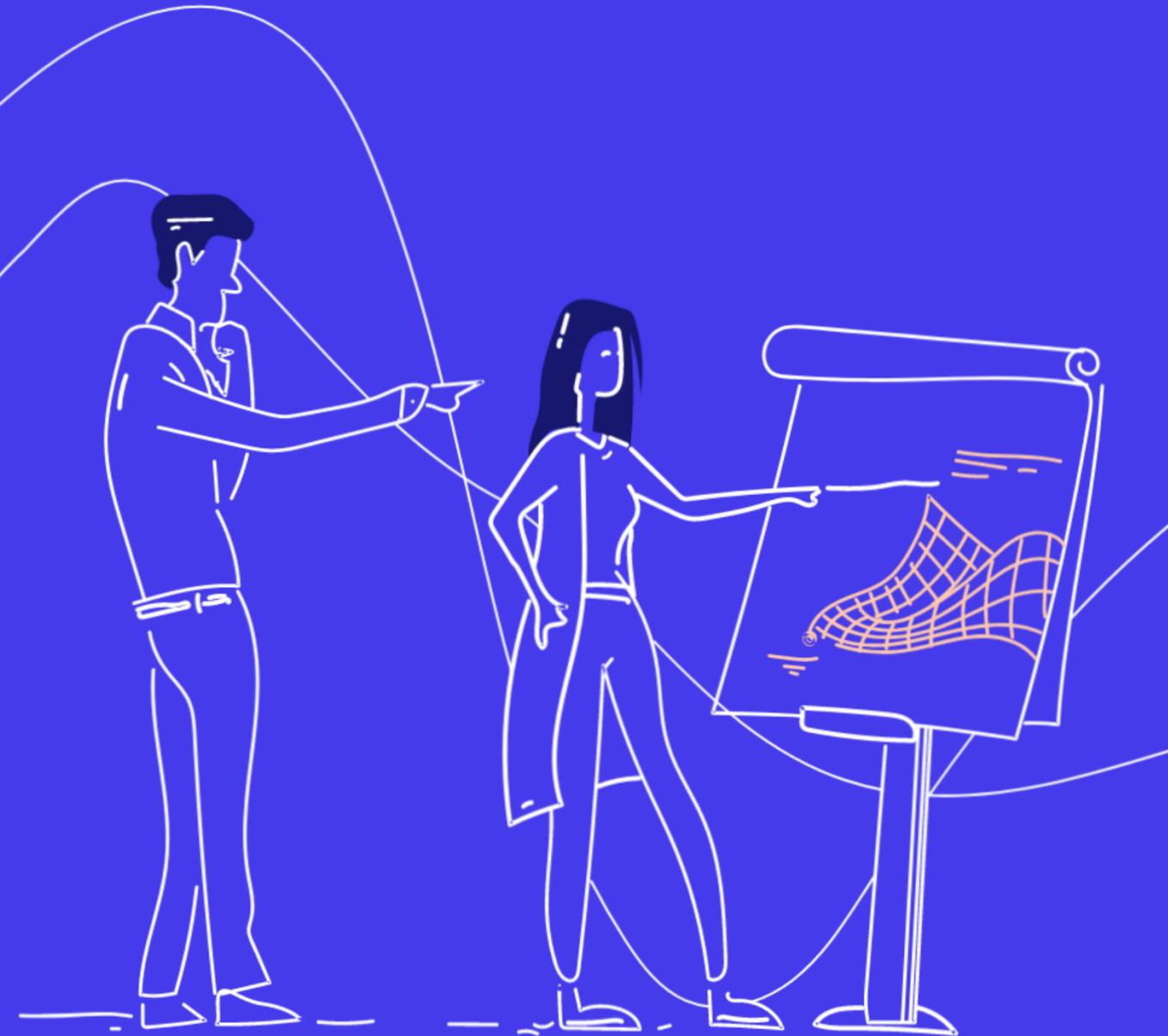




# Model-Based Design for DO-178C/DO-331 Compliance



SciEngineer's training courses are designed to help organizations and individuals close skills gaps, keep up-to-date with the industry-accepted best practices and achieve the greatest value from MathWorks® and COMSOL® Products.

# Model-Based Design for DO-178C/DO-331 Compliance

This three-day course is intended for software engineers that are using MathWorks tools to generate production code intended for DO-178C certification. The course assumes prior knowledge of Simulink modeling principles and verification workflows in Simulink and Polyspace. The course focuses on generating artifacts from those workflows that can be used in the DO certification process. Note that this training course does not cover the DO-178C or DO-331 standards themselves, but rather how to use MathWorks tools to demonstrate DO-178C and DO-331 compliance.

## Prerequisites

Simulink Fundamentals (or Simulink Fundamentals for Aerospace System Design), Simulink Model Management and Architecture, and Simulation-Based Testing with Simulink. Knowledge of the DO-178C standard and its DO-331 supplement for model-based software development. As needed: Stateflow for Logic-Driven System Modeling, Design Verification with Simulink, Embedded Coder for Production Code Generation, and Polyspace for C/C++ Code Verification.

DURATION	LEVEL
3 Days	Advanced
	

### TOPICS

## Day 1

- Model-Based Design for DO-178C/DO-331 Overview
- Requirements Traceability
- Modeling Standards for DO-178C/DO-331

## Day 2

- Model Verification
- Model Coverage
- Derived Requirements and Design Verification

## Day 3

- Model-Code Verification and Traceability
- Static Code Analysis and Formal Code Verification
- Code Verification Against Software Requirements
- Tool Qualification

## Model-Based Design for DO-178C/DO-331 Overview

OBJECTIVE: Understand how MathWorks workflows can satisfy and support DO-178C and DO-331 objectives.

- Overview of system, software, and hardware life cycles
- DO-178C/DO-331 workflow with Model-Based Design
- Satisfying DO-178C/DO-331 objectives using Model-Based Design

## Requirements Traceability

OBJECTIVE: Trace a model to requirements according to DO-178/DO-331 compliance.

- Requirements overview in the DO-178C process
- Using Requirements Toolbox for requirement traceability
- Generating a System Design Description (SDD)
- Understanding compatibility with external requirement tools

## Modeling Standards for DO-178C/DO-331

OBJECTIVE: Understand conformance to modeling standards for DO-178C/DO-331.

- Modeling standards for DO-178C/DO-331 applications
- Modeling standards in the Model-Based Design (MBD) process
- Using Simulink Check pre-packaged rules for high-integrity systems

## Model Verification

OBJECTIVE: Learn to verify software requirements through simulation.

- Test cases authoring in Simulink Test
- Test case inputs and data collection options
- Tracing test cases to requirements
- Requirements-based testing with Simulink Test

## Model Coverage

OBJECTIVE: Utilize Simulink Coverage to collect model coverage data.

- Model coverage metrics
- Capturing model coverage with Simulink Coverage
- Missing model coverage resolution

## Derived Requirements and Design Verification

OBJECTIVE: Learn how to utilize Simulink Design Verifier for low-level requirement verification.

- Simulink Design Verifier (SLDV) to identify unreachable elements
- Developing low-level tests for missing Model coverage
- Model coverage reporting

## Model-Code Verification and Traceability

**OBJECTIVE:** Use Simulink Code Inspector to demonstrate structural and algorithmic equivalence between a model and its generated code.

- Code inspection overview
- Checking model compatibility for Simulink Code Inspector
- Verifying model-to-code traceability

## Static Code Analysis and Formal Code Verification

**OBJECTIVE:** Show that generated code is compliant with coding standards and prove that it is robust against various run-time errors.

- Polyspace tools overview
- Running Polyspace analyses from Simulink
- Enforcing coding standards with Polyspace Bug Finder
- Proving code robustness with Polyspace Code Prover
- Demonstrating control and data coupling coverage with Polyspace Code Prover

## Code Verification Against Software Requirements

**OBJECTIVE:** Use Simulink Test to demonstrate complete code coverage via software requirements test cases, and verify software requirements on the target hardware.

- Software-in-the-loop (SIL) and processor-in-the-loop (PIL) overview
- Testing executable object code against high-level requirements
- Collecting code coverage

# Tool Qualification

OBJECTIVE: Use DO Qualification Kit to qualify MathWorks products and features for DO-178C certification.

- Tool qualification requirements
- DO Qualification Kit overview
- Tool qualification example
- Qualifiable tools and credits taken
- DO Qualification Kit customization



**Expand your  
knowledge**

