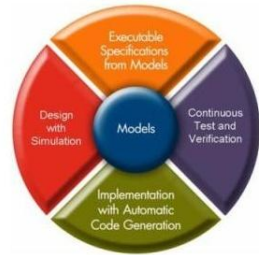


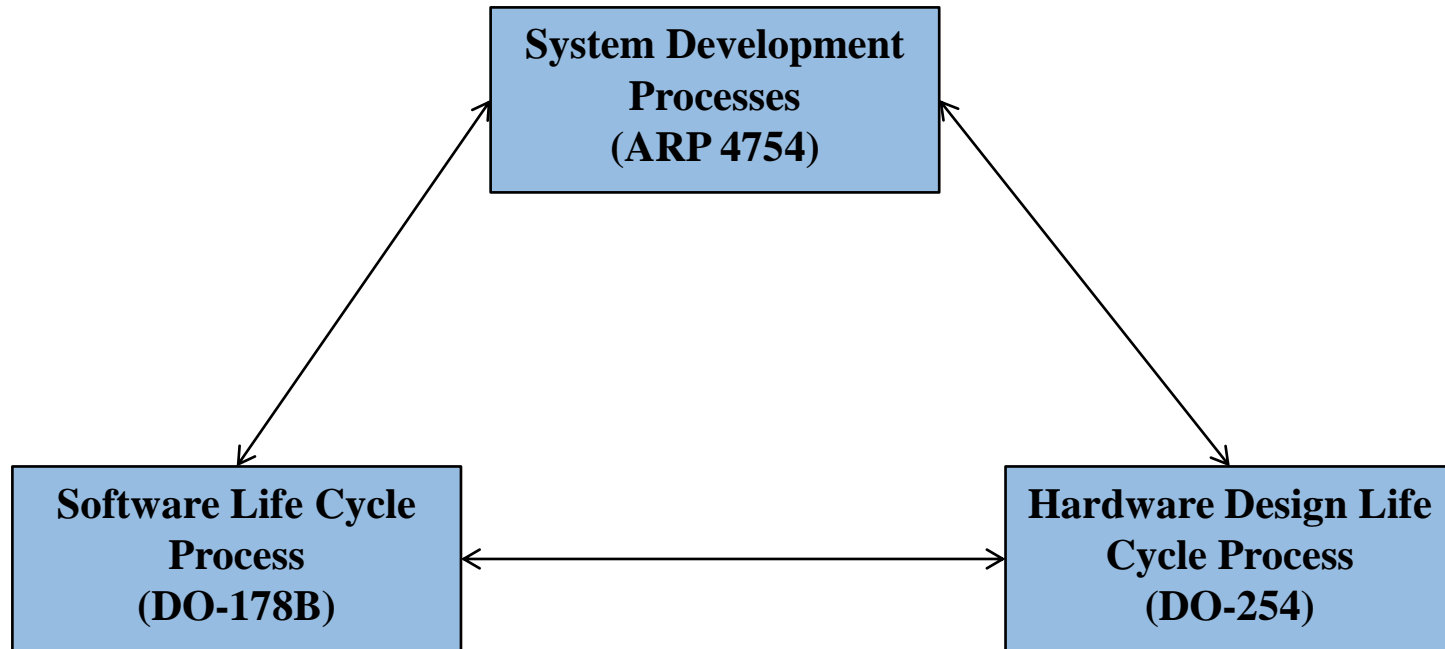
Model-Based Design for High Integrity Software and Hardware

Agenda

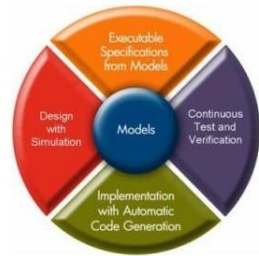


- Relevant standards
- DO workflow – Common Elements
- DO-178B - Software Considerations and Workflows
- DO-254 – Hardware Considerations and Workflows
- Additional Topics

Standards Background

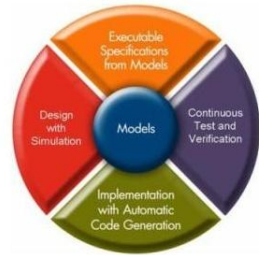


Benefits of Model-Based Design



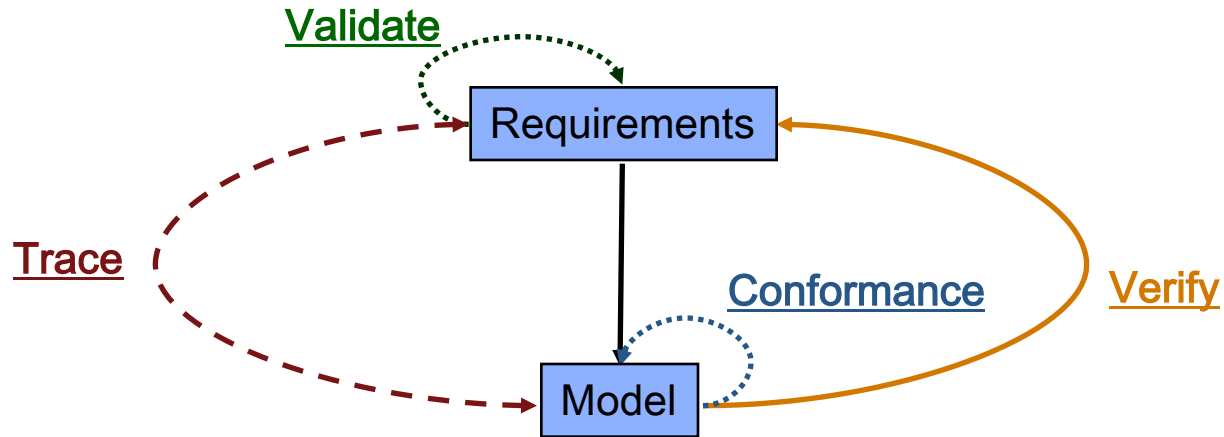
- Use models to validate and verify requirements and designs early in the process
- Re-use tests throughout design cycle
- Automatically generate design and verification artifacts
- Streamline process by qualifying verification tools

Agenda

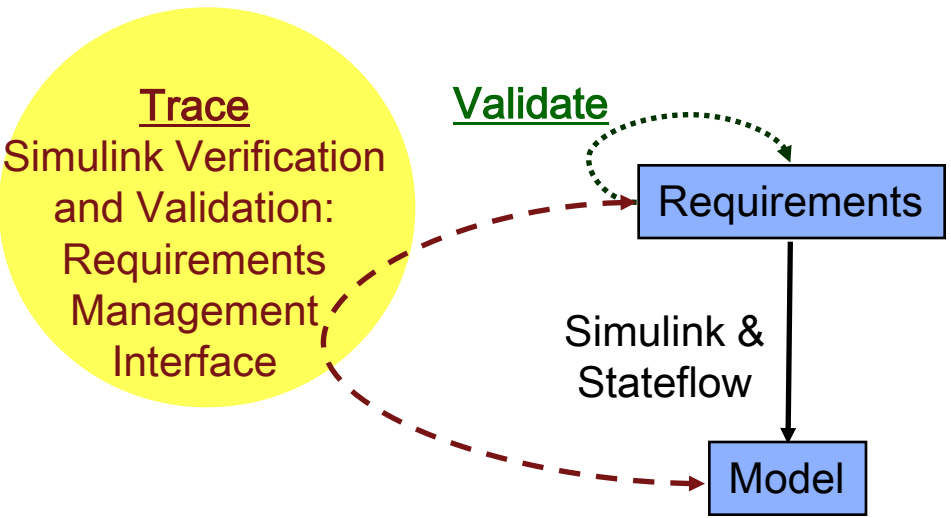


- Relevant standards
- DO workflow – Common Elements
- DO-178B - Software Considerations and Workflows
- DO-254 – Hardware Considerations and Workflows
- Additional Topics

DO Workflow Example



DO Workflow Example

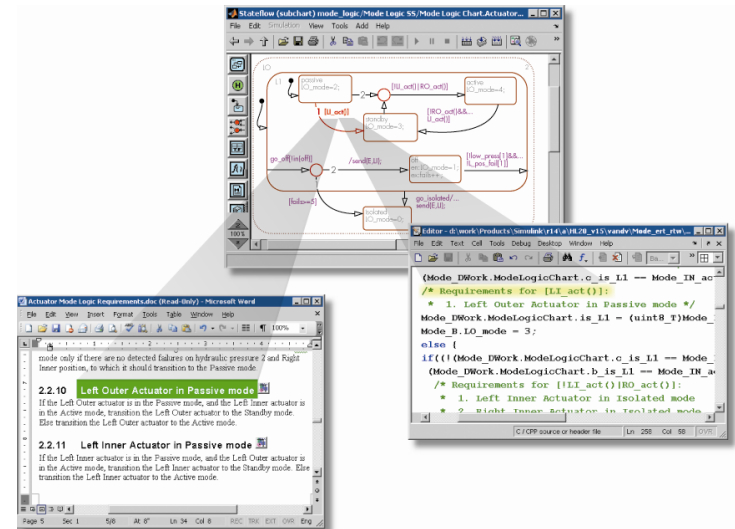


Requirements linking and traceability

DO-178B DO-254

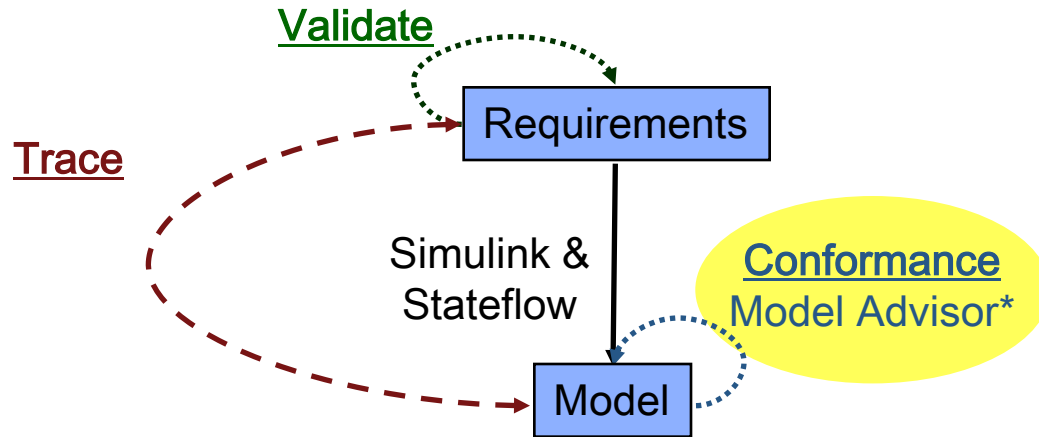
- Bi-directional linking with requirements
 - For Simulink and Stateflow
 - Requirements consistency checks
 - Extensibility API
 - Report generation

- Links to Documents and Requirements Management Packages.



IBM DOORS
 ReqTracer
 Microsoft Word
 Microsoft Excel
 PDF
 HTML

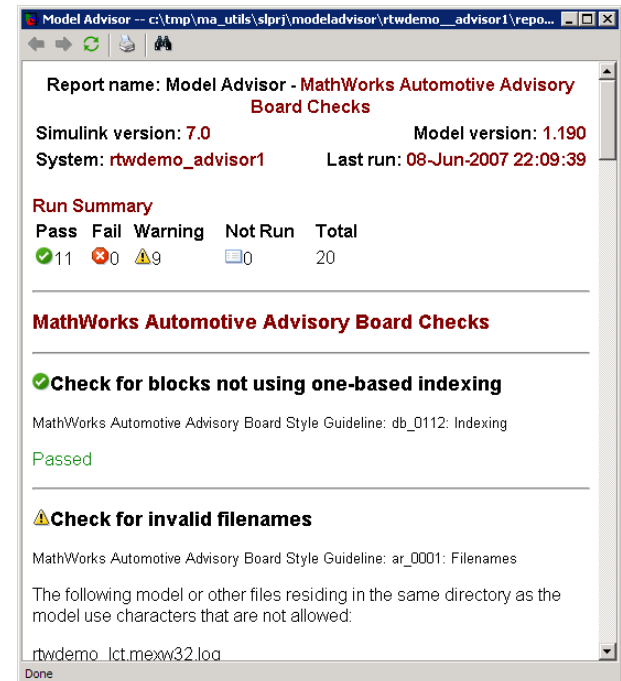
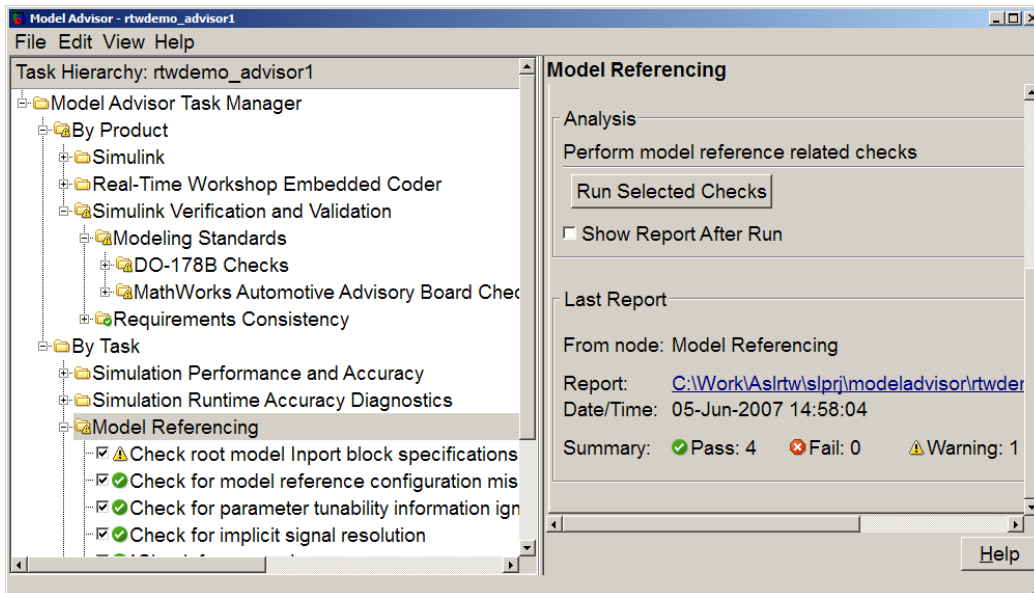
DO Workflow



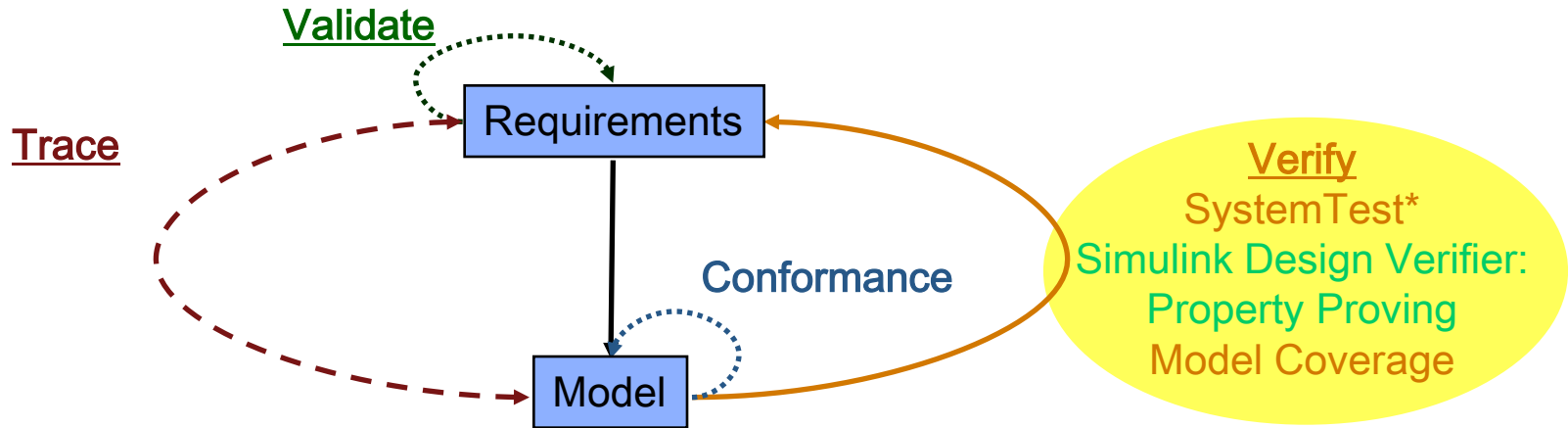
* DO Qualifiable Tool

Simulink Model Advisor

- Model Advisor is used to
 - Enforce model standards and best practices
 - Detect modeling and code generation issues
 - Pre-defined sets of checks for DO-178B and MAAB Style Guides
 - Automated report generation



DO Workflow Example

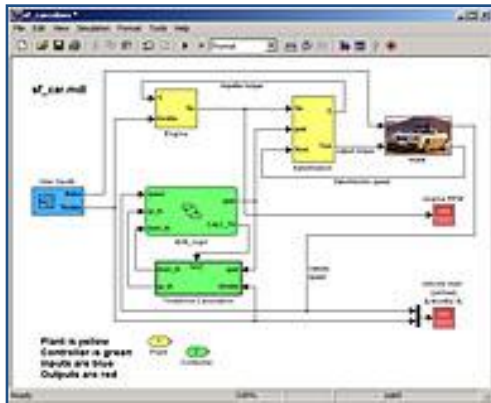


* DO Qualifiable Tool

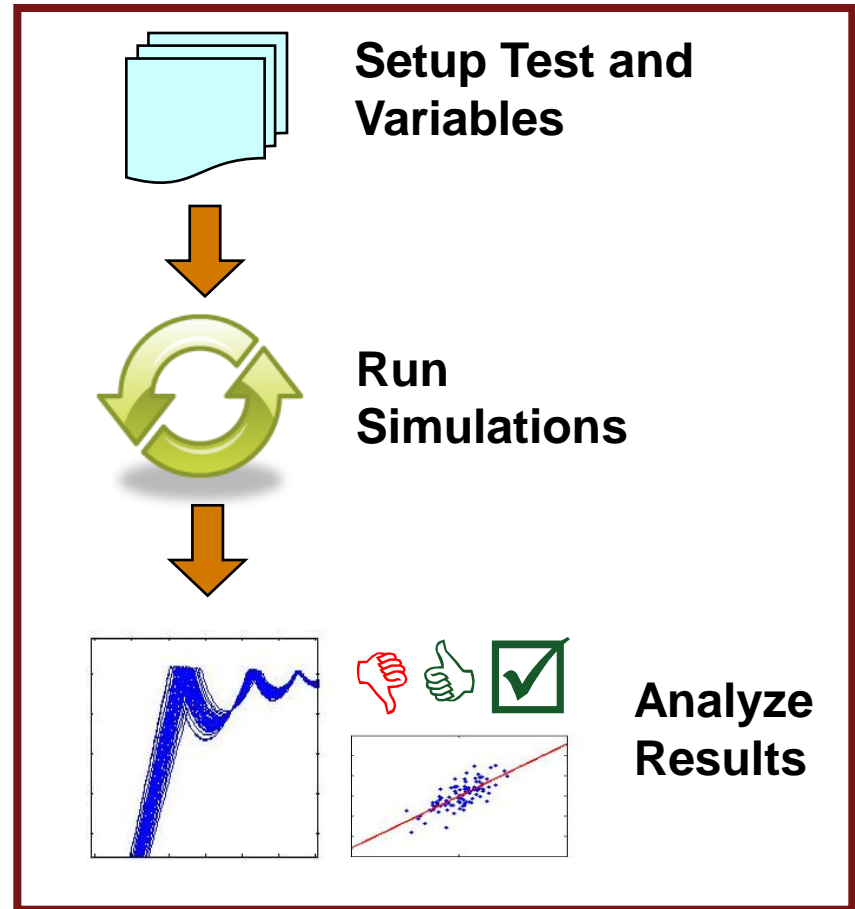
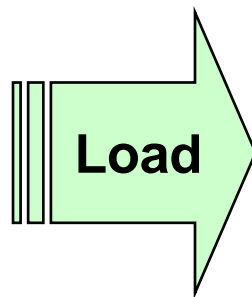
SystemTest

DO-178B DO-254

- Manage tests and analyze results for system verification and validation



Simulink System Model

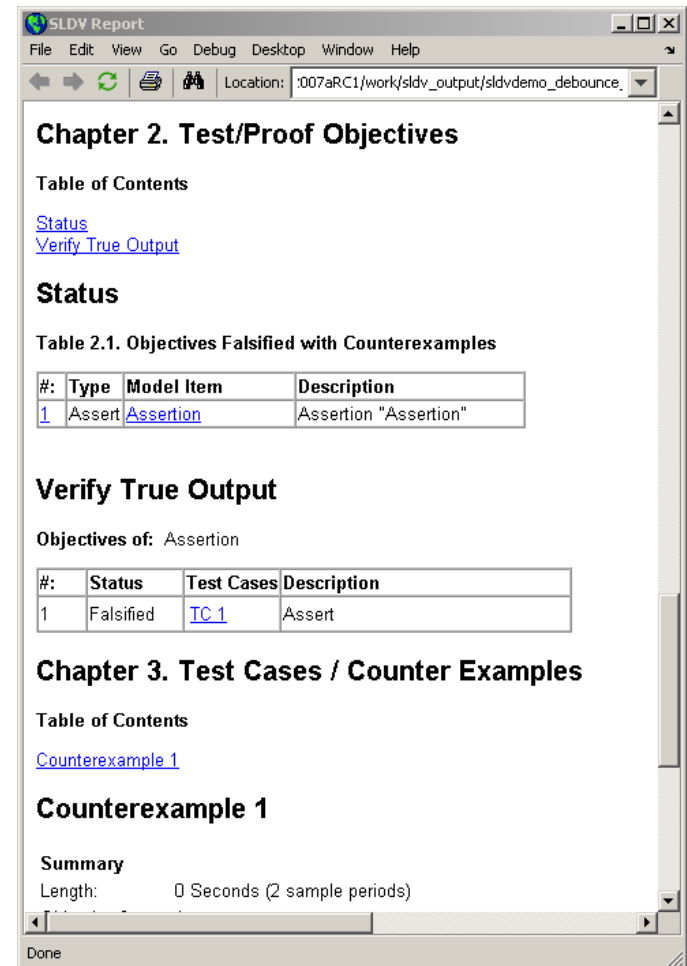


SystemTest

Simulink Design Verifier

Property Proving & Test Generation

- Property Proving
 - Functional testing
 - Property proving
 - Generates an example of a violation
 - Produces detailed analysis reports
- Test Generation
 - Automatically generates test vectors for model coverage
 - Detects unreachable states
 - Saves test vectors and generates report
- Uses formal methods, not simulation



SLDV Report

File Edit View Go Debug Desktop Window Help

Location: :007aRC1/work/sldv_output/sldvdemo_debounce

Chapter 2. Test/Proof Objectives

Table of Contents

[Status](#)
[Verify True Output](#)

Status

Table 2.1. Objectives Falsified with Counterexamples

#:	Type	Model Item	Description
1	Assert	Assertion	Assertion "Assertion"

Verify True Output

Objectives of: Assertion

#:	Status	Test Cases	Description
1	Falsified	TC 1	Assert

Chapter 3. Test Cases / Counter Examples

Table of Contents

[Counterexample 1](#)

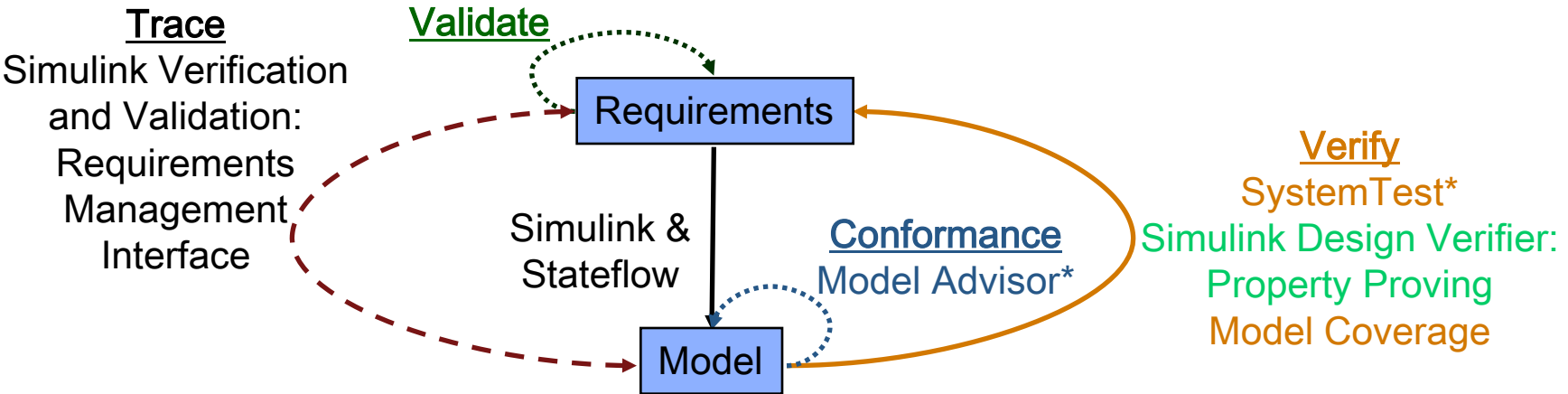
Counterexample 1

Summary

Length: 0 Seconds (2 sample periods)

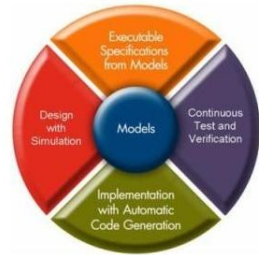
Done

DO Workflow Example



* DO Qualifiable Tool

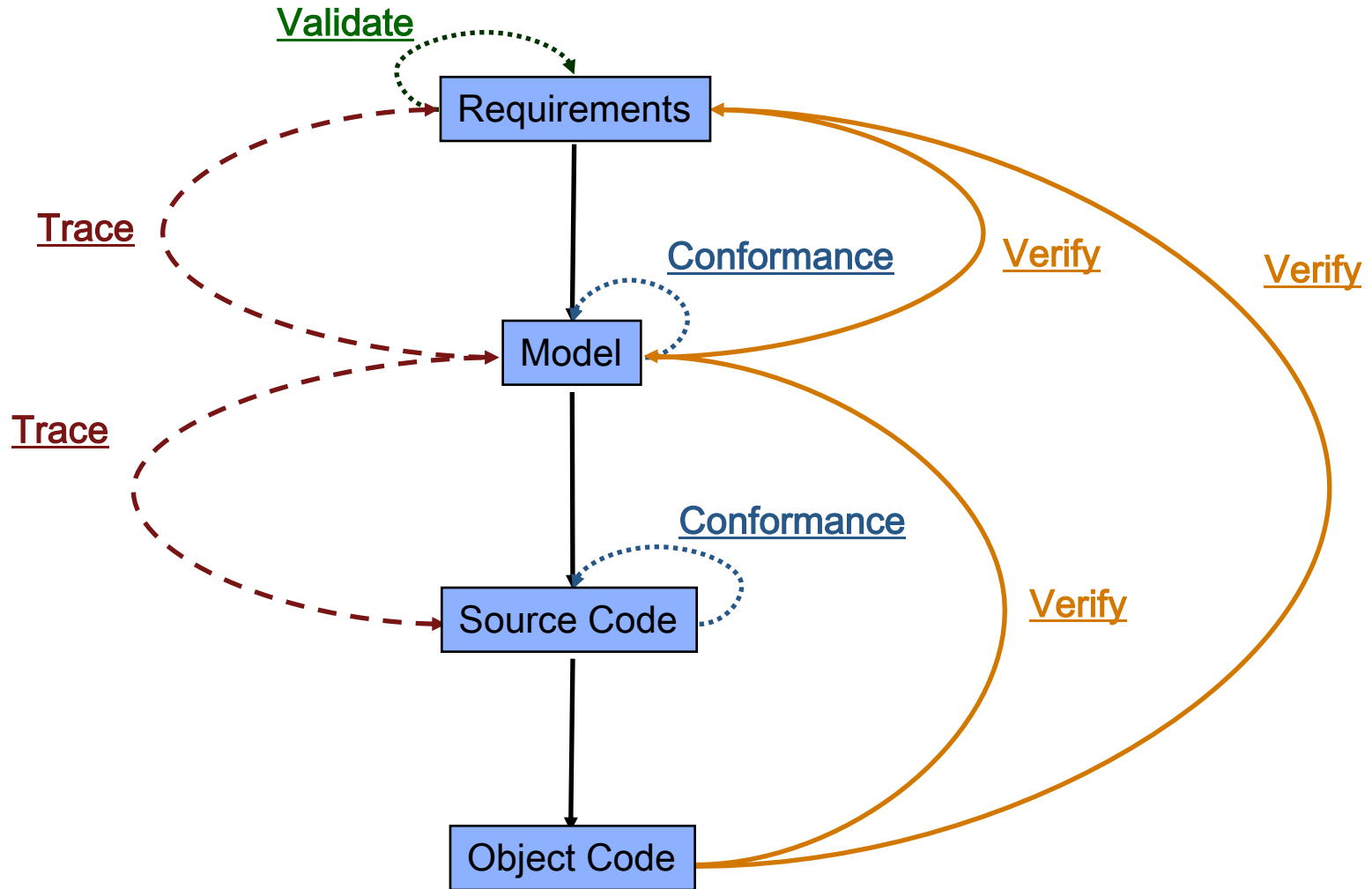
Agenda



- Relevant standards
- DO workflow – Common Elements
- DO-178B - Software Considerations and Workflows
- DO-254 – Hardware Considerations and Workflows
- Additional Topics

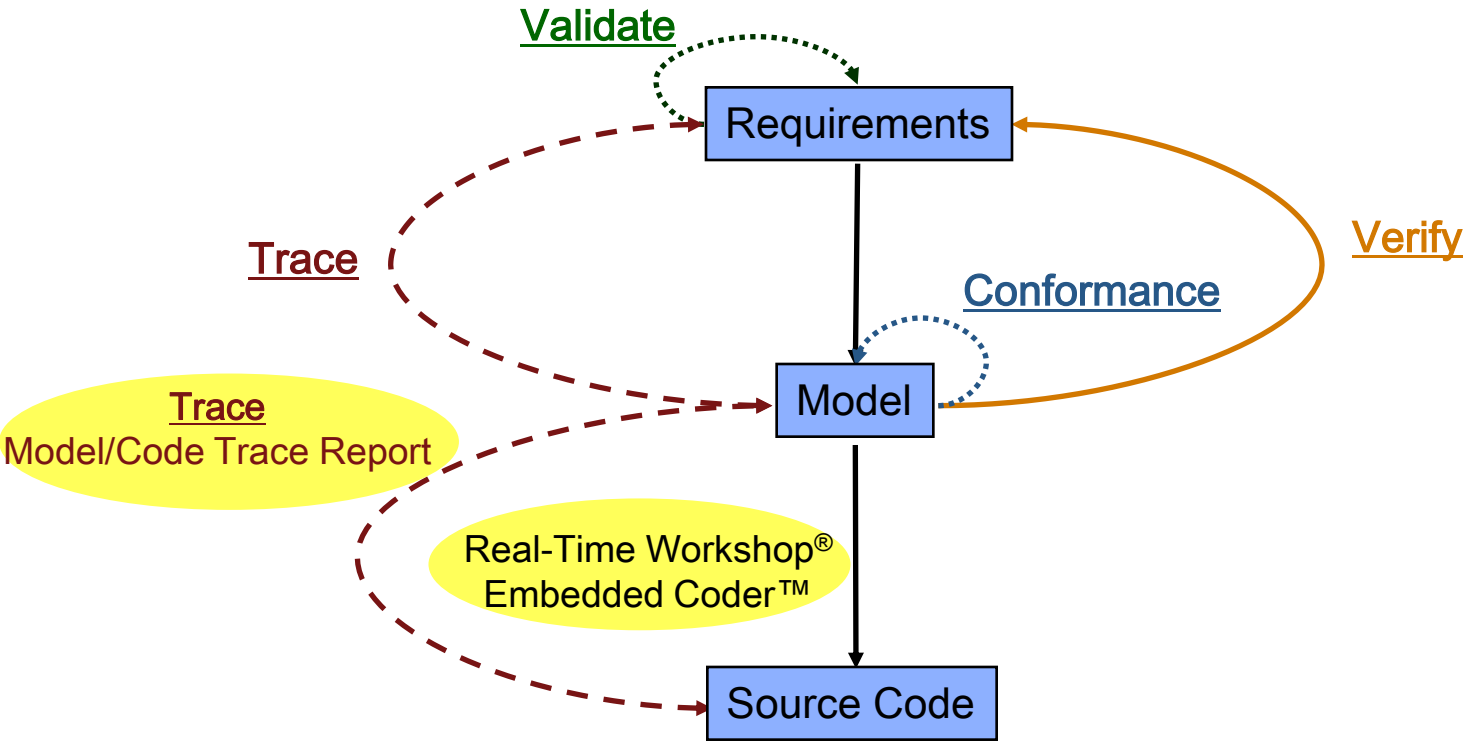
DO-178B Workflow Example

DO-178B DO-254



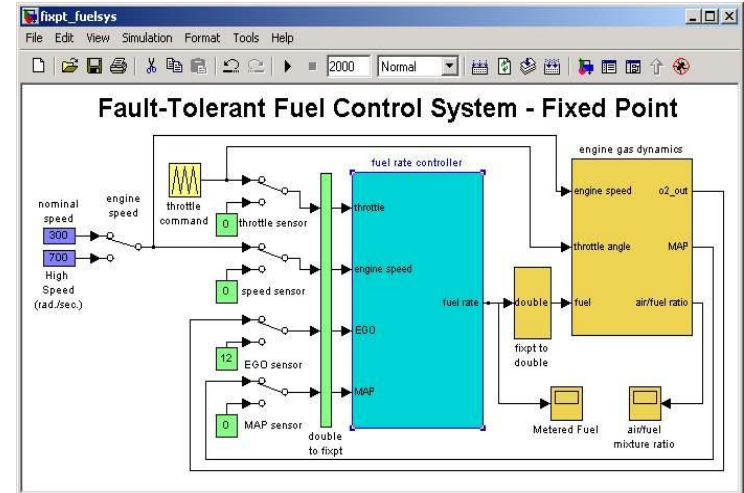
DO-178B Workflow Example

DO-178B DO-254



Real-Time Workshop® Embedded Coder

- Automatically generates C code from Simulink® and Stateflow® models
- Code is ANSI/ISO-C compliant



```

/* Switch: '<S20>/Switch' */
if (fixpt_fuelsys_B.throt_fail_o) {

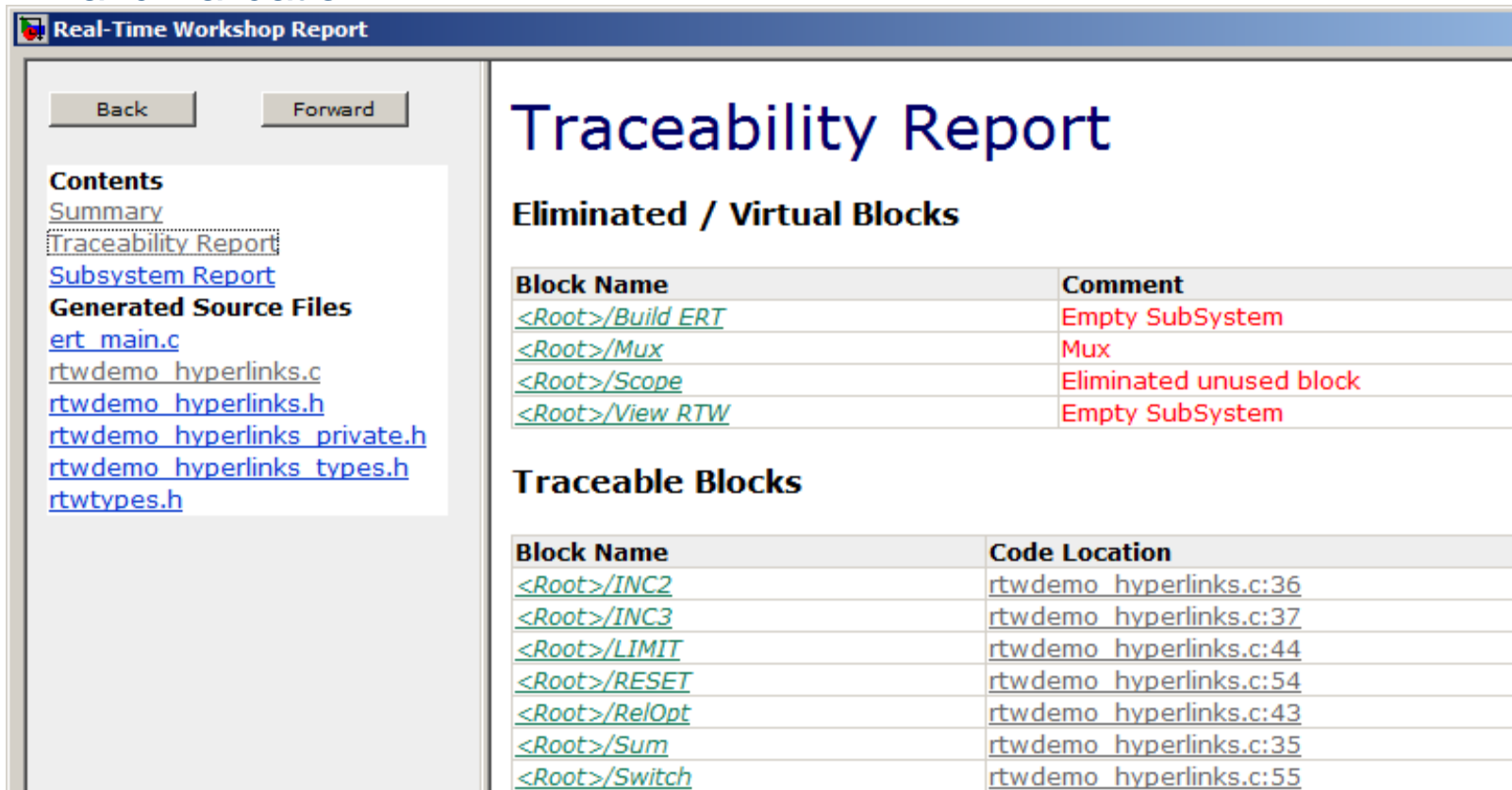
    /* Output and update for atomic system: '<S20>/Throttle Estimate' */

    /* Lookup2D Block: '<S31>/Thrott Estimation Table (2-D)' */
    * Input0 Data Type: Fixed Point S16 2^-1
    * Input1 Data Type: Fixed Point S16 2^-14
    * Output0 Data Type: Fixed Point S16 2^-5
    
```

Model-to-Code and Code-to-Model Traceability

Simulink Verification
and Validation

Real-Time Workshop
Embedded Coder



Real-Time Workshop Report

Back Forward

Contents
[Summary](#)
[Traceability Report](#)
[Subsystem Report](#)
Generated Source Files
[ert_main.c](#)
[rtwdemo_hyperlinks.c](#)
[rtwdemo_hyperlinks.h](#)
[rtwdemo_hyperlinks_private.h](#)
[rtwdemo_hyperlinks_types.h](#)
[rtwtypes.h](#)

Traceability Report

Eliminated / Virtual Blocks

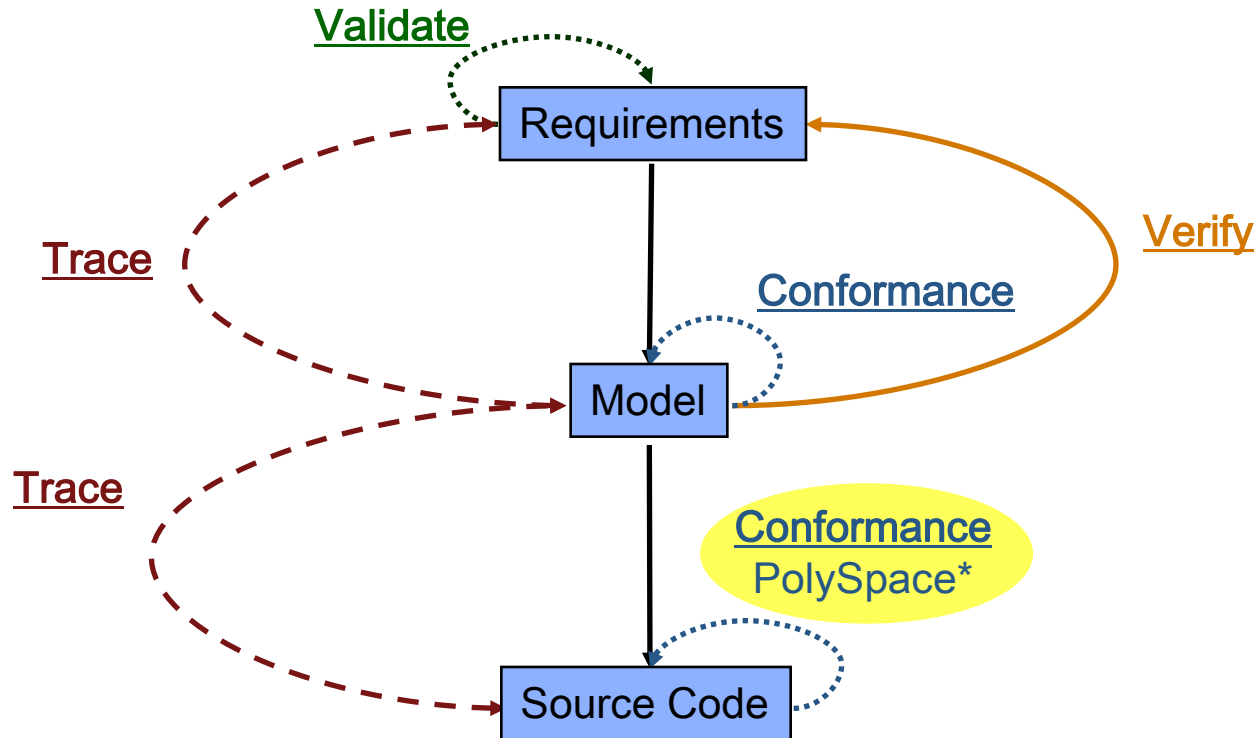
Block Name	Comment
<Root>/Build ERT	Empty SubSystem
<Root>/Mux	Mux
<Root>/Scope	Eliminated unused block
<Root>/View RTW	Empty SubSystem

Traceable Blocks

Block Name	Code Location
<Root>/INC2	rtwdemo_hyperlinks.c:36
<Root>/INC3	rtwdemo_hyperlinks.c:37
<Root>/LIMIT	rtwdemo_hyperlinks.c:44
<Root>/RESET	rtwdemo_hyperlinks.c:54
<Root>/RelOpt	rtwdemo_hyperlinks.c:43
<Root>/Sum	rtwdemo_hyperlinks.c:35
<Root>/Switch	rtwdemo_hyperlinks.c:55

DO-178B Workflow Example

DO-178B DO-254



PolySpace

- Verification of C/C++ and Ada code
- Detects run-time errors
- Streamlines high integrity DO-178B workflows
 - Rule checking features (MISRA-C and JSF++)
 - Source code color scheme
 - DO-178B artifact generation capabilities
 - Qualification kit available

P
r
o
v
e
n

```

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

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

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

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

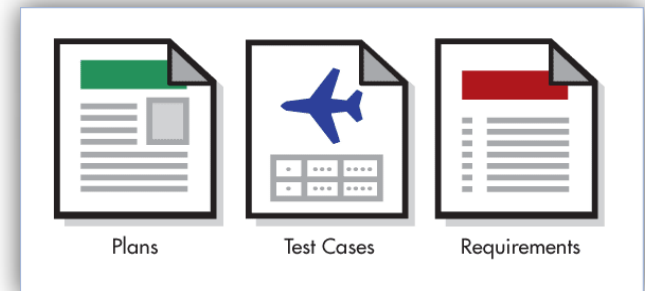
  if ((0 < i) && (i <= 100)) {
    p = p - i;
    *p = 5;
  }
}
          
```

Green: reliable

Red: faulty

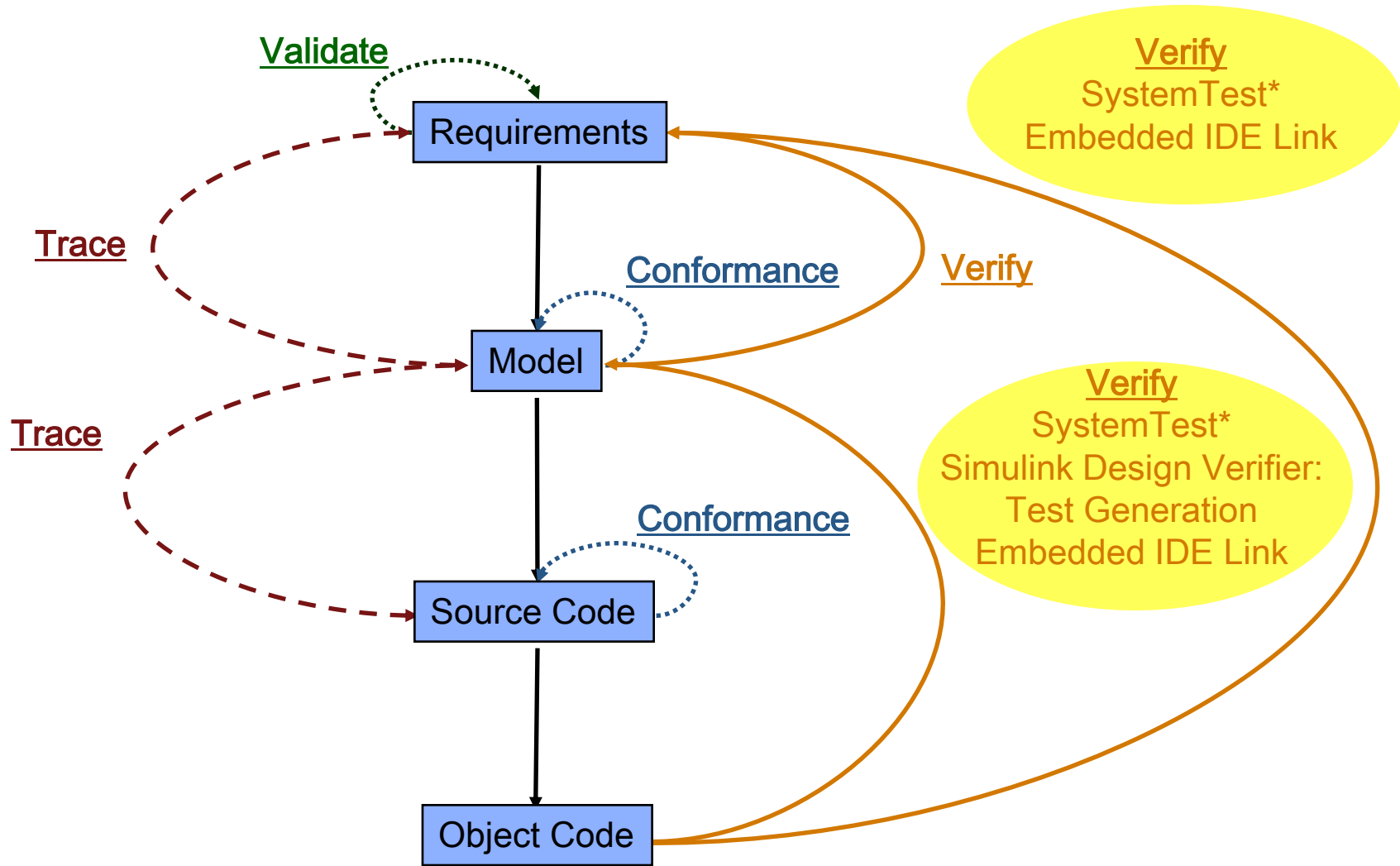
Gray: dead

Orange: unproven



DO-178B Workflow Example

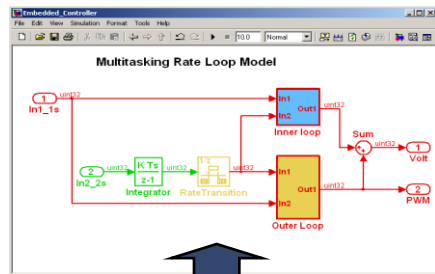
DO-178B DO-254



Processor-in-the-Loop Testing

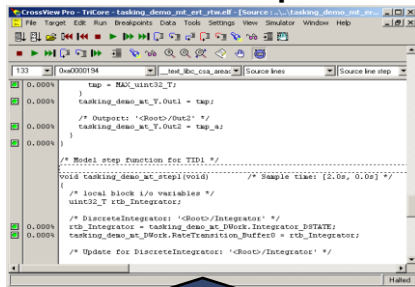
Embedded IDE Link

Simulink:

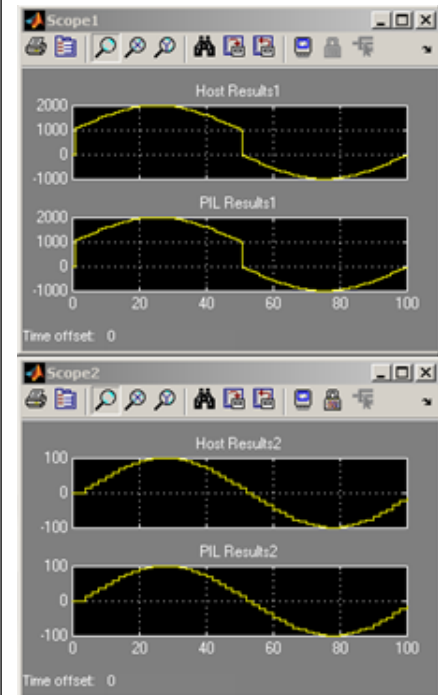
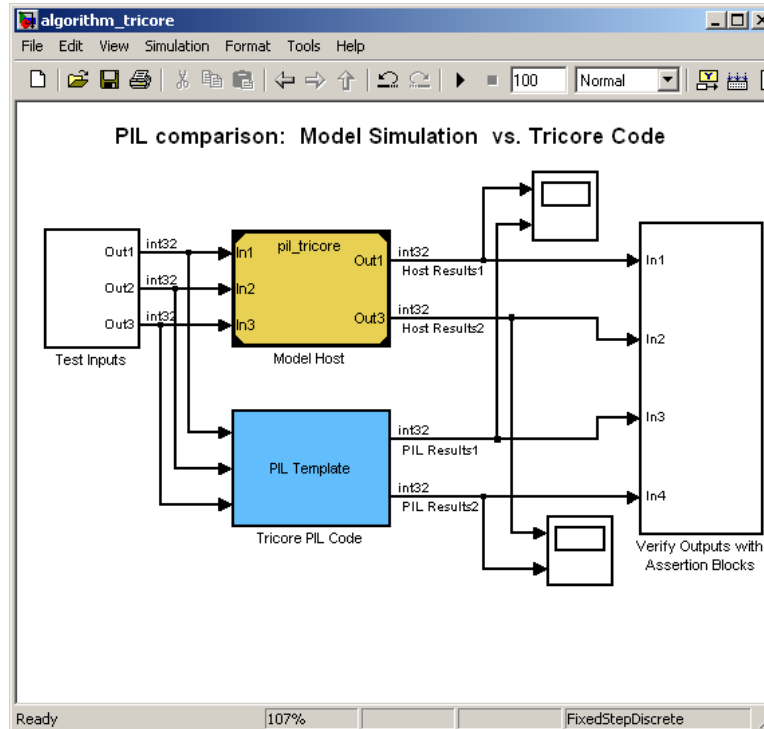


- Model in simulation and code on the processor running in parallel

Real-Time Workshop and IDE

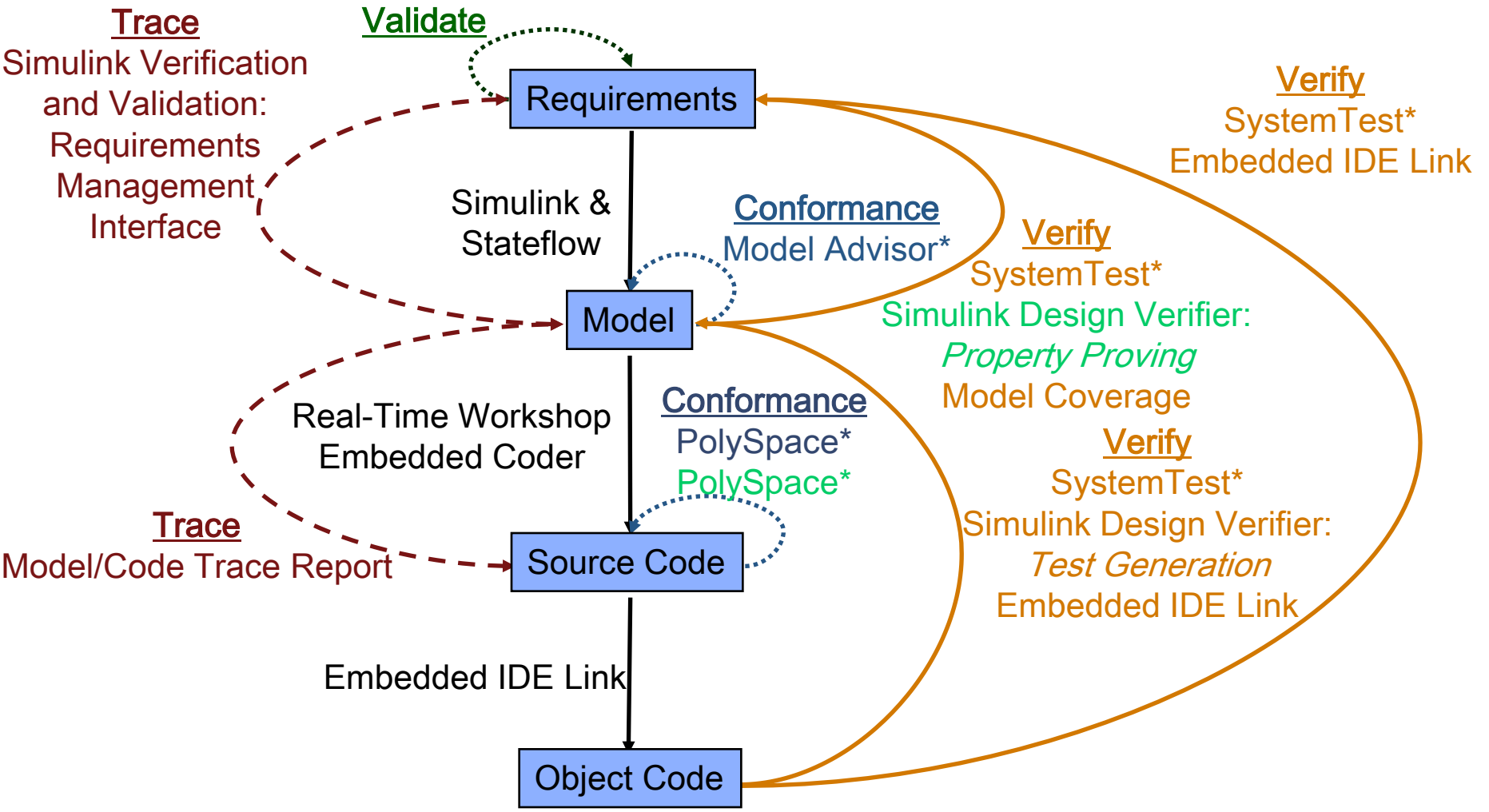


ECU:



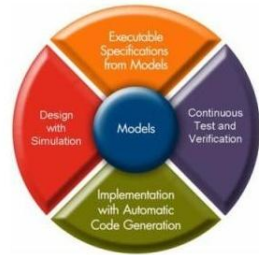
PIL also provides execution profiling, code coverage reports, and interactive debugging.

DO-178B Workflow Summary



* DO Qualifiable Tool

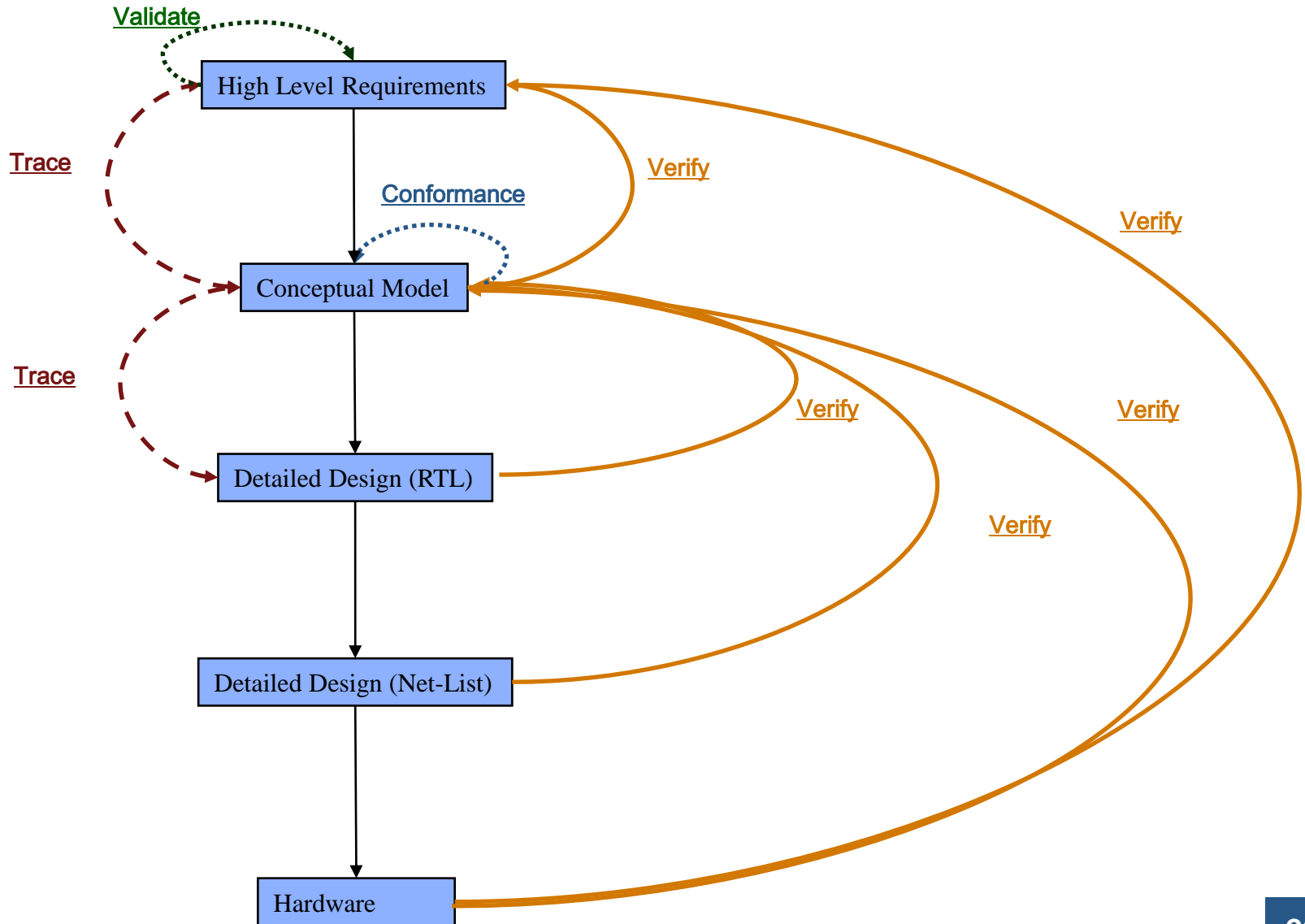
Agenda



- Relevant standards
- DO workflow – Common Elements
- DO-178B - Software Considerations and Workflows
- DO-254 – Hardware Considerations and Workflows
- Additional Topics

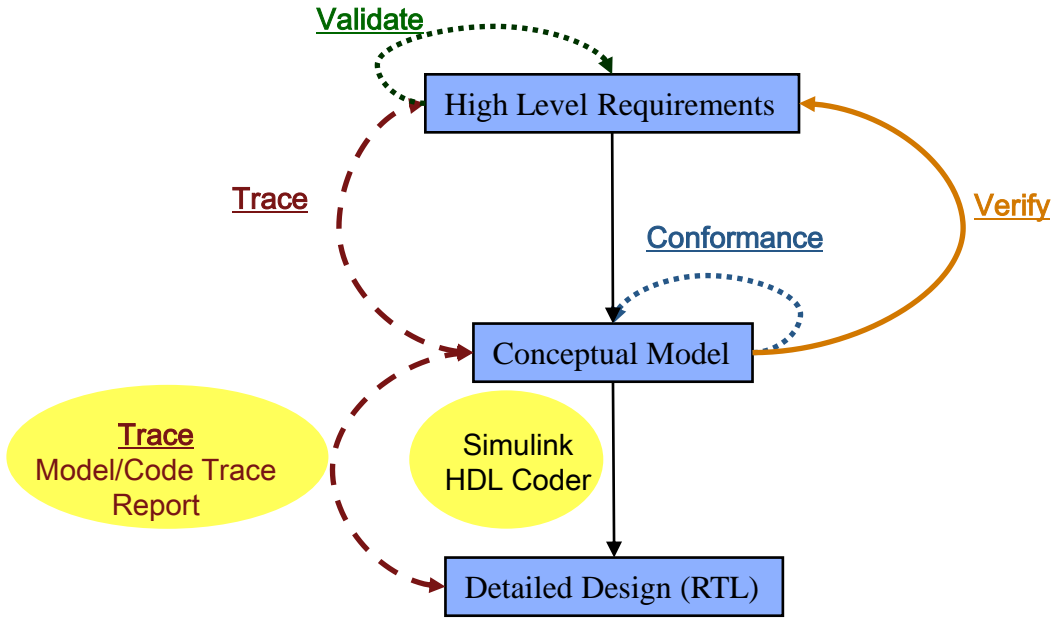
DO-254 Workflow Example

DO-178B **DO-254**



DO-254 Workflow Example

DO-178B DO-254



HDL Code Generation with Simulink HDL Coder

DO-178B

DO-254

- Simulink HDL Coder
 - Generate behavioral HDL
 - Readable and traceable to requirements
 - Target-Independent
 - Bit Accurate/Cycle True
 - Customizable via options and Control Files

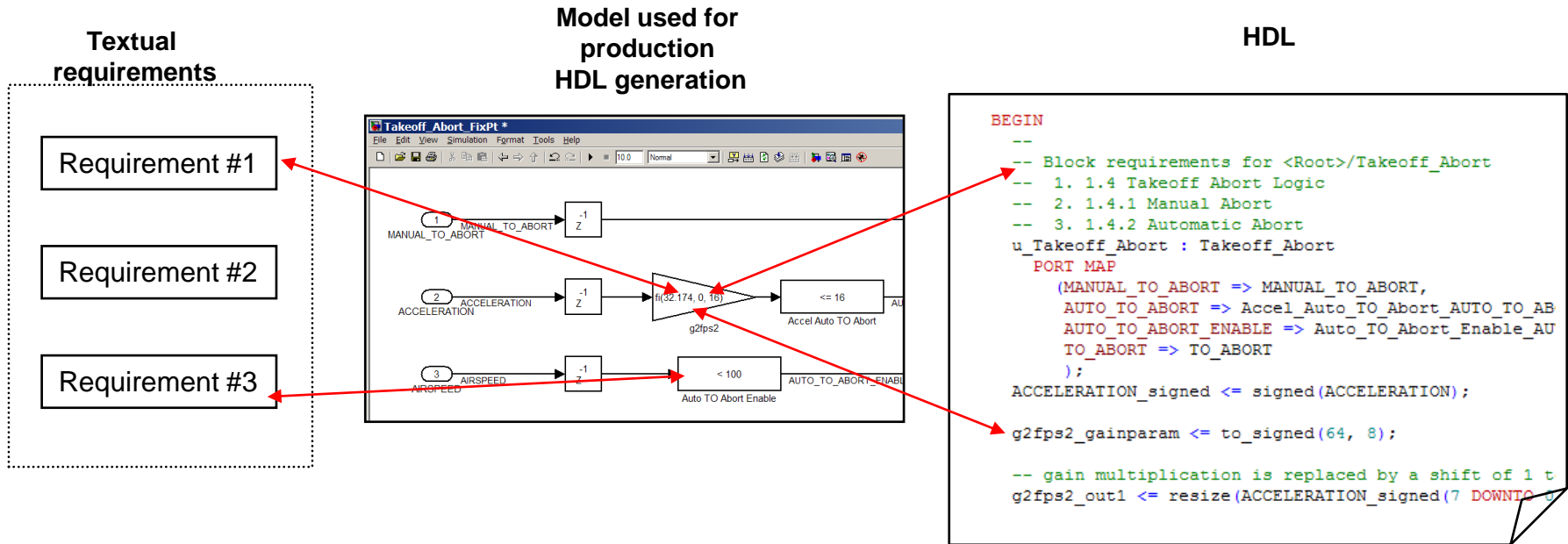
- Full model support
 - Simulink (datapath)
 - Stateflow® (control logic)
 - Embedded MATLAB

The image illustrates the HDL code generation process. It starts with a Simulink model of a 16-QAM transmitter and receiver. The 'Code generation control file' dialog is used to configure the target (FPGA), language (Verilog), and other options. The final output is a Verilog code file, which includes comments about the file name, generation date, and target hardware (MATLAB 7.5 and Simulink HDL Coder 1.2). The code defines a 16-QAM transmitter and receiver, including a library of 16-QAM symbols and a stateful transmitter block.

Model-to-HDL and HDL-to-Model Traceability

Simulink
Verification and
Validation

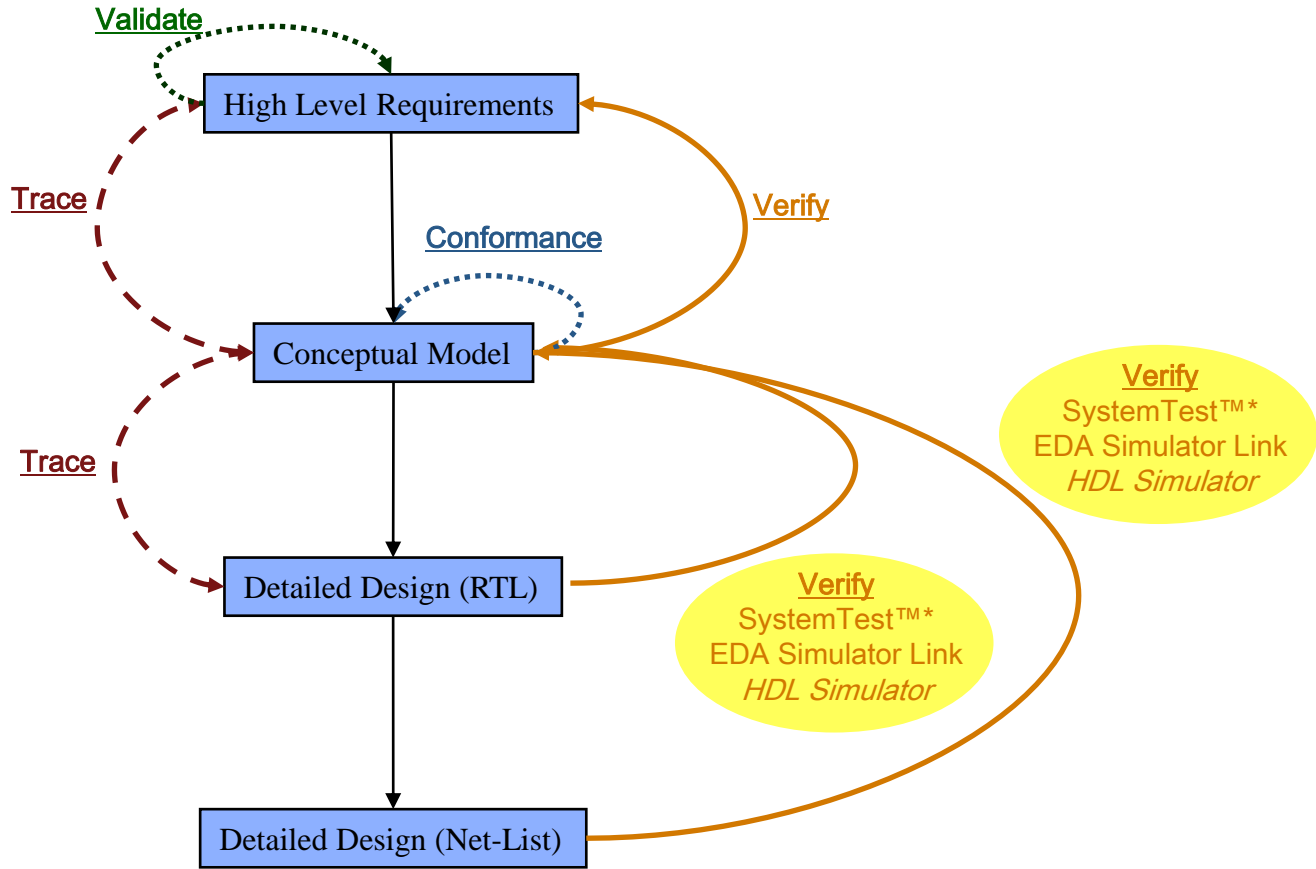
Simulink HDL Coder



- Use the **Traceability Report** section of the Simulink HDL Coder HDL generation report to review mapping

DO-254 Workflow Example

DO-178B **DO-254**

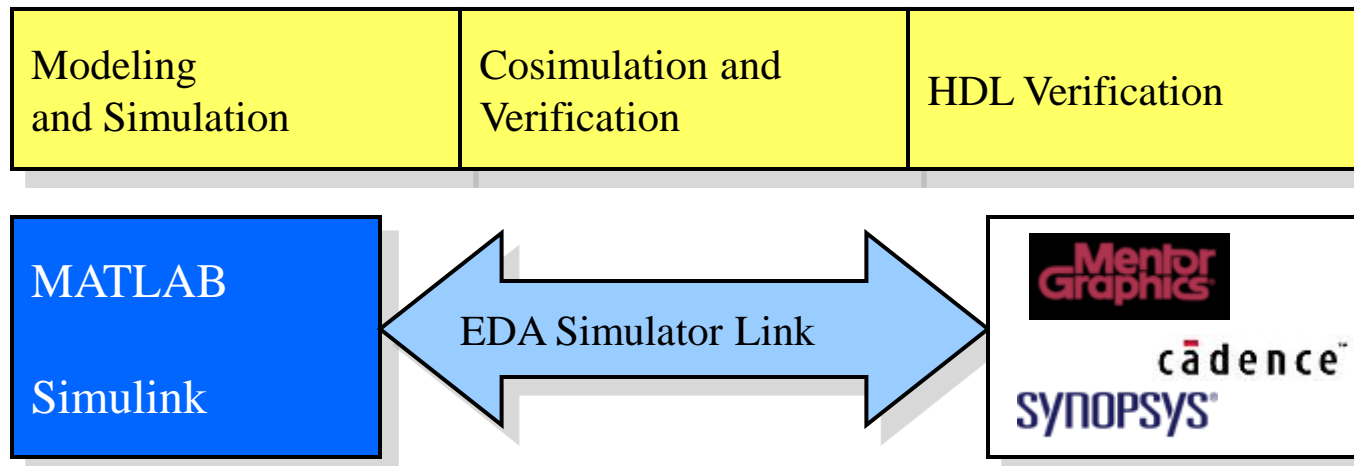


EDA Simulator Link™ block Brings Together Leading Tools for Modeling and HDL Simulation

DO-178B

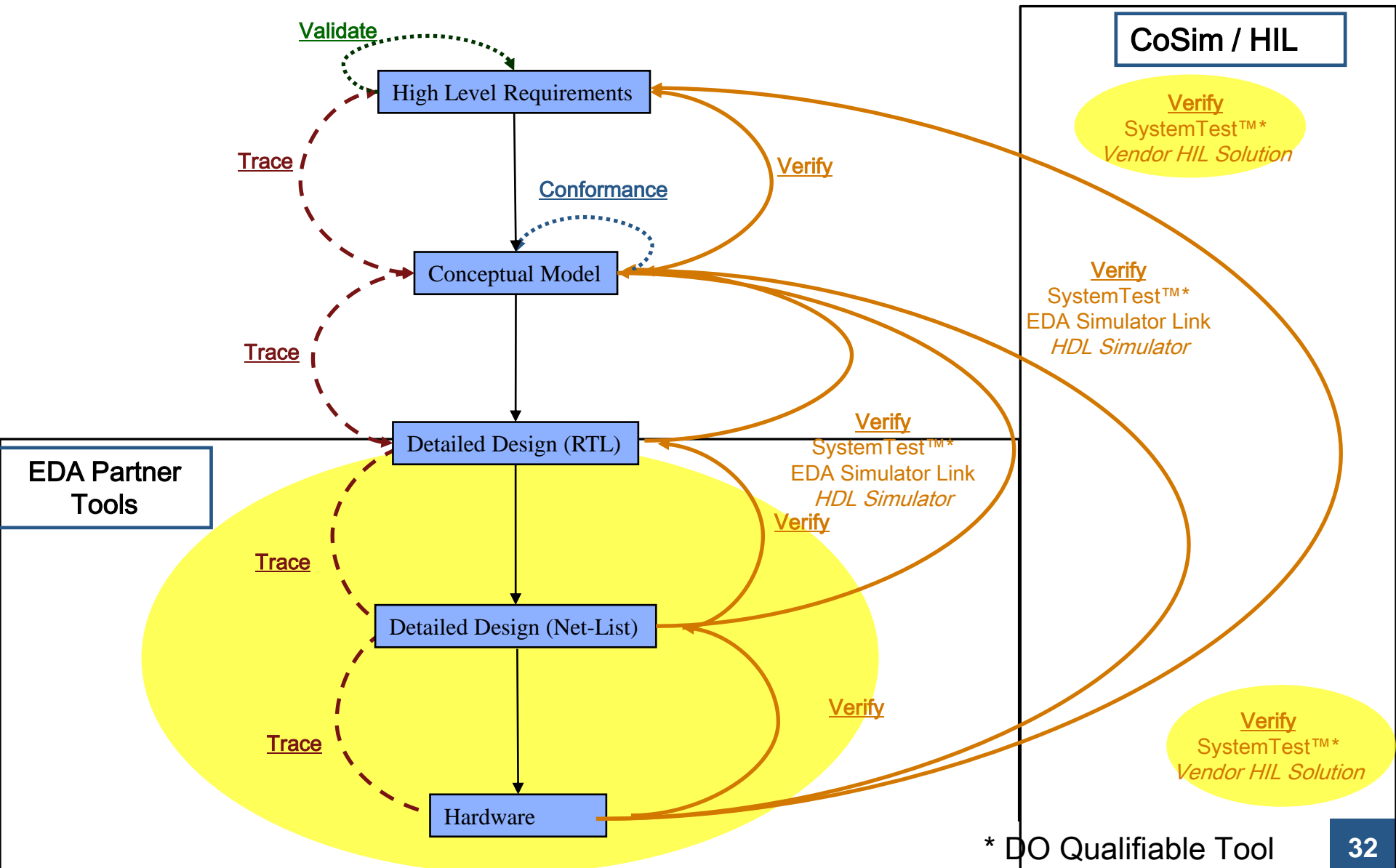
DO-254

- EDA Simulator Link™ block is a fast, bidirectional cosimulation interface for system-level functional verification using ModelSim, Incisive or Discovery
- Benefit: Reuse test environment in the executable specification to verify the implementation.



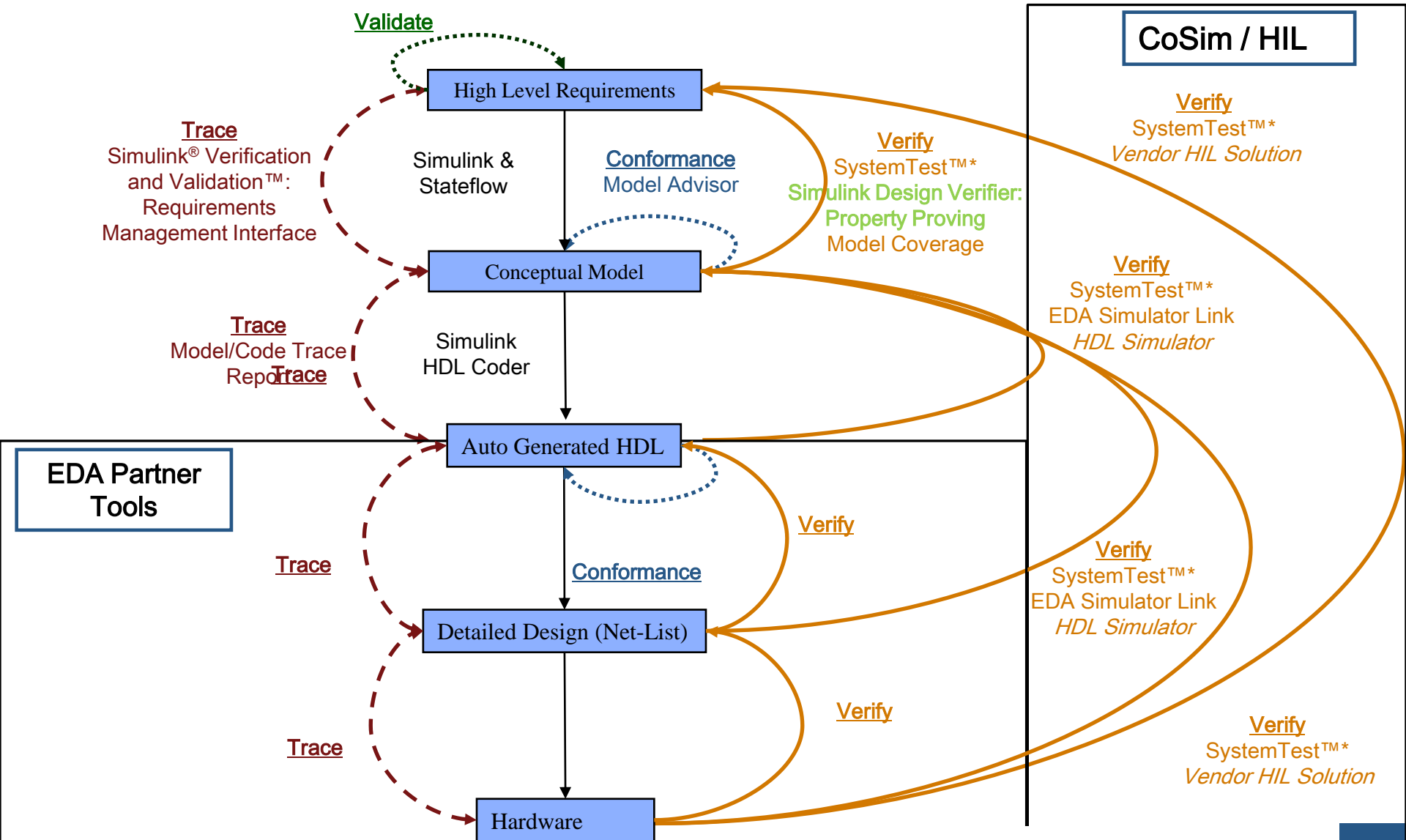
DO-254 Workflow Example

DO-178B **DO-254**

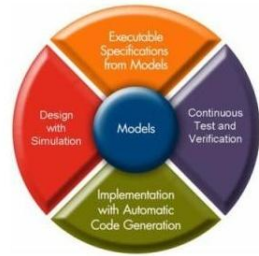


DO-254 Workflow Example (Partners)

DO-178B **DO-254**



Agenda



- Relevant standards
- Benefits of Model-Based Design
- DO-178B - Software Considerations and Workflows
- DO-254 – Hardware Considerations and Workflows
- Additional Topics

DO Qualification Kit

- Tool Qualification Plan and Tool Operational Requirements
- Test case models and code, test procedures, and expected results
- Traceability tables mapping test cases to requirements
- Qualification materials for Simulink verification, validation, and test tools
- Qualification materials for PolySpace code verification tools

