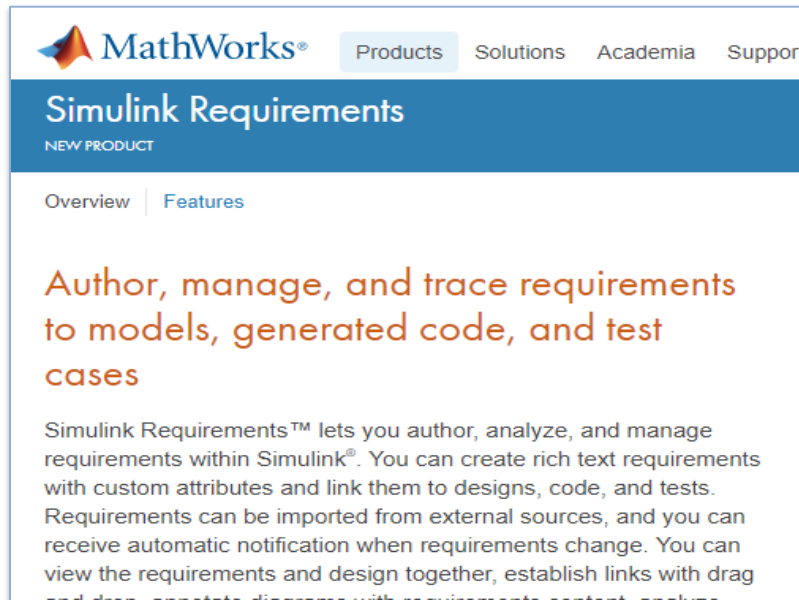
- Simulink Design Verifier requires *Simulink Check* and *Simulink Coverage*

- Simulink Check provides license key for usage of R2017a or earlier releases

Learn More - New Product Web Pages

mathworks.com/products

[Simulink Requirements](#)



MathWorks® Products Solutions Academia Support

Simulink Requirements

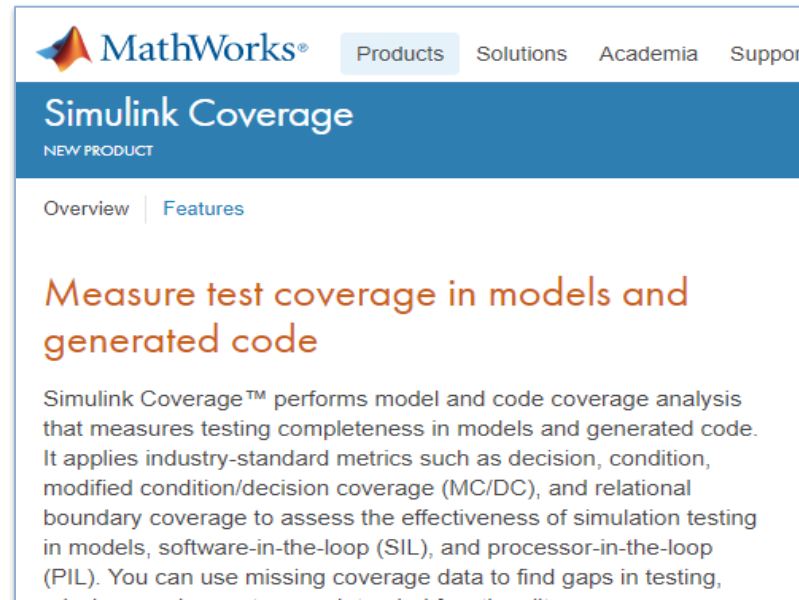
NEW PRODUCT

Overview | Features

Author, manage, and trace requirements to models, generated code, and test cases

Simulink Requirements™ lets you author, analyze, and manage requirements within Simulink®. You can create rich text requirements with custom attributes and link them to designs, code, and tests. Requirements can be imported from external sources, and you can receive automatic notification when requirements change. You can view the requirements and design together, establish links with drag and drop, annotate diagrams with requirements, context, and...

[Simulink Coverage](#)



MathWorks® Products Solutions Academia Support

Simulink Coverage

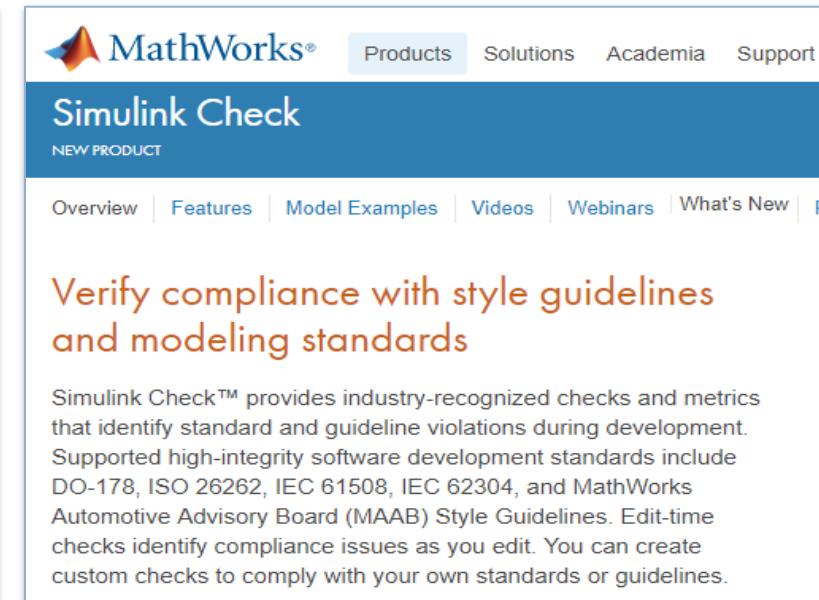
NEW PRODUCT

Overview | Features

Measure test coverage in models and generated code

Simulink Coverage™ performs model and code coverage analysis that measures testing completeness in models and generated code. It applies industry-standard metrics such as decision, condition, modified condition/decision coverage (MC/DC), and relational boundary coverage to assess the effectiveness of simulation testing in models, software-in-the-loop (SIL), and processor-in-the-loop (PIL). You can use missing coverage data to find gaps in testing,...

[Simulink Check](#)



MathWorks® Products Solutions Academia Support

Simulink Check

NEW PRODUCT

Overview | Features | Model Examples | Videos | Webinars | What's New | P

Verify compliance with style guidelines and modeling standards

Simulink Check™ provides industry-recognized checks and metrics that identify standard and guideline violations during development. Supported high-integrity software development standards include DO-178, ISO 26262, IEC 61508, IEC 62304, and MathWorks Automotive Advisory Board (MAAB) Style Guidelines. Edit-time checks identify compliance issues as you edit. You can create custom checks to comply with your own standards or guidelines.

Learn More

See License Related Changes link at mathworks.com/release

If you are:

<p>Subscribed to Software Maintenance Service as of R2017b for Simulink Verification and Validation</p>	<p>License(s) will be updated to include the following products at no initial cost:</p> <ul style="list-style-type: none"> Simulink Check Simulink Coverage Simulink Requirements <p>They will appear on future Software Maintenance Service renewal invoices.</p>
<p>Not subscribed to Software Maintenance Service as of R2017b for Simulink Verification and Validation</p>	<p>License(s) will be updated to include the following products at no initial cost:</p> <ul style="list-style-type: none"> Simulink Check Simulink Coverage Simulink Requirements <p>You will need to renew your Software Maintenance Service subscription to access the updated products.</p>