Verification & Validation of an Autonomous Quadcopter System

Jeremy Ross
Senior Application Engineer
November 7th, 2017

Agenda

- "*Why do verification & validation at all?*"
- Our Quadcopter Story
- Implementing Requirements and Handling Changes
- Verifying Requirements Through Regression Testing
- "*When am I finally done testing*!?"
- Using Static Analysis to Complement Dynamic Testing

Why Do Verification & Validation? → Failure is Not an Option!

- Increasing product complexity
  - Manual testing takes too long and may be incomplete
- Finding defects late
  - Increased costs for rework or significant costs of recalls
- Meeting Industry or Customer's Standards
  - DO-178 (Aero), ISO 26262 (Auto), IEC 62304 (Medical), MAAB, MISRA, etc.
  - Time and cost for safety critical projects estimated 20-30 times more costly*

AUTOS GM See the Fortune 500 list

GM to Recall 4.3 Million Vehicles for Software-Related Airbag Defect

by Reuters SEPTEMBER 9, 2016, 12:37 PM EDT

POPULAR SCIENCE CARS

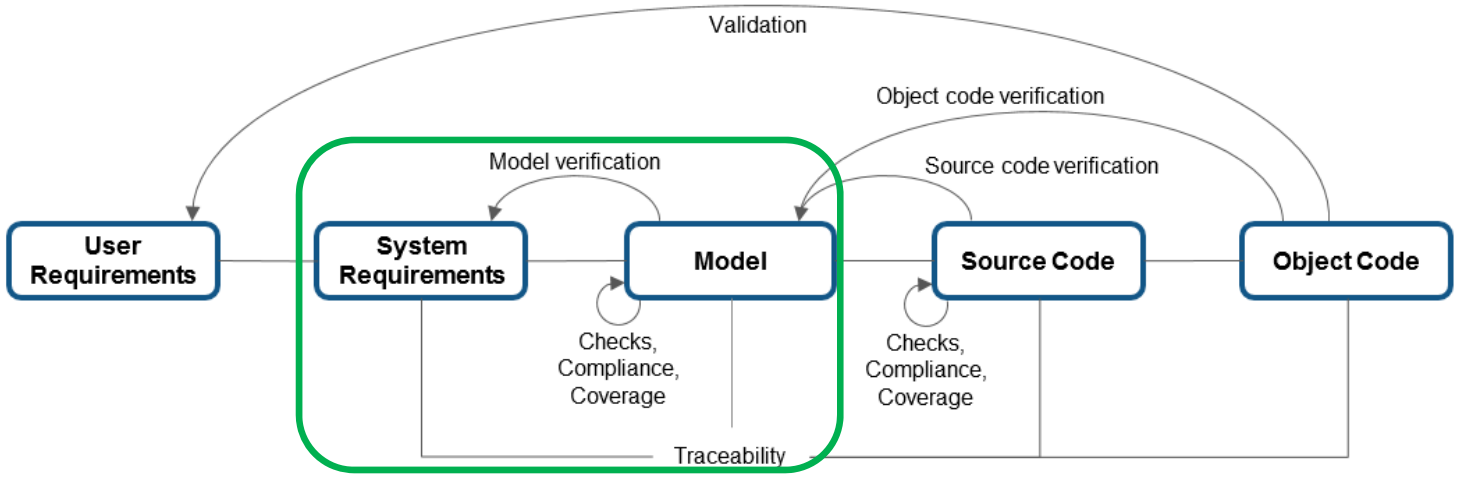
SOFTWARE NOW TO BLAME FOR 15 PERCENT OF CAR RECALLS

YOU CAN'T JUST HOLD THE HOME AND LOCK BUTTONS TO SOLVE THIS ONE

Bengt Halvorson / The Car Connection June 2, 2016

*Source: **Certification Requirements for Safety-Critical Software,** http://www.rtcmagazine.com/articles/view/100010

Where MathWorks V&V Products Fit into a Design Workflow



Simulink Requirements, Coverage and Check

- Requirement Traceability
- Model and Code Coverage
- Standards Checking and Metrics

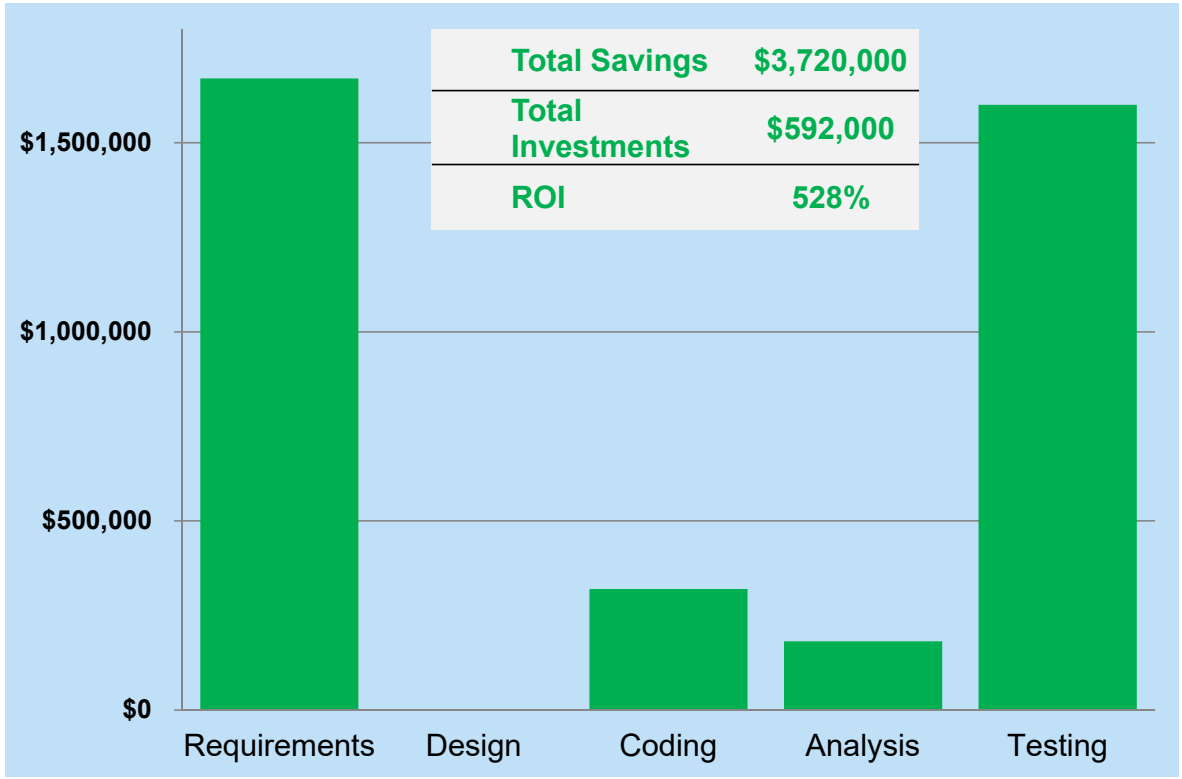
Simulink Test

- Test Harnesses
- Test Sequences
- Test Manager

Simulink Design Verifier

- Test Generation
- Design Error Detection
- Requirements Proving
- Model Slicer

Where Customers Measure the Biggest ROI with V&V Tools



Aerospace Customer Data Shared with MathWorks

Customer User Stories – Not Just Big Aerospace!

NEW!

Bell Helicopter

Traceability enabled the team to perform an impact analysis to identify areas of the Simulink model that would be affected if requirements were updated later in the project

Link to User Story

NEW!

Chery Automobile

Established bidirectional links between requirements and the model elements for Engine Management System software in Simulink that implemented the requirements

Link to User Story

NEW!

ESA and Airbus

Linked elements of the model to system requirements. Automated documentation that incorporated the comments, the linked requirements, and the simulation results for each requirement.

Link to User Story

Baker Hughes

Checked compliance of Oil and Gas Drilling Equipment with MathWorks Automotive Advisory Board (MAAB) modeling standards and measure model coverage of their test cases

Link to User Story

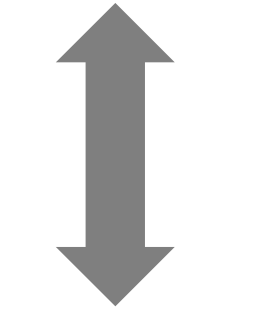
ITK Engineering

Produced model coverage reports for MATLAB unit testing scripts for IEC 62304 Compliant Dental Drill Motor

Link to User Story

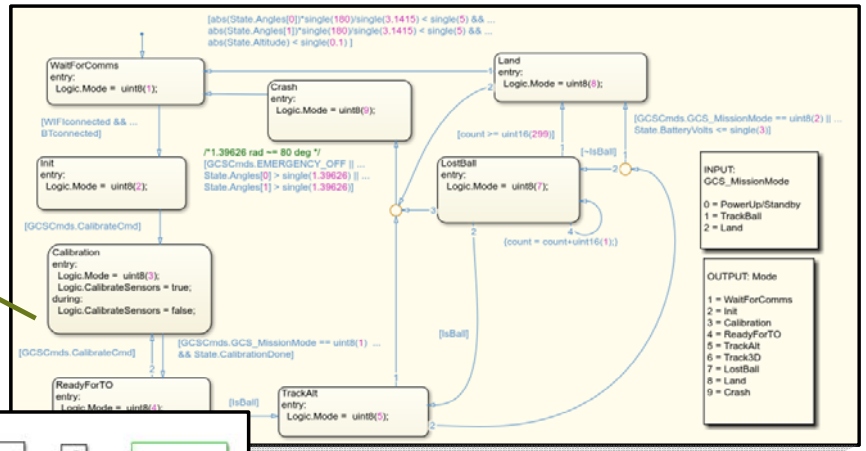
Initial Quadcopter Design Problem

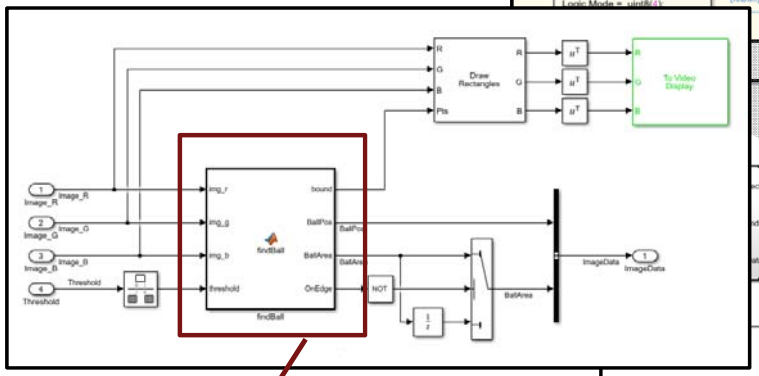
- Control a quadcopter to track a ball up and down



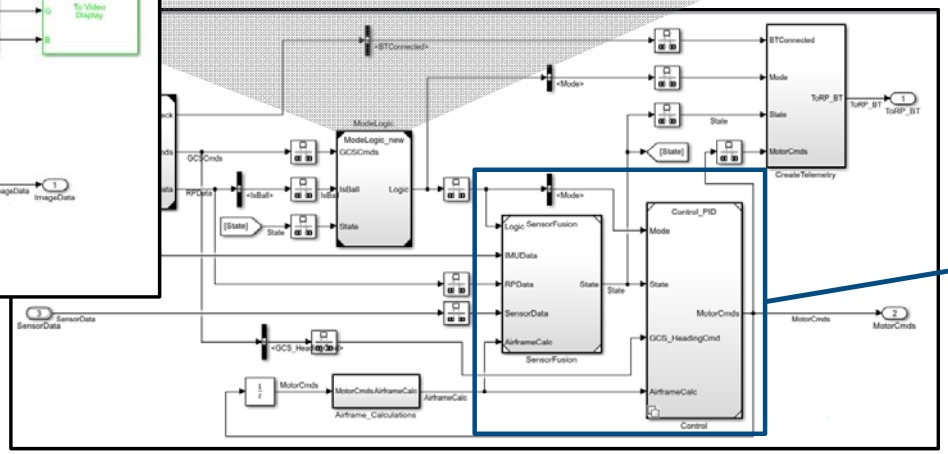
Quadcopter Design Model

Stateflow
Supervisory Logic



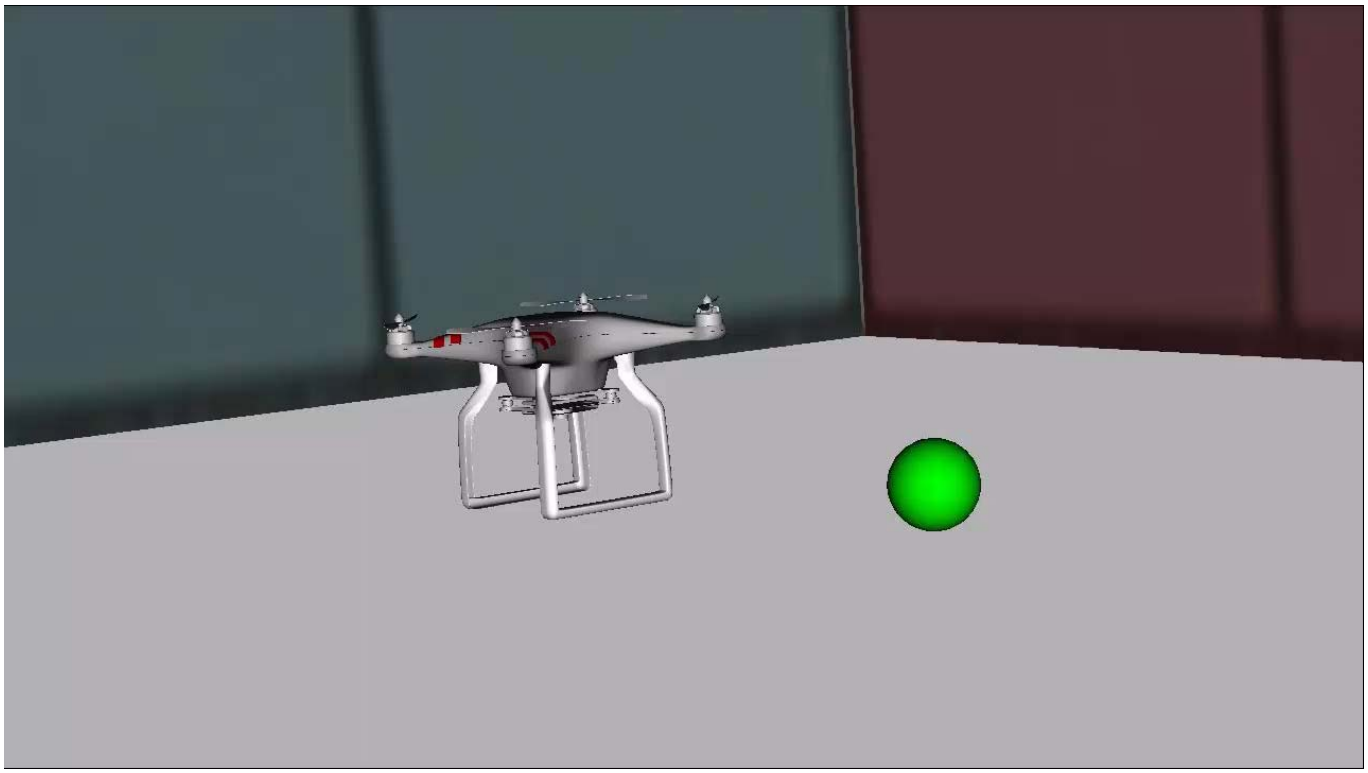


MATLAB
Image Processing and Object Detection



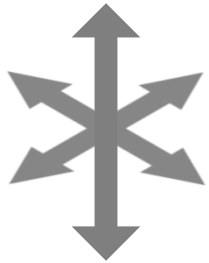
Simulink
Kalman Filtering and PID Airframe Control

Simulation Results



Modified Quadcopter Design Problem

- Requirement Change: Control a quadcopter to track a ball **all around** ~~up and down~~

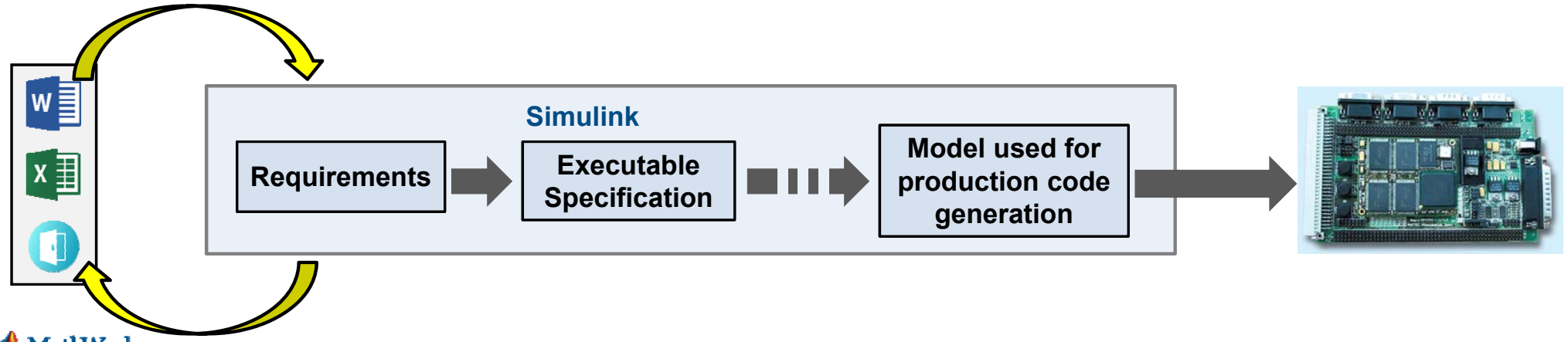


Simulink Requirements

R2017b

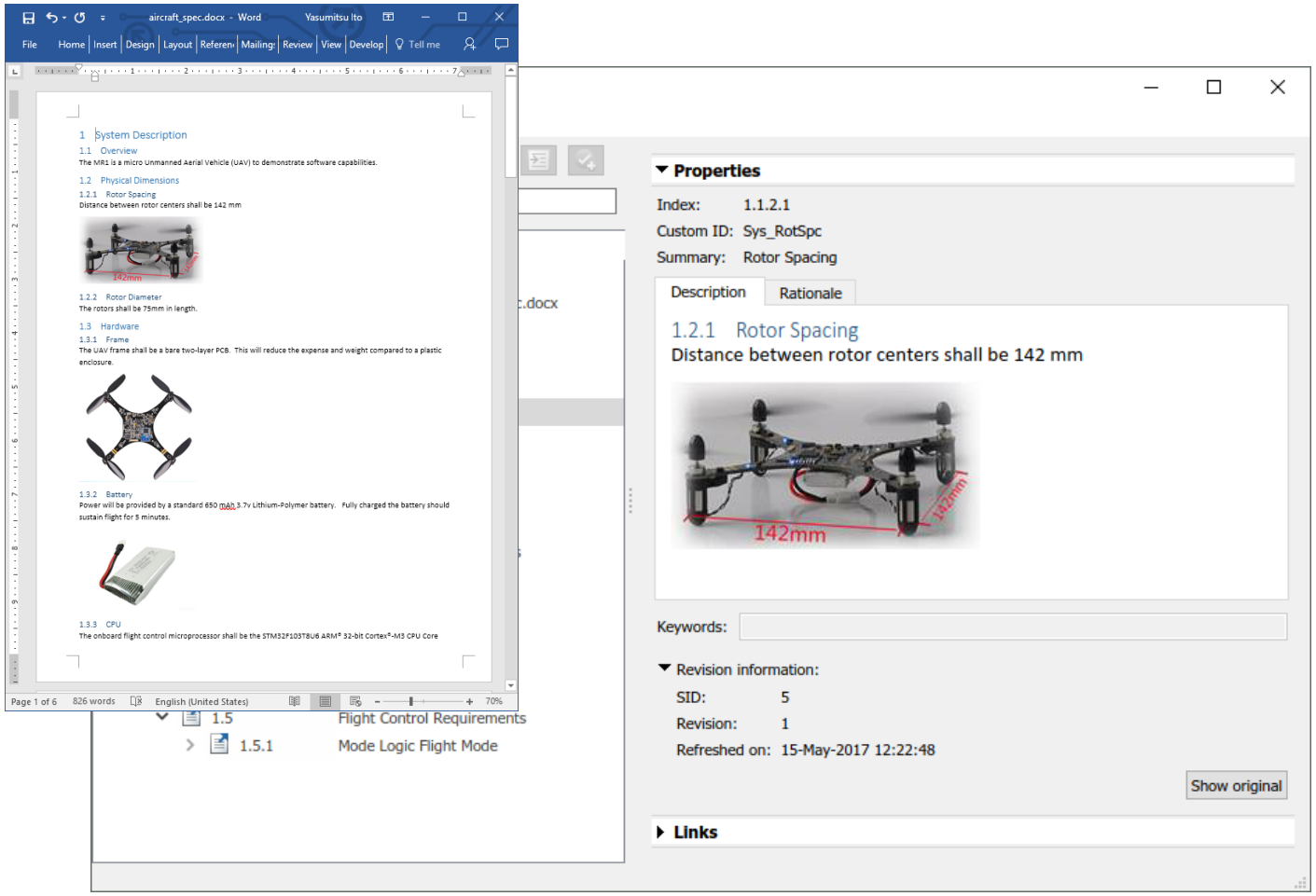
Work with Requirements without Leaving Simulink!

- Author Requirements and Models Together
- View and Link Requirements within the Model
- Track Status and Quickly Manage Requirement Changes
- Trace Requirements to Models and Test Cases → Generated Code



MathWorks

Author or Import Requirements



Author Requirements

- Supports Rich Text
  - Images
  - Tables
  - Bullets
  - ...

OR

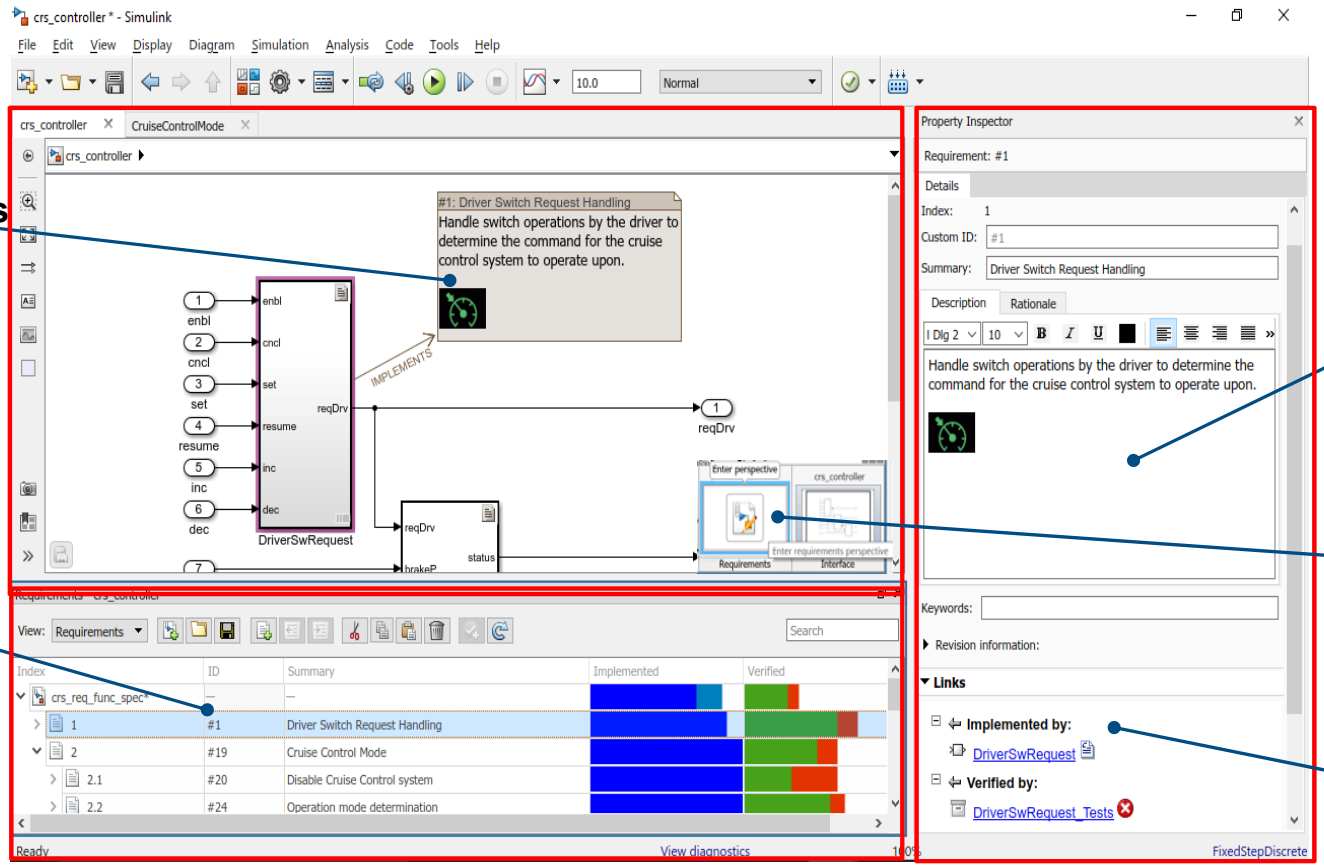
Import External Docs

- Word
- Excel
- DOORS

Requirements Perspective: Combining Design and Requirements

Requirement Text on Simulink Canvas

Requirements Browser with Implementation and Verification Status

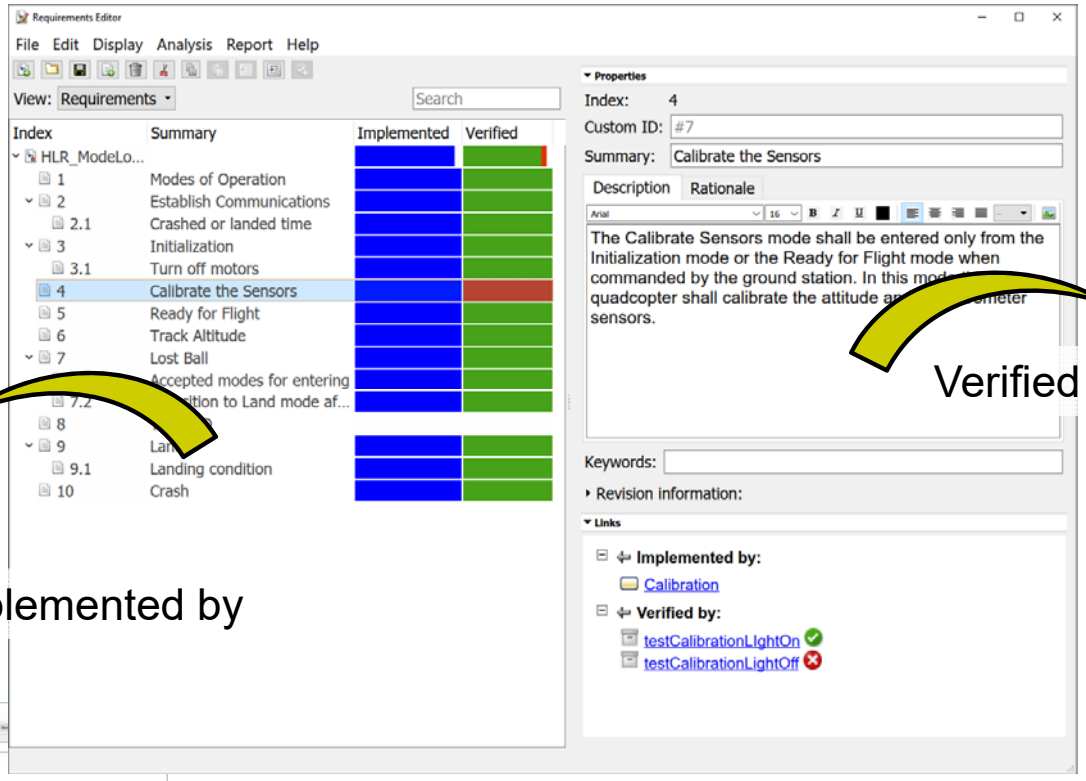


Property Inspector to Modify Requirement and View Links

Enter and Exit Requirements Perspective

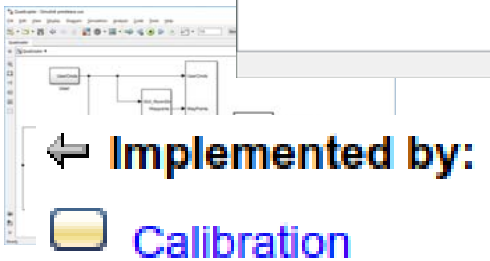
Navigate to associated links (design and test cases)

Track Requirements' Implementation and Verification Status



Implemented by

Verified by



⇐ **Implemented by:**

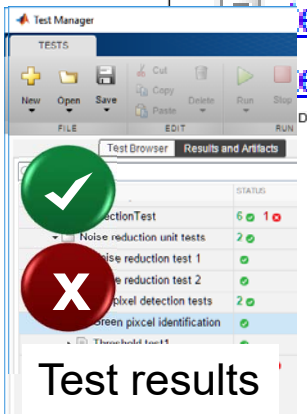
Calibration

⇐ **Verified by:**

testCalibrationLightOn

testCalibrationLightOff

DetectGreenPixels



Test results

Respond to Changes

Identifying Modified Requirements Quickly

Index	Summary	Implemented
aircraft_spec		
1	References to aircraft_spec.docx	
1.1	System Description	
1.2	General Characteristics	
1.3	Performance Requirements	
1.4	Systems Requirements	
1.5	Flight Control Requirements	
1.5.1	Mode Logic Flight Mode	
1.5.1.1	Wait for Communications	
1.5.1.2	Initializatic	
1.5.1.3	Calibration	
1.5.1.4	Ready for T	
1.5.1.5	Track Altitu	
1.5.1.6	Track 3D	
1.5.1.7	Land	
1.5.1.8	Crash	

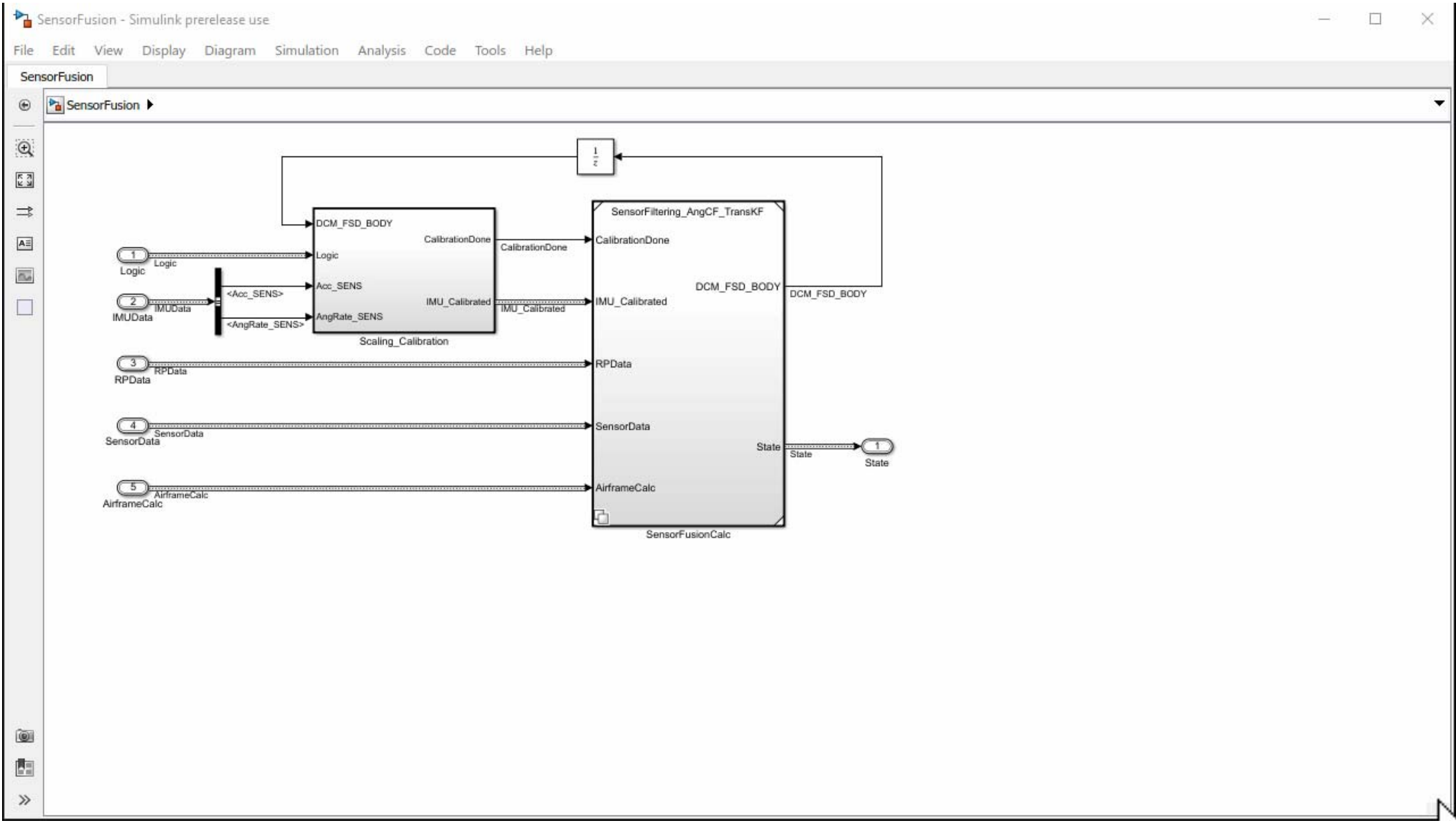
1.5.1.3 Calibration

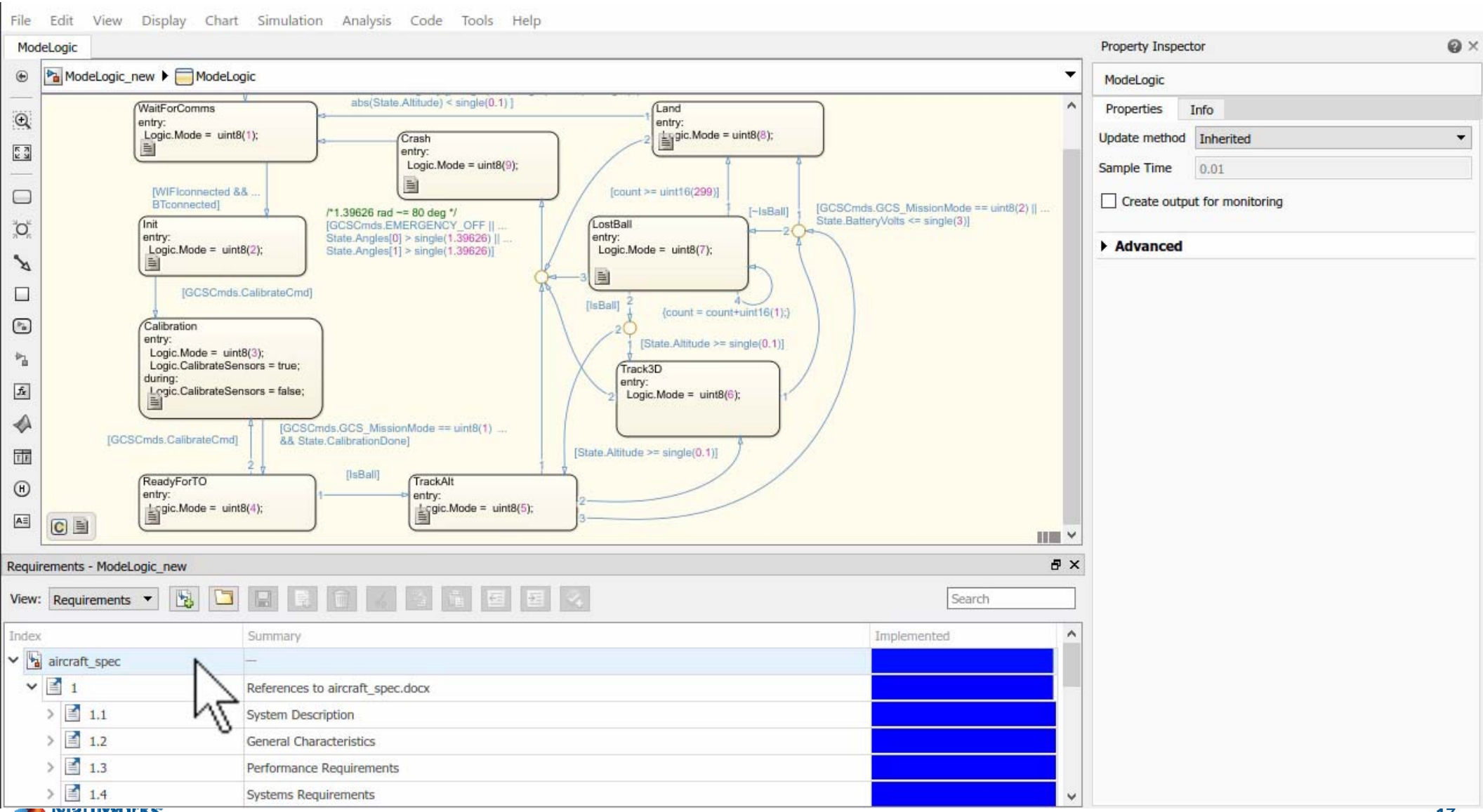
1.5.1.6 Track 3D

- Control a quadcopter to track a ball ~~up and down~~ **all around.**

Updated Requirement

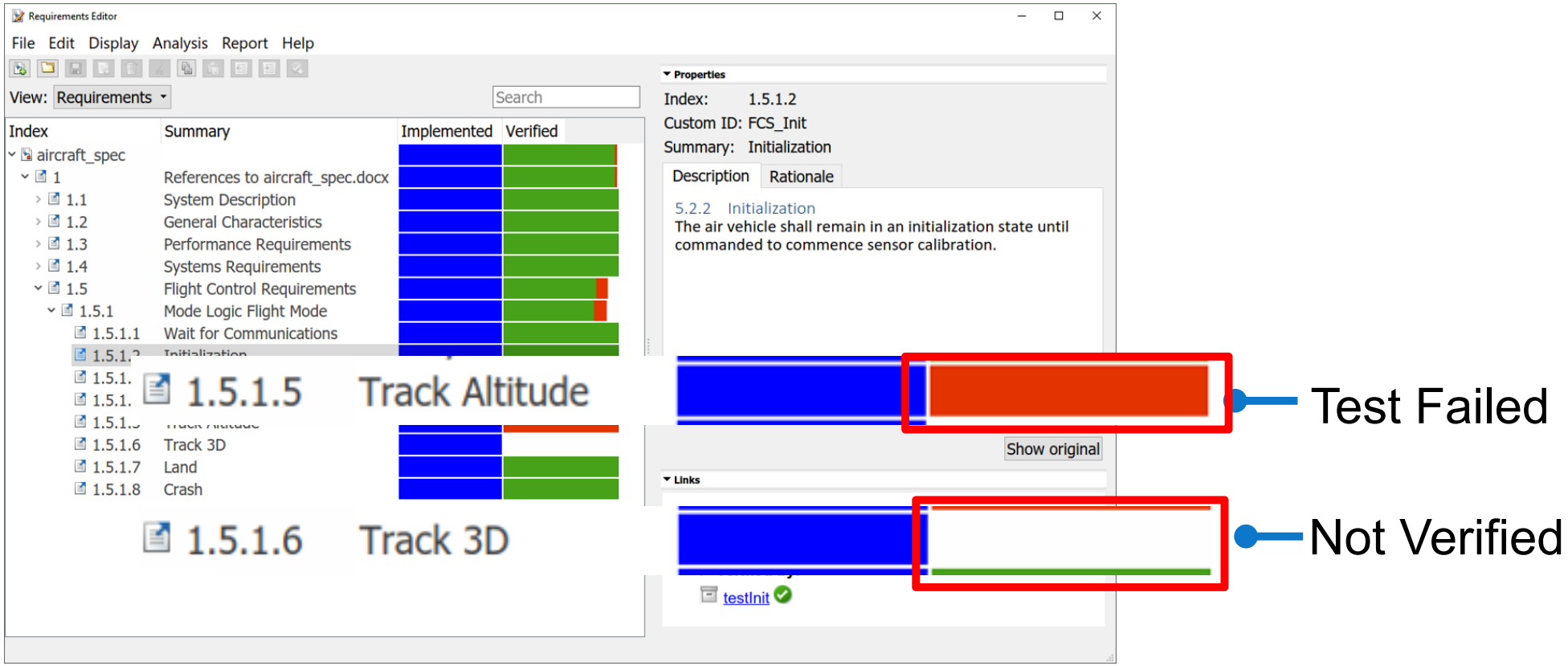
New Requirement


SensorFusion - Simulink prerelease use
File Edit View Display Diagram Simulation Analysis Code Tools Help
SensorFusion
SensorFusion
1/z
DCM_FSD_BODY
CalibrationDone
Logic
Acc_SENS
IMU_Calibrated
AngRate_SENS
Scaling_Calibration
SensorFiltering_AngCF_TransKF
CalibrationDone
DCM_FSD_BODY
IMU_Calibrated
RPData
SensorData
State
AirframeCalc
SensorFusionCalc
1 Logic
2 IMUData
3 RPData
4 SensorData
5 AirframeCalc
<Acc_SENS>
<AngRate_SENS>
1 State

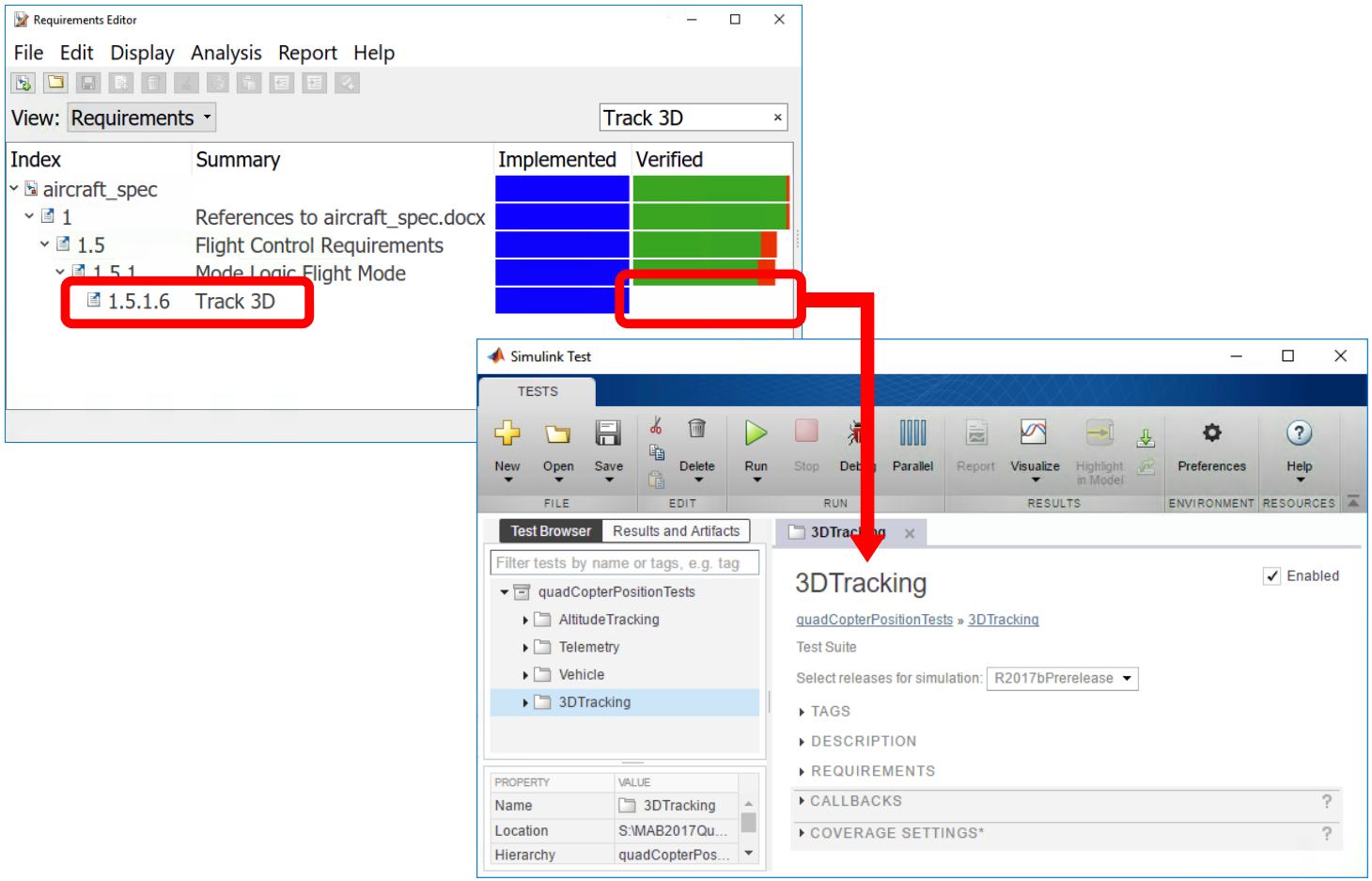

File Edit View Display Chart Simulation Analysis Code Tools Help
ModeLogic
ModeLogic_new ▸ ModeLogic
WaitForComms
entry:
Logic.Mode = uint8(1);
abs(State.Altitude) < single(0.1)]
Land
entry:
Logic.Mode = uint8(8);
Crash
entry:
Logic.Mode = uint8(9);
[WIFIconnected && ...
BTconnected]
/*1.39626 rad ~= 80 deg */
[GCSCmds.EMERGENCY_OFF || ...
State.Angles[0] > single(1.39626) || ...
State.Angles[1] > single(1.39626)]
[count >= uint16(299)]
[~IsBall]
[GCSCmds.GCS_MissionMode == uint8(2) || ...
State.BatteryVolts <= single(3)]
Init
entry:
Logic.Mode = uint8(2);
LostBall
entry:
Logic.Mode = uint8(7);
[GCSCmds.CalibrateCmd]
[IsBall]
{count = count+uint16(1);}
Calibration
entry:
Logic.Mode = uint8(3);
Logic.CalibrateSensors = true;
during:
Logic.CalibrateSensors = false;
[State.Altitude >= single(0.1)]
Track3D
entry:
Logic.Mode = uint8(6);
[GCSCmds.CalibrateCmd]
[GCSCmds.GCS_MissionMode == uint8(1) ...
&& State.CalibrationDone]
[State.Altitude >= single(0.1)]
[IsBall]
ReadyForTO
entry:
Logic.Mode = uint8(4);
TrackAlt
entry:
Logic.Mode = uint8(5);
Property Inspector
ModeLogic
Properties
Info
Update method
Inherited
Sample Time
0.01
Create output for monitoring
Advanced
Requirements - ModeLogic_new
View:
Requirements
Search
Index
Summary
Implemented
aircraft_spec
1
References to aircraft_spec.docx
1.1
System Description
1.2
General Characteristics
1.3
Performance Requirements
1.4
Systems Requirements

Respond to Changes

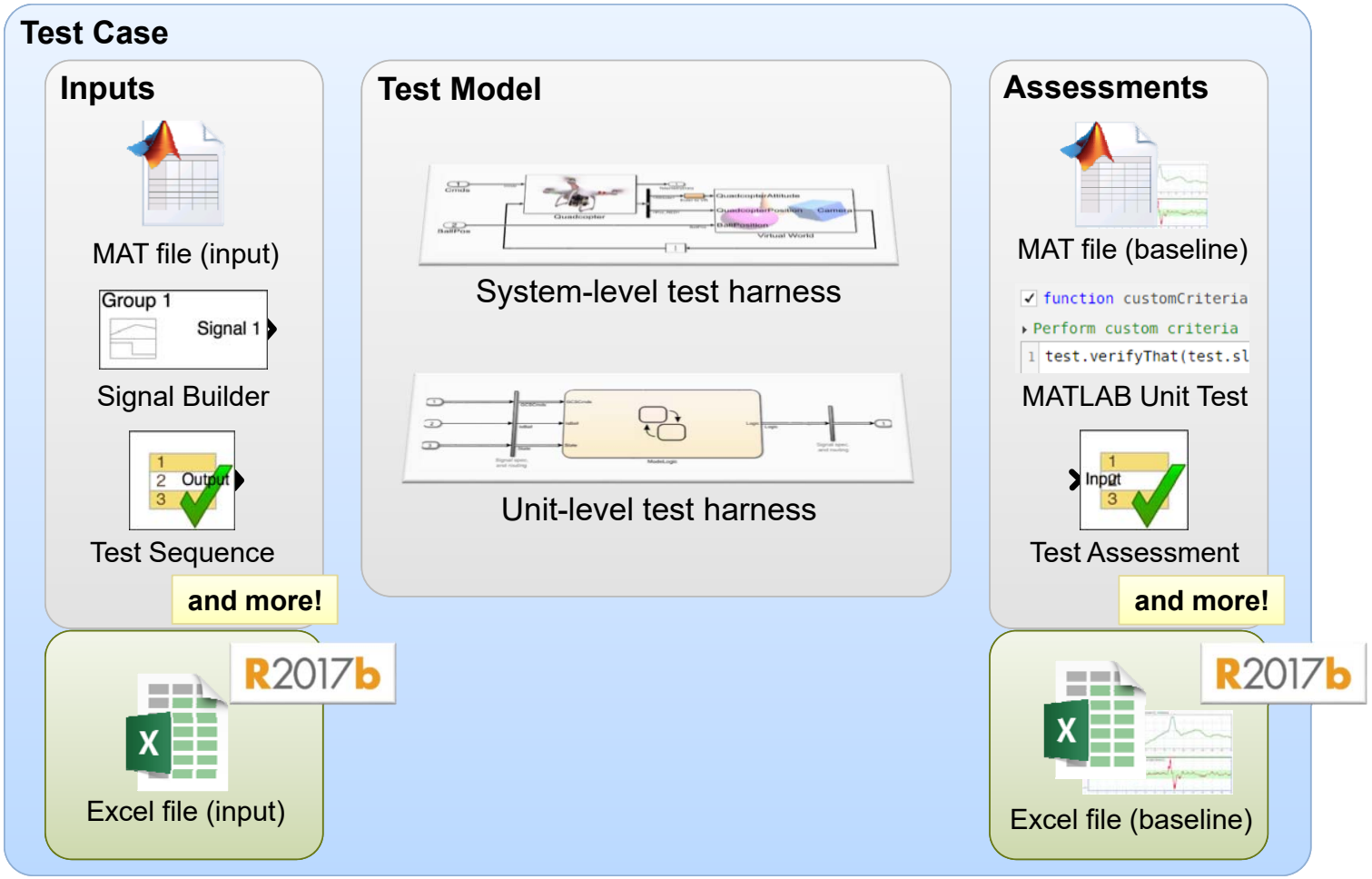
Qualifying with Regression Tests



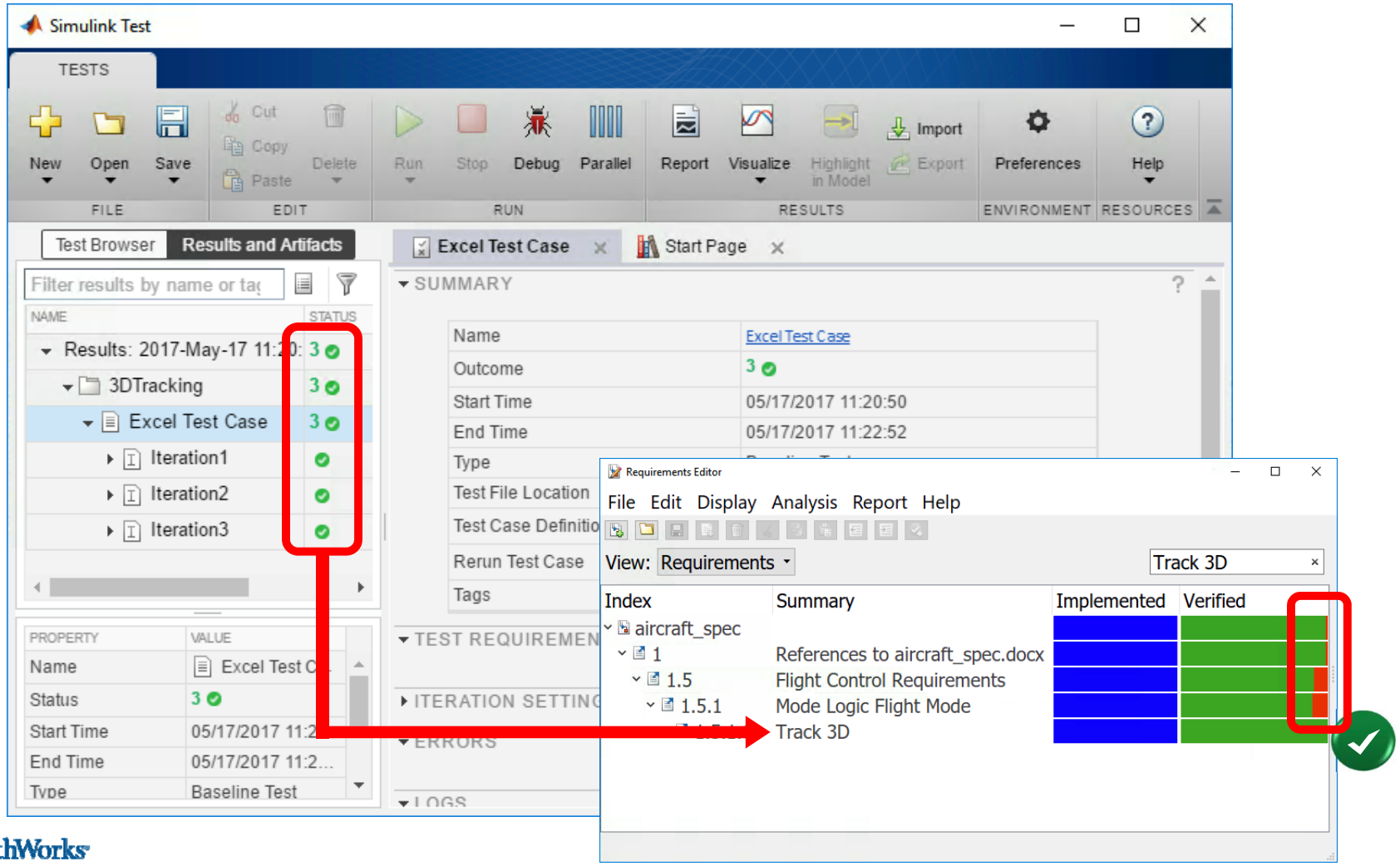
New Requirements → New Test Cases



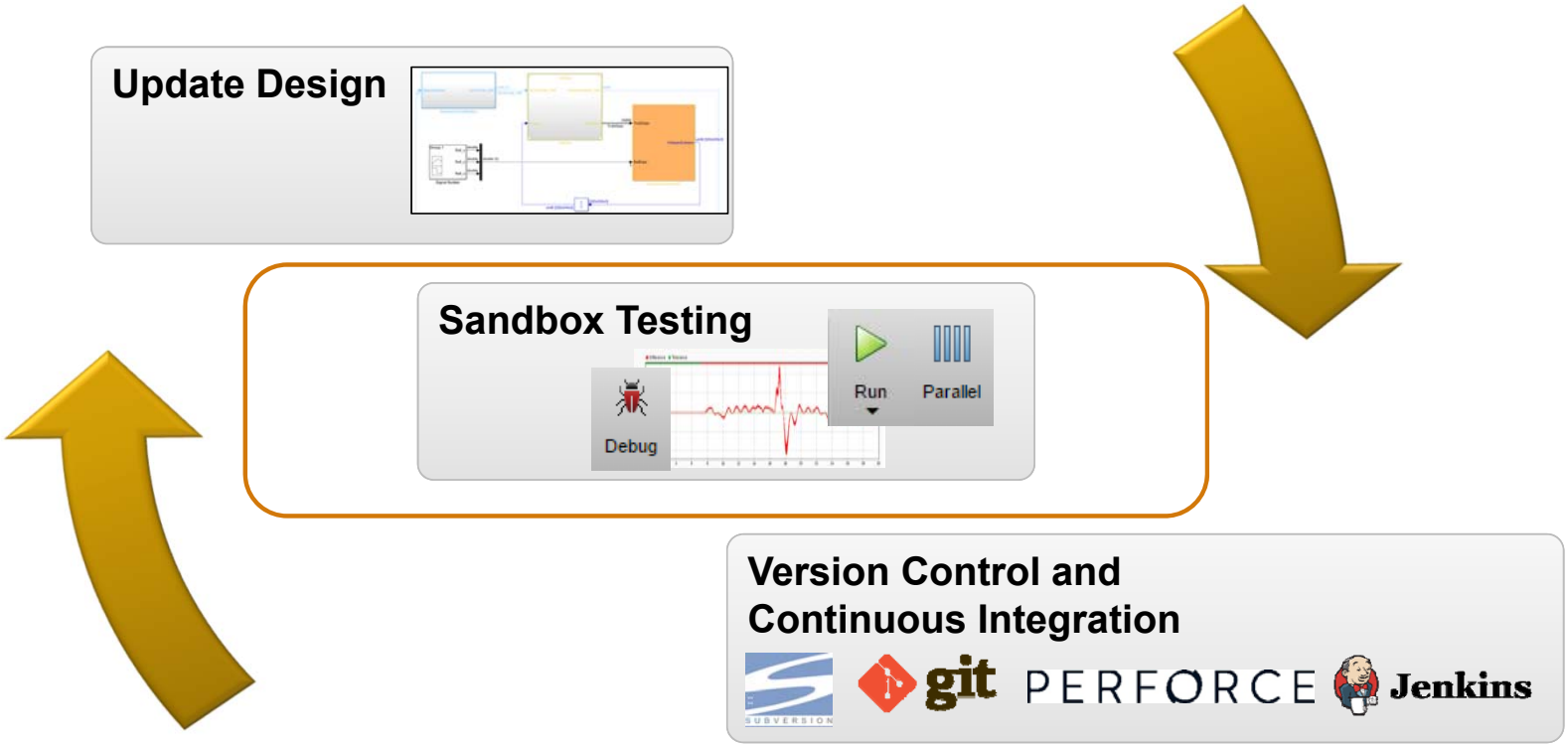
Testing in Simulink Test



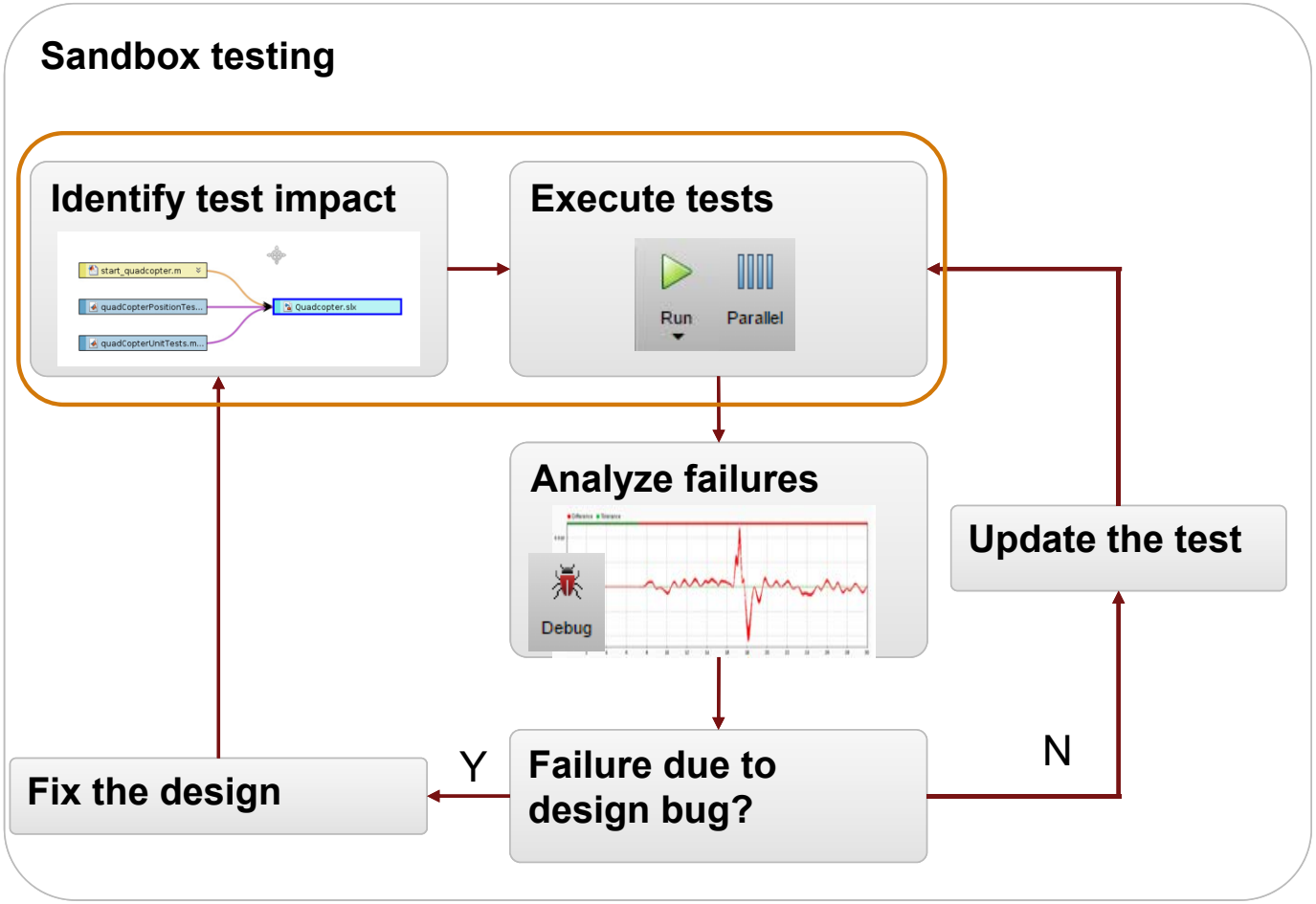
Running Tests with Simulink Test

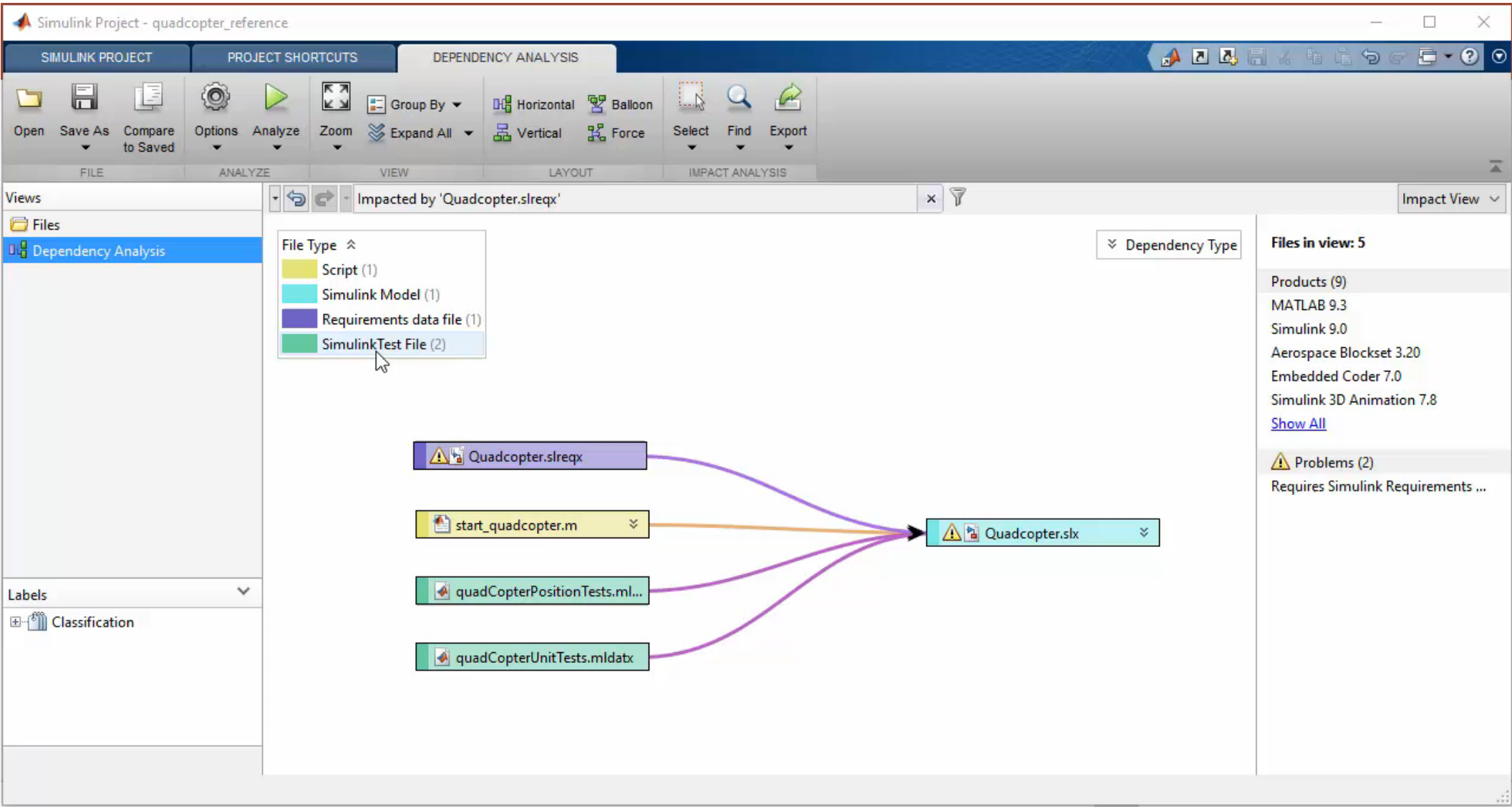


Regression Testing Process

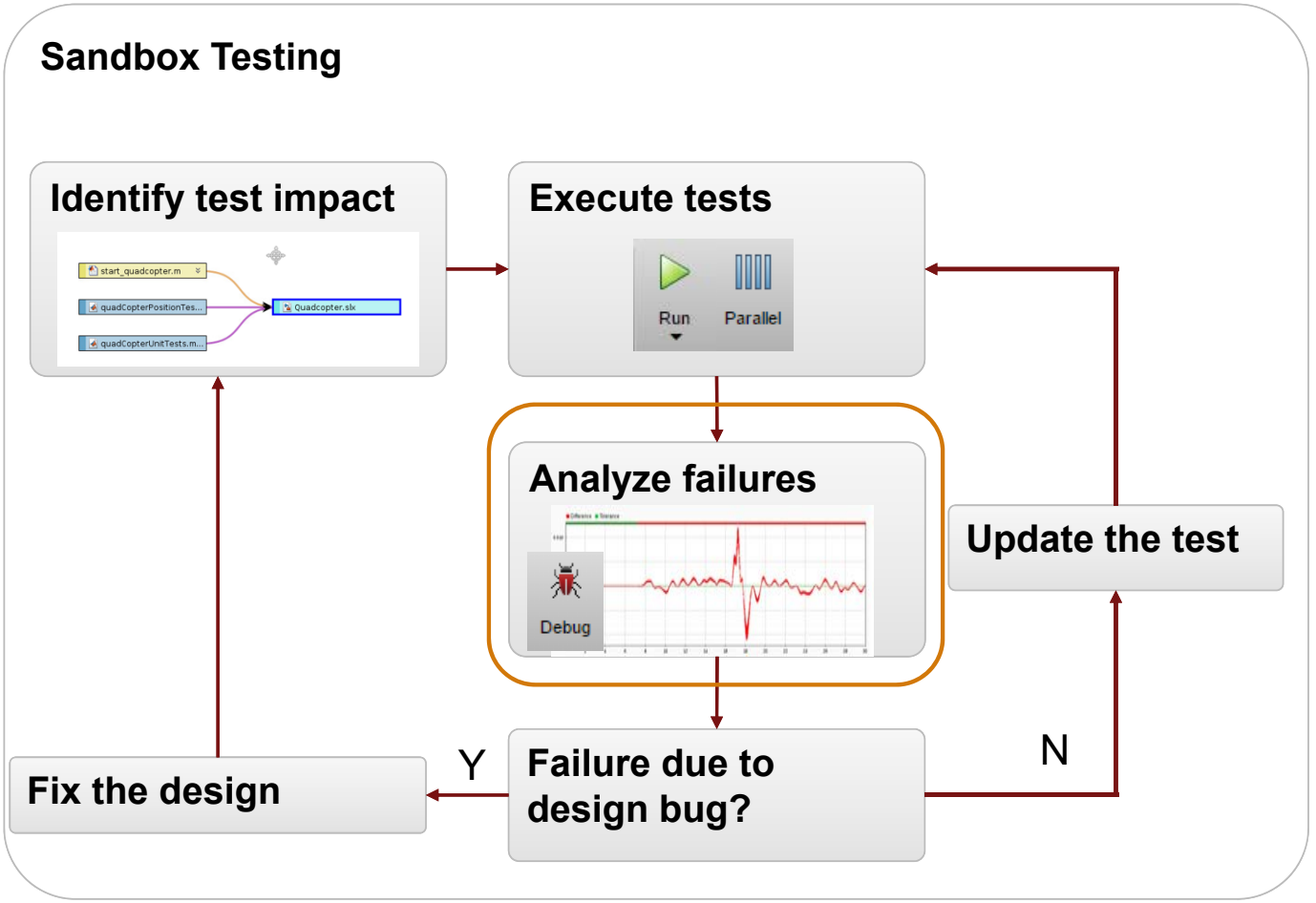


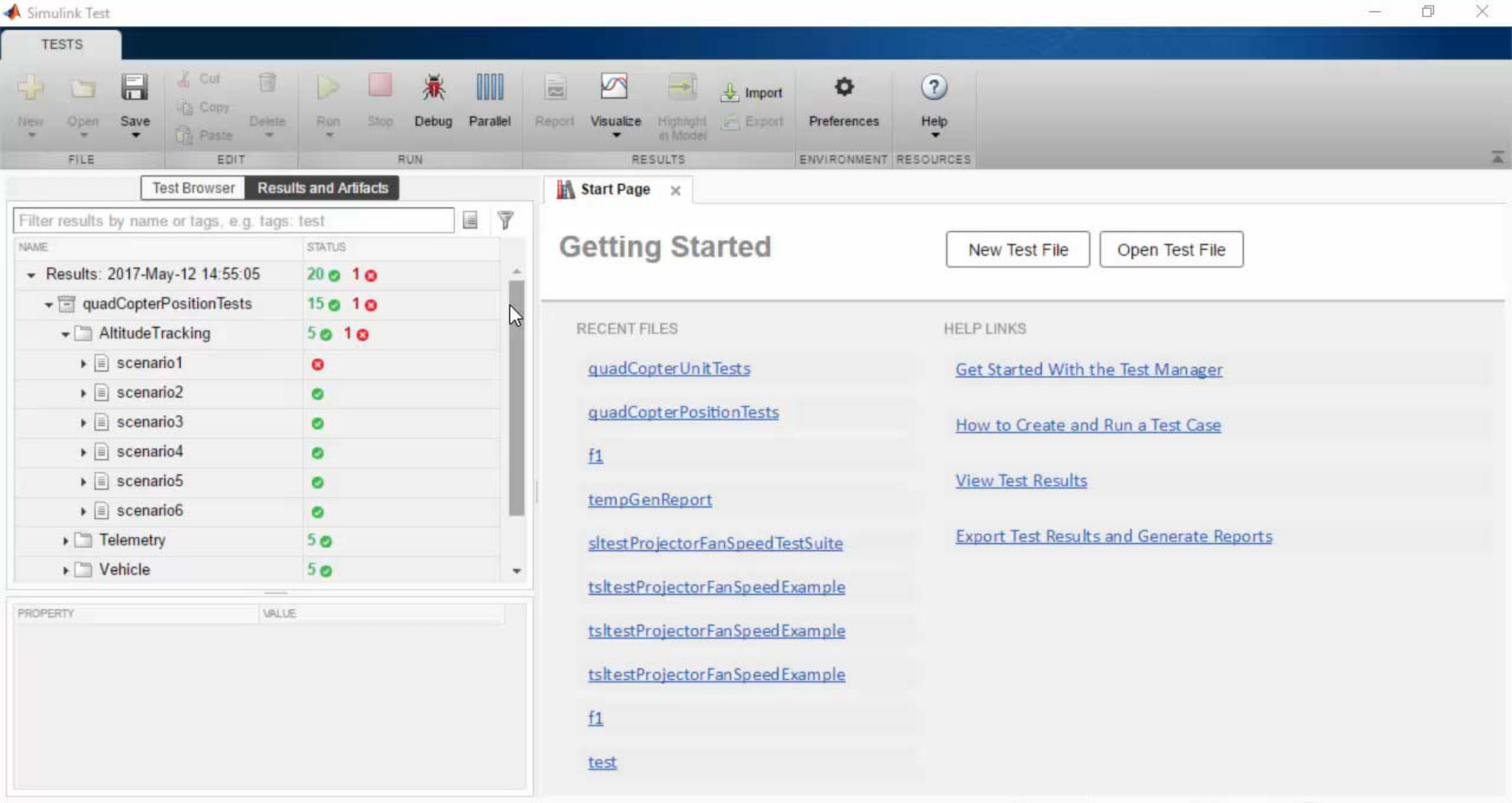
Testing in a Sandbox



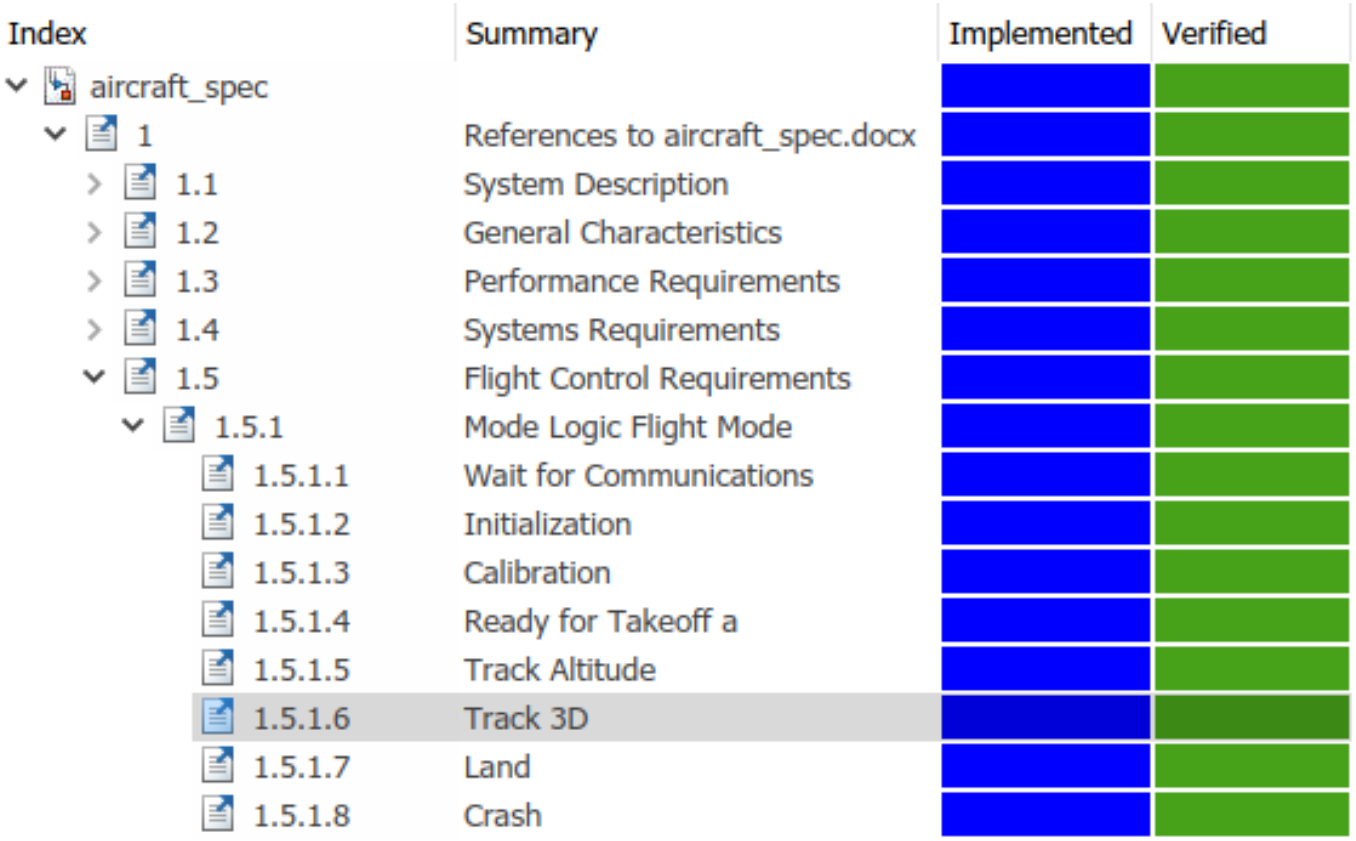

Simulink Project - quadcopter_reference
SIMULINK PROJECT
PROJECT SHORTCUTS
DEPENDENCY ANALYSIS
Open
Save As
Compare to Saved
Options
Analyze
Zoom
Group By
Expand All
Horizontal
Vertical
Balloon
Force
Select
Find
Export
FILE
ANALYZE
VIEW
LAYOUT
IMPACT ANALYSIS
Views
Files
Dependency Analysis
Impacted by 'Quadcopter.slreqx'
Impact View
File Type
Script (1)
Simulink Model (1)
Requirements data file (1)
SimulinkTest File (2)
Dependency Type
Files in view: 5
Products (9)
MATLAB 9.3
Simulink 9.0
Aerospace Blockset 3.20
Embedded Coder 7.0
Simulink 3D Animation 7.8
Show All
Problems (2)
Requires Simulink Requirements ...
Quadcopter.slreqx
start_quadcopter.m
quadCopterPositionTests.ml...
quadCopterUnitTests.mldatx
Quadcopter.slx
Labels
Classification

Testing in a Sandbox




Simulink Test
TESTS
New
Open
Save
Cut
Copy
Paste
Delete
Run
Stop
Debug
Parallel
Report
Visualize
Highlight in Model
Import
Export
Preferences
Help
FILE
EDIT
RUN
RESULTS
ENVIRONMENT
RESOURCES
Test Browser
Results and Artifacts
Filter results by name or tags, e.g. tags: test
NAME
STATUS
Results: 2017-May-12 14:55:05
20 1
quadCopterPositionTests
15 1
AltitudeTracking
5 1
scenario1
scenario2
scenario3
scenario4
scenario5
scenario6
Telemetry
5
Vehicle
5
PROPERTY
VALUE
Start Page
Getting Started
New Test File
Open Test File
RECENT FILES
quadCopterUnitTests
quadCopterPositionTests
f1
tempGenReport
sltestProjectorFanSpeedTestSuite
tsltestProjectorFanSpeedExample
tsltestProjectorFanSpeedExample
tsltestProjectorFanSpeedExample
f1
test
HELP LINKS
Get Started With the Test Manager
How to Create and Run a Test Case
View Test Results
Export Test Results and Generate Reports

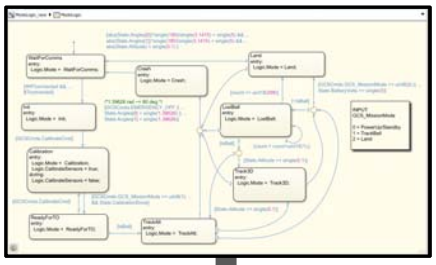
Requirements are Fully Implemented and Verified



"But, how do I know when we've done <u>enough</u> testing?"

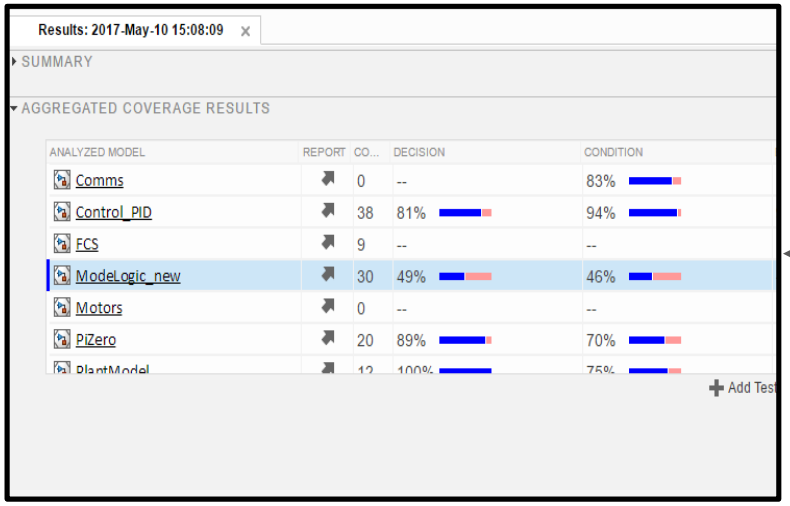
Model Coverage in Dynamic Testing

Stateflow Logic

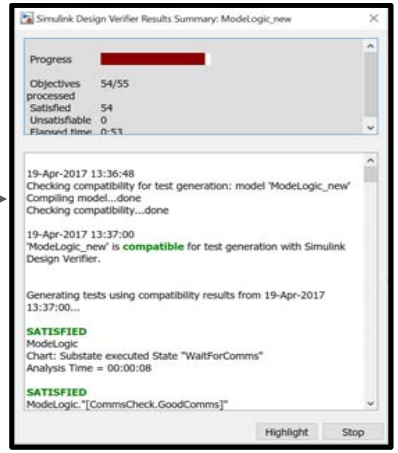


Address Missing Coverage using Static Analysis

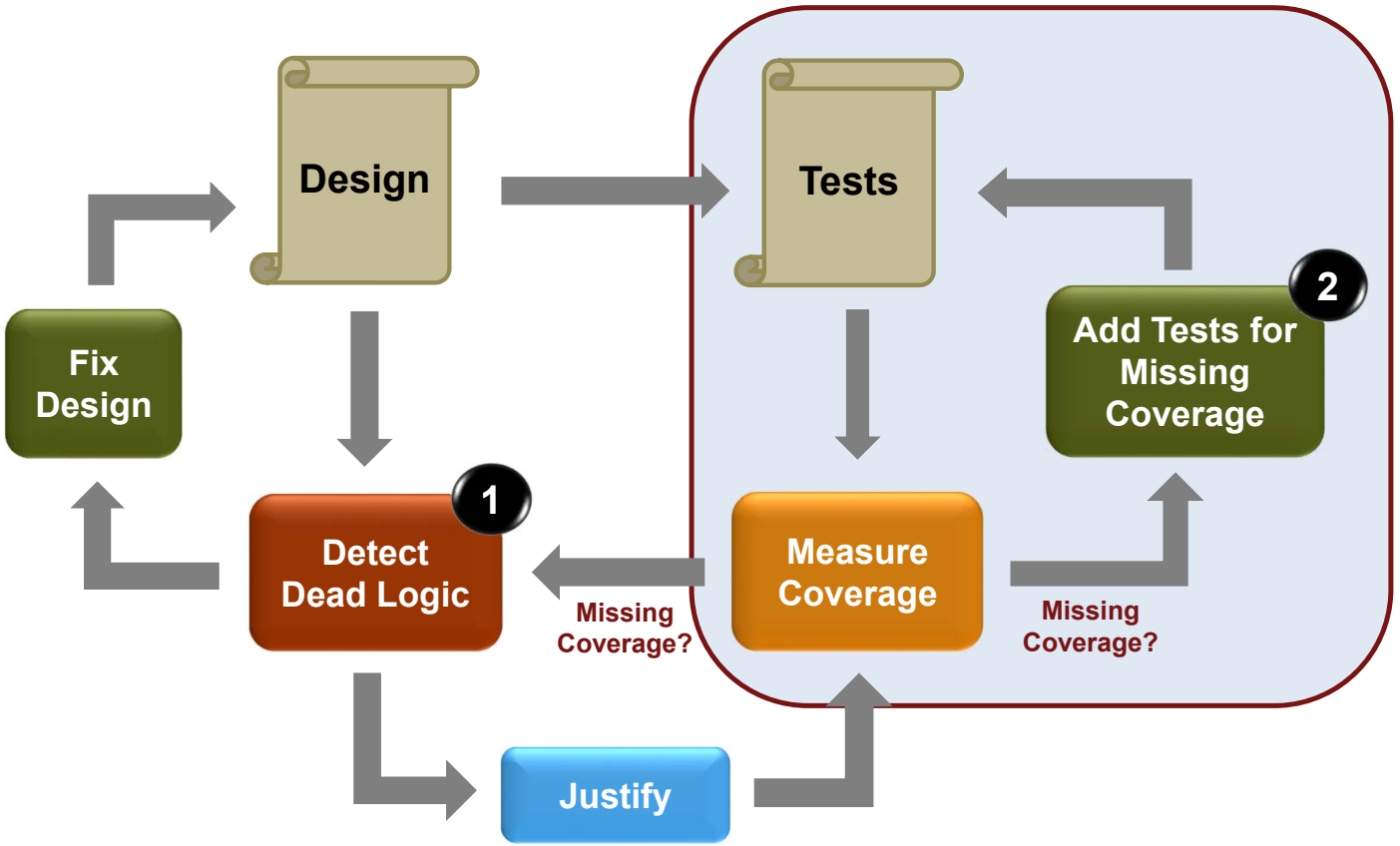
Simulink Coverage Analysis



Simulink Design Verifier



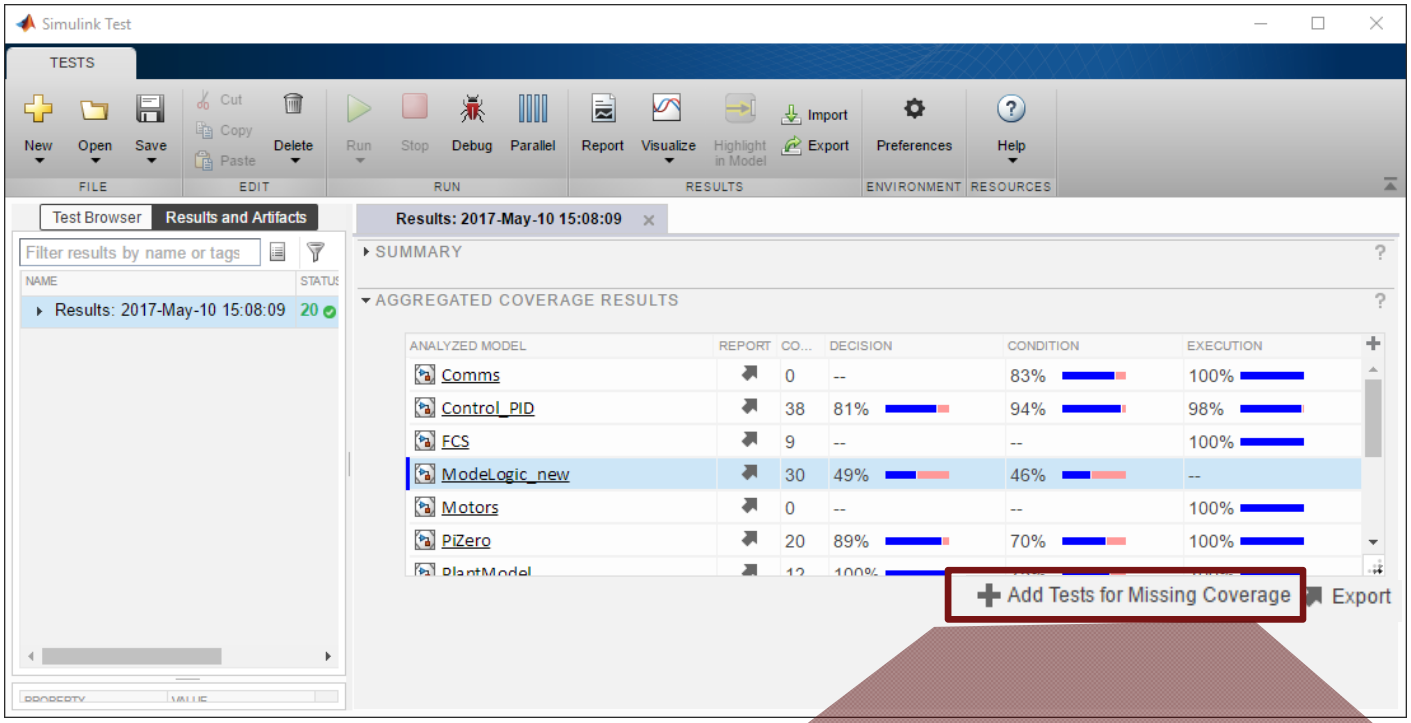
Addressing Missing Coverage Workflow

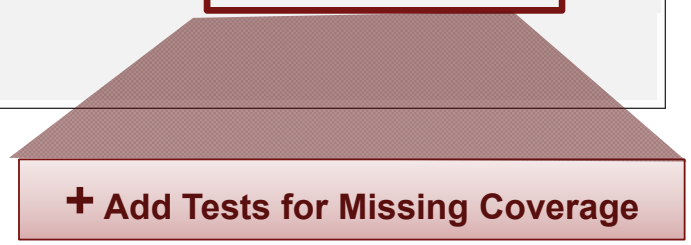


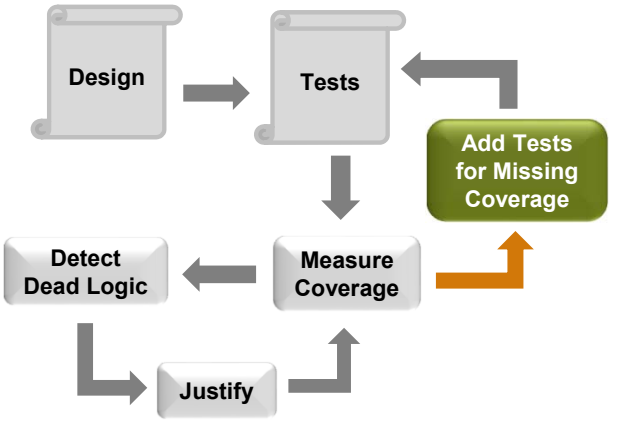
Simulink Design Verifier:

1. Dead Logic Detection
2. Test Generation

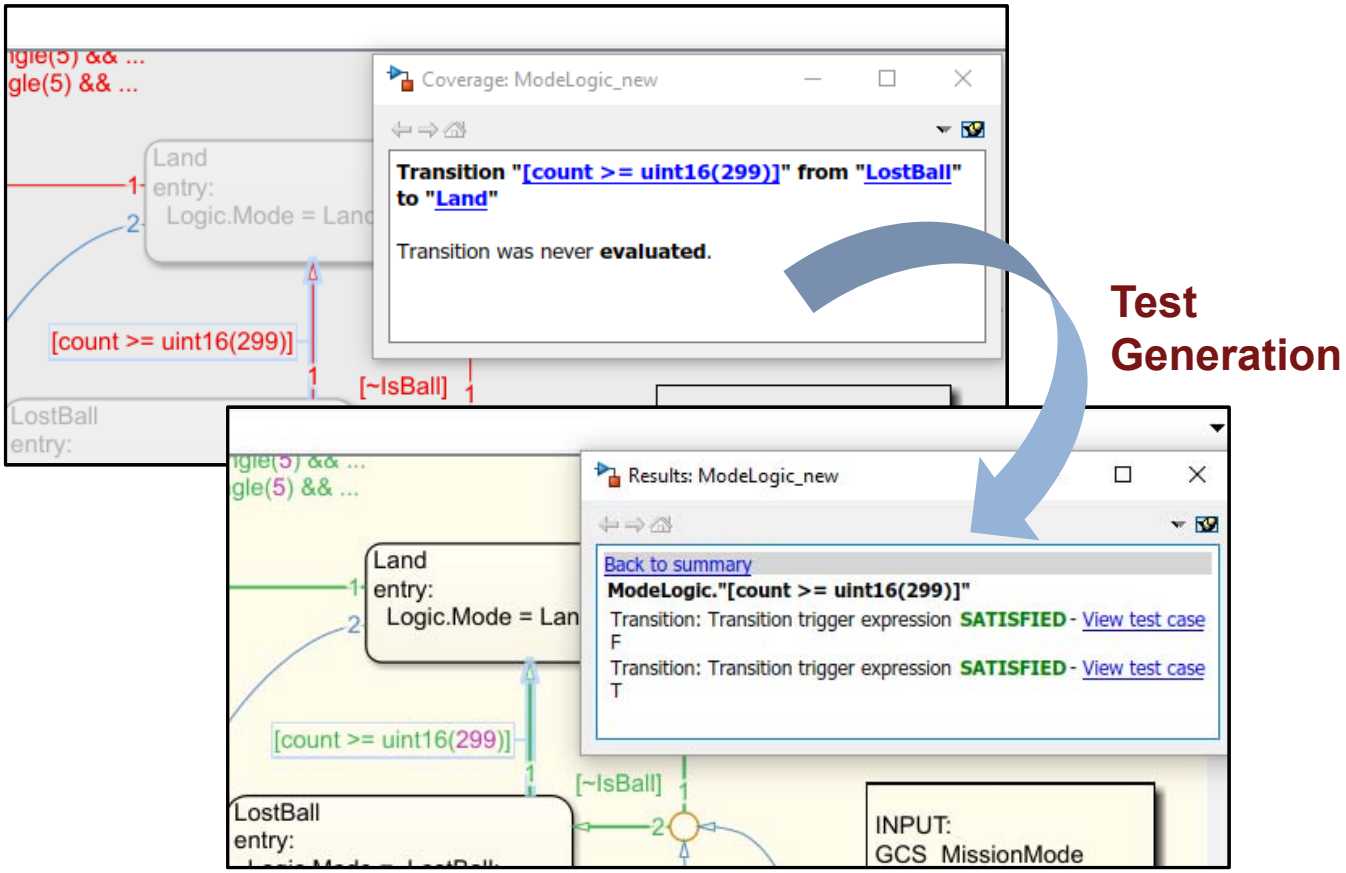
Generate Tests for Missing Coverage



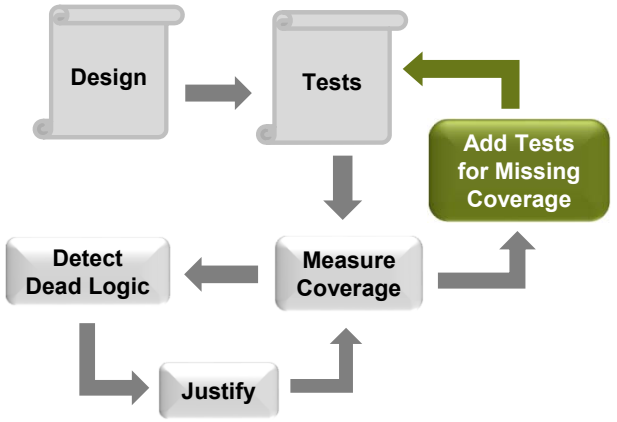




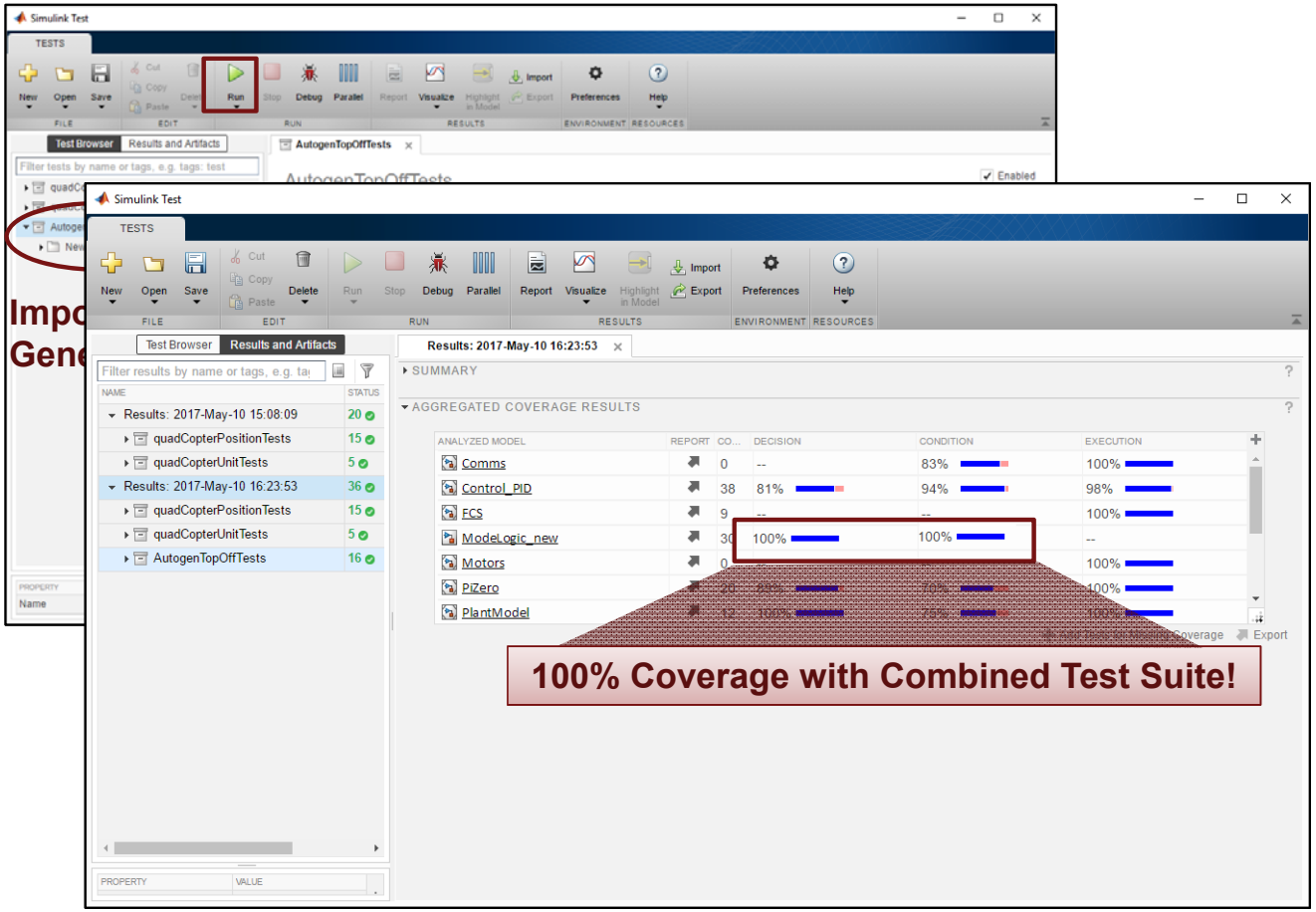
Generate Tests for Missing Coverage

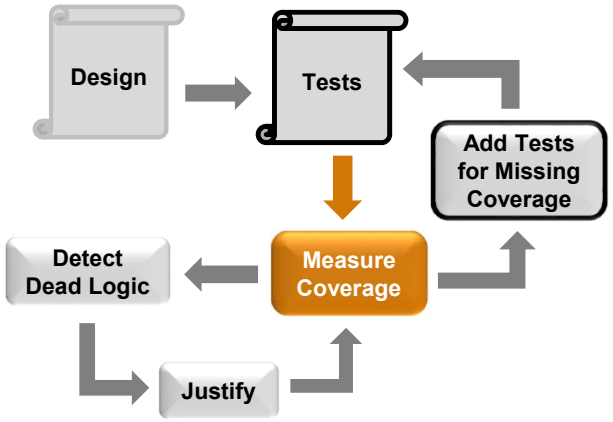


Test Generation



Final Coverage Results





Quadcopter Verification & Validation Workflow Summary

- **Implement requirements without leaving Simulink**
  - Edit requirements and models together
  - Track implementation and verification of requirements
  - Respond to changes faster
- **Verify requirement changes through regression testing**
  - Find impacted tests through file dependency analysis
  - Update test's pass/fail criteria (when design is correct!)
  - Automate tests (in parallel!) to ensure all tests still pass
- **Measure model coverage from test cases**
  - Identify unreachable design content via dead logic analysis
  - Fix design or justify dead logic when it's acceptable
  - Generate additional tests to help fill coverage gaps

MathWorks V&V Solutions Page: https://www.mathworks.com/solutions/verification-validation.html