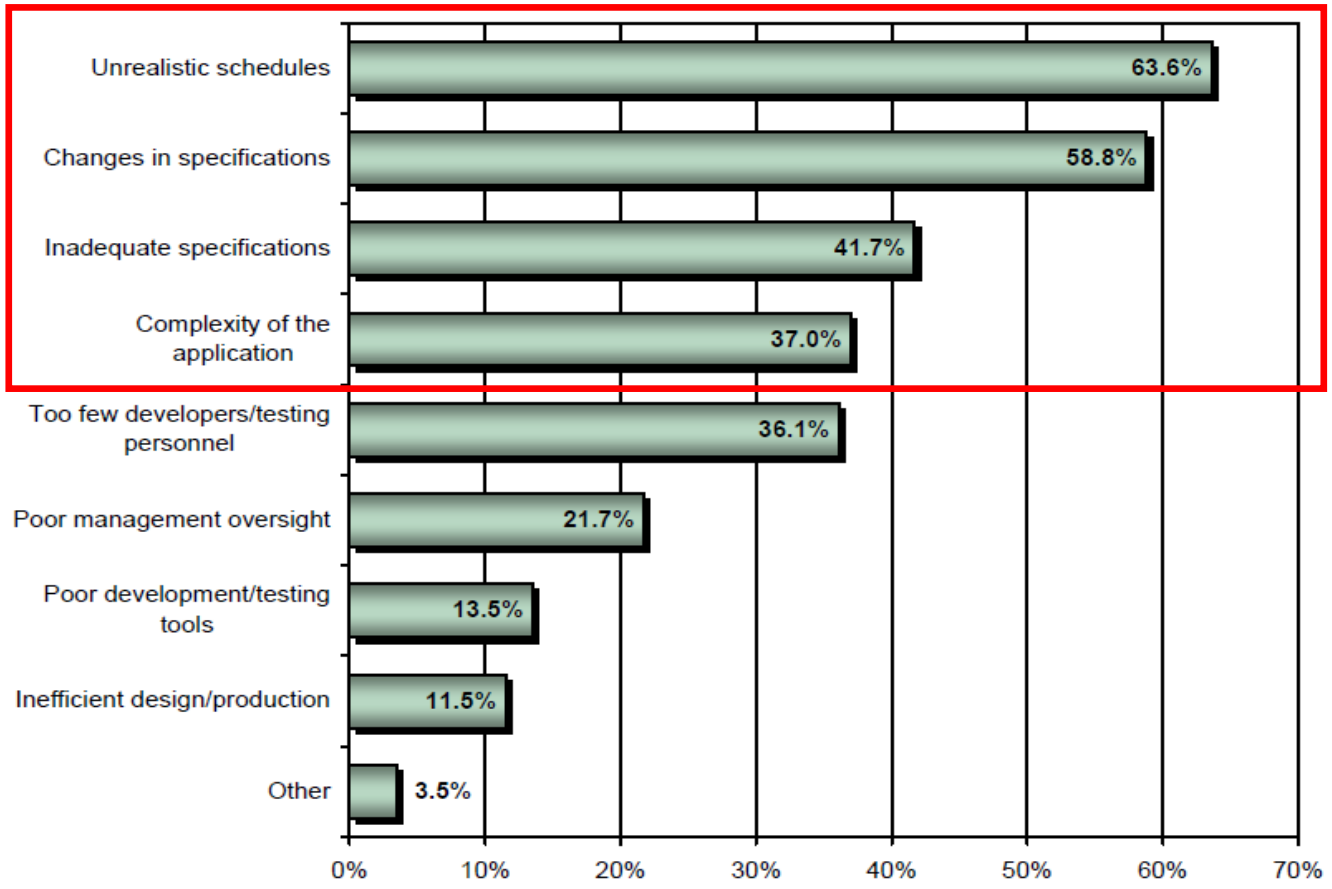
Model-Based Design for High-Integrity Software Development

Flight Software Workshop 2016

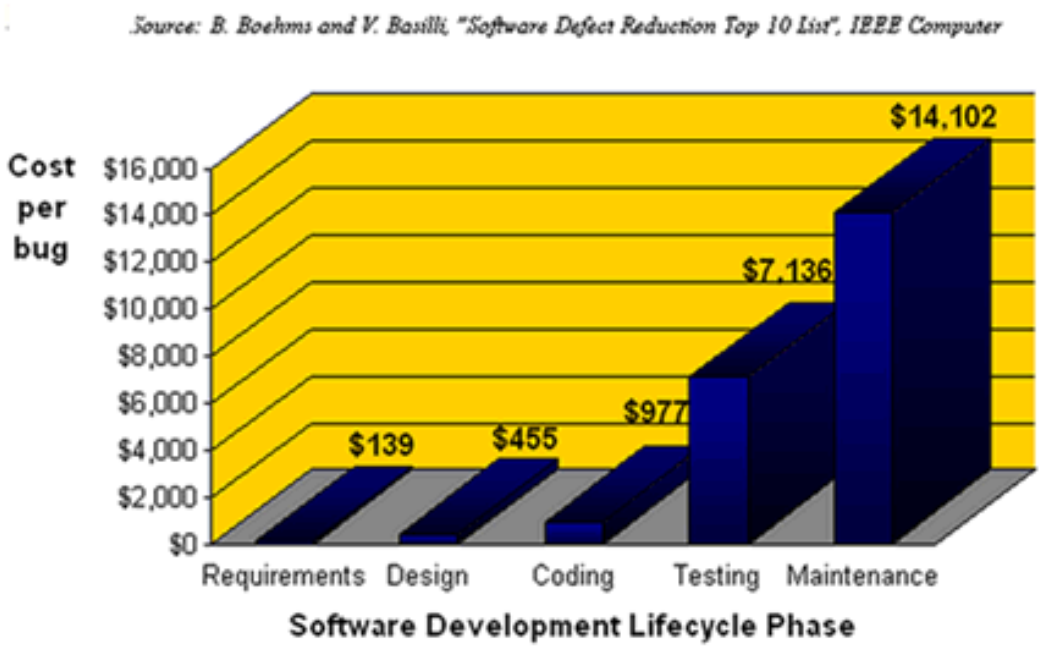
Matt Rhodes

Why did we miss our deadline?



Reasons for late projects, as reported by Venture Development Corporation.
Source: Embedded Software Strategic Market Intelligence report, Volume 4, December 2007, VDC.
Note: Percentages sum to over 100% due to multiple responses.

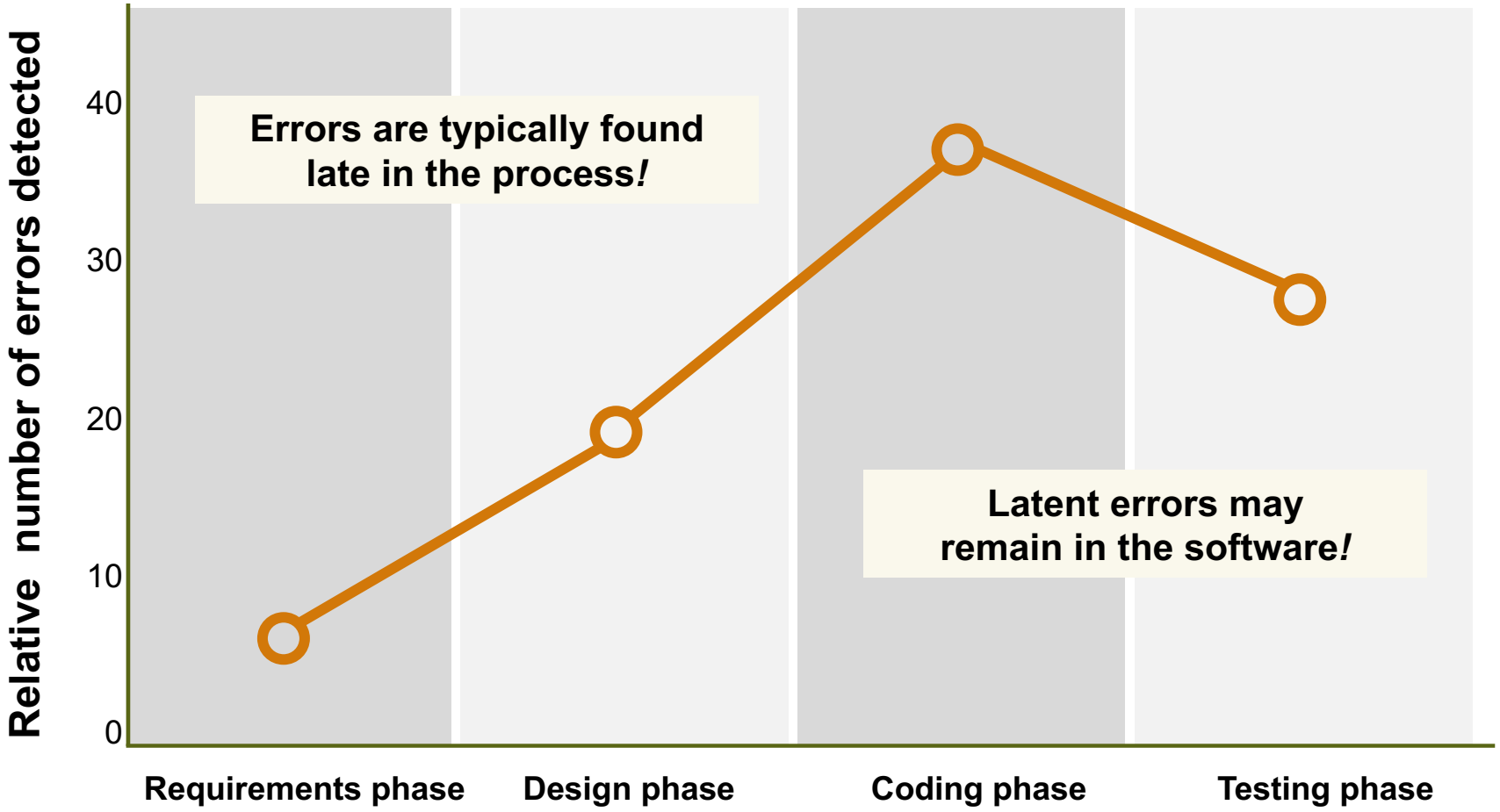
Minimize Costs by Detecting Errors Earlier



"…each delay in the detection and correction of a design problem makes it <u>an order of magnitude more expensive</u> to fix…"

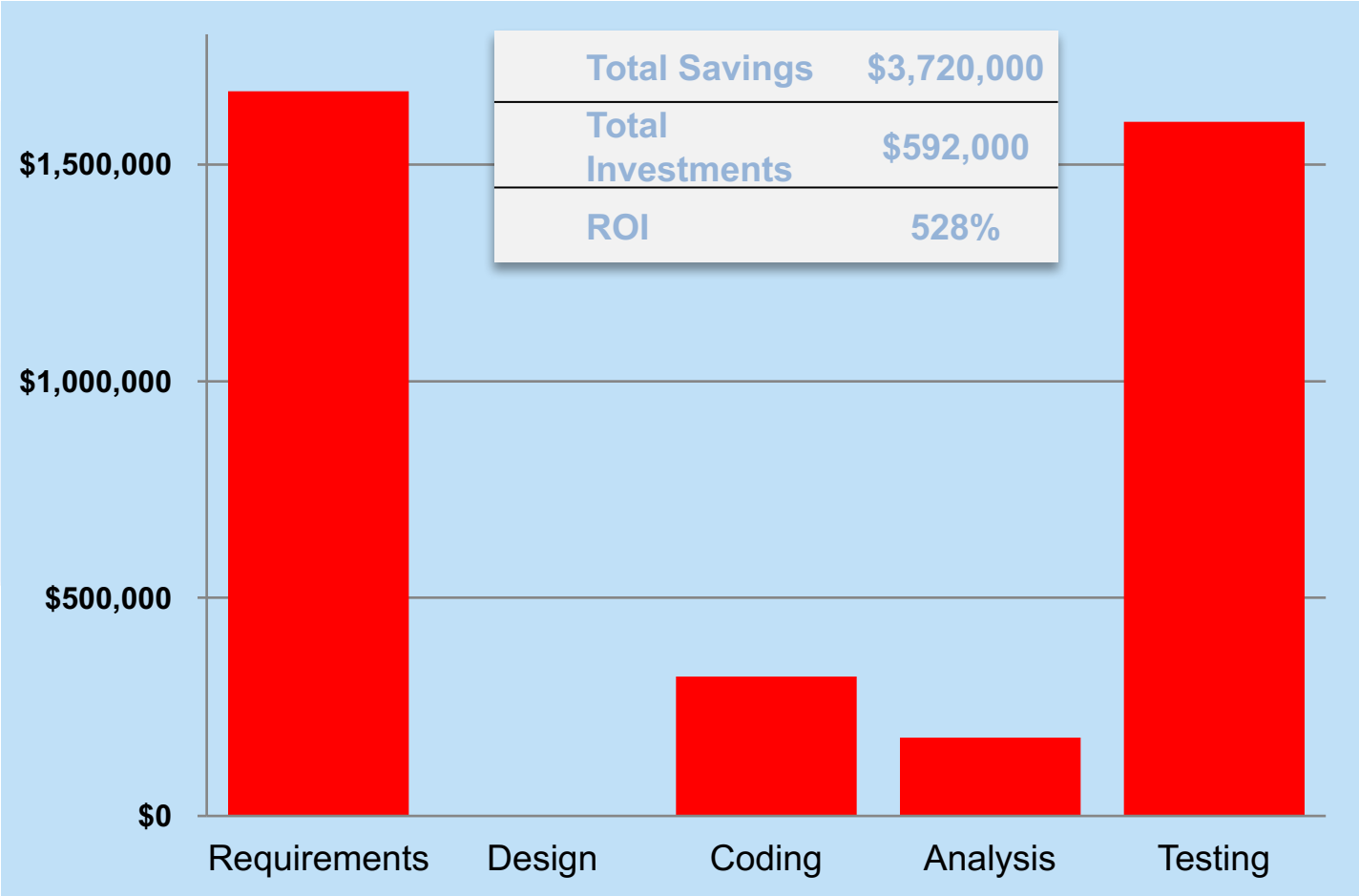
Clive Maxfield and Kuhoo Goyal
"EDA: Where Electronics Begins"
TechBites Interactive, October 1, 2001
ISBN: 0971406308]

Phase where errors are *found*

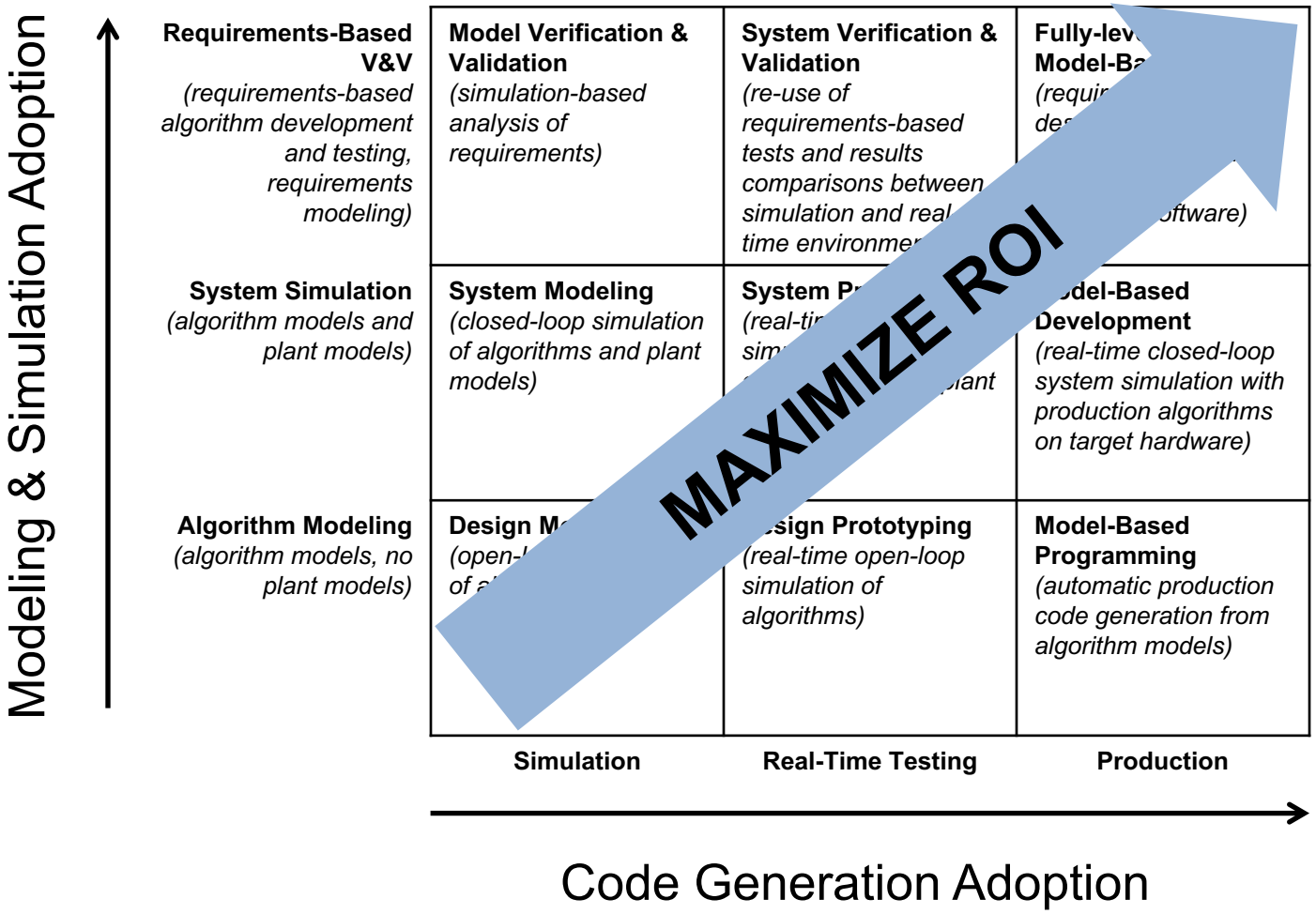


Source: How good is the software: A review of defect prevention techniques; Brad Clark, David Zubrow, Carnegie Mellon Software Engineering Institute; Software Engineering Symposium 2001

62% Cost Savings

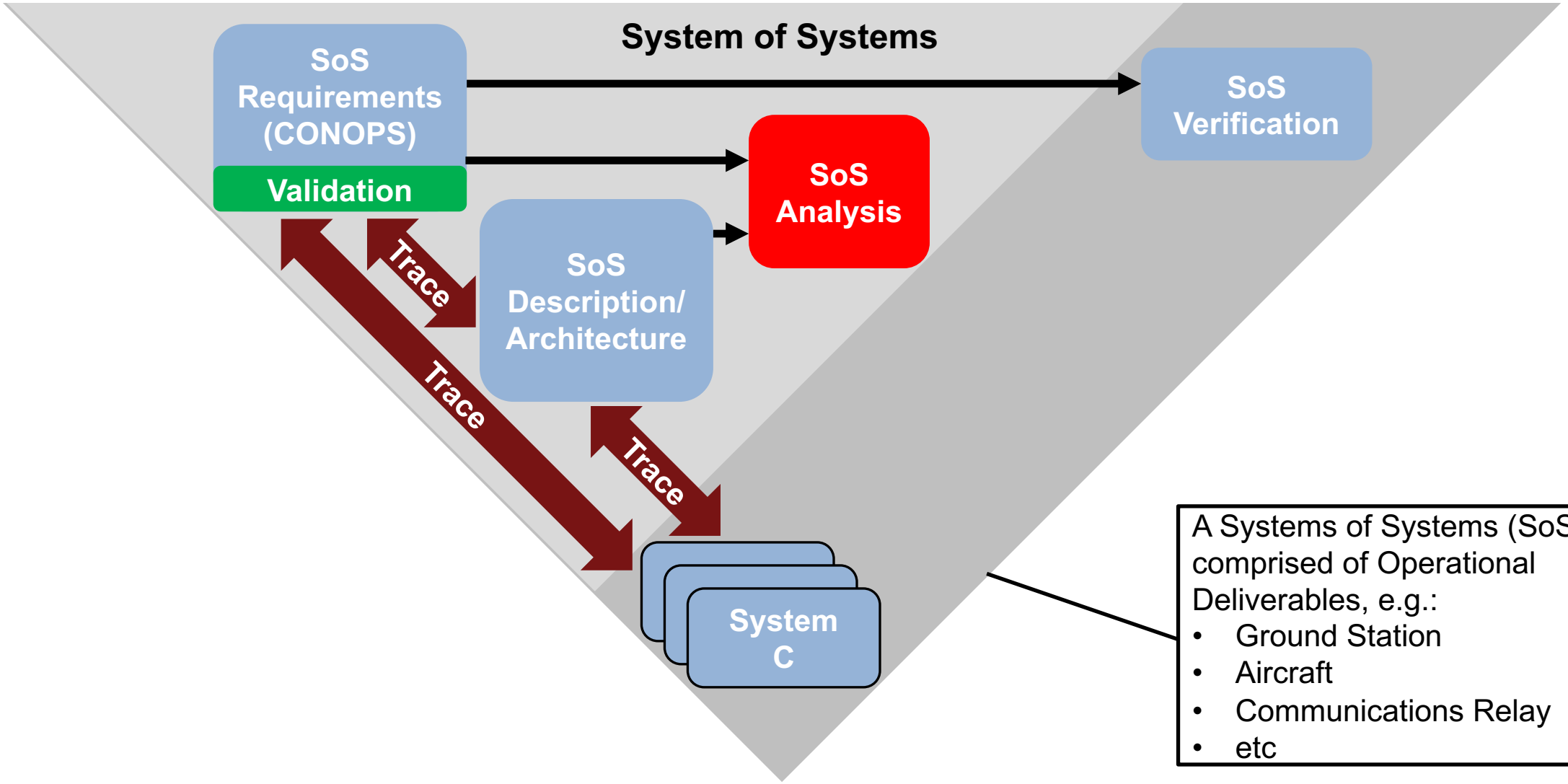


Model-Based Design Maturity



The Design V: Let’s Start from the Very Top

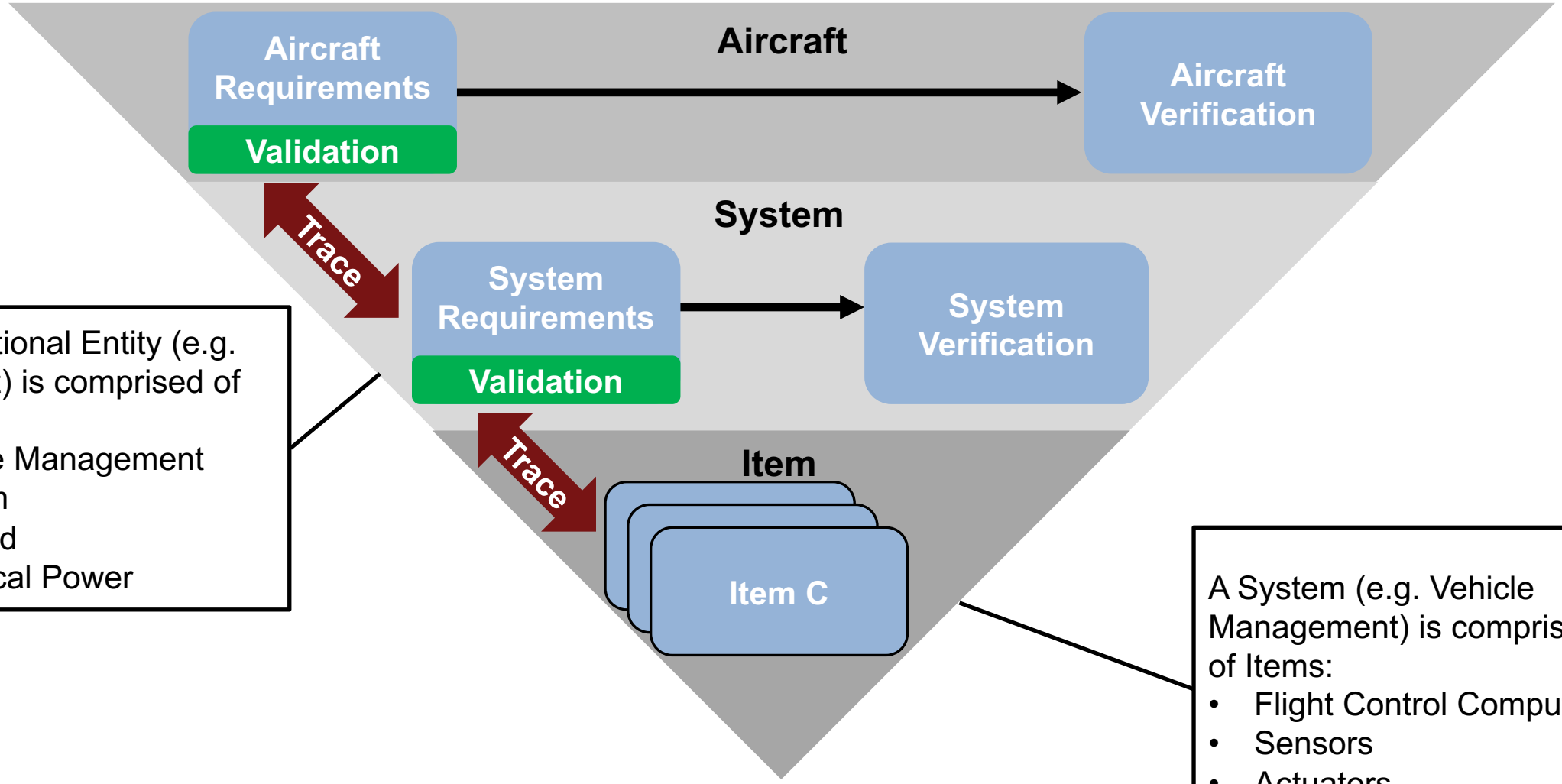
Systems Engineering Workflow



A Systems of Systems (SoS) is comprised of Operational Deliverables, e.g.:

- Ground Station
- Aircraft
- Communications Relay
- etc

Aircraft/Subsystems Workflow (ARP4754A)



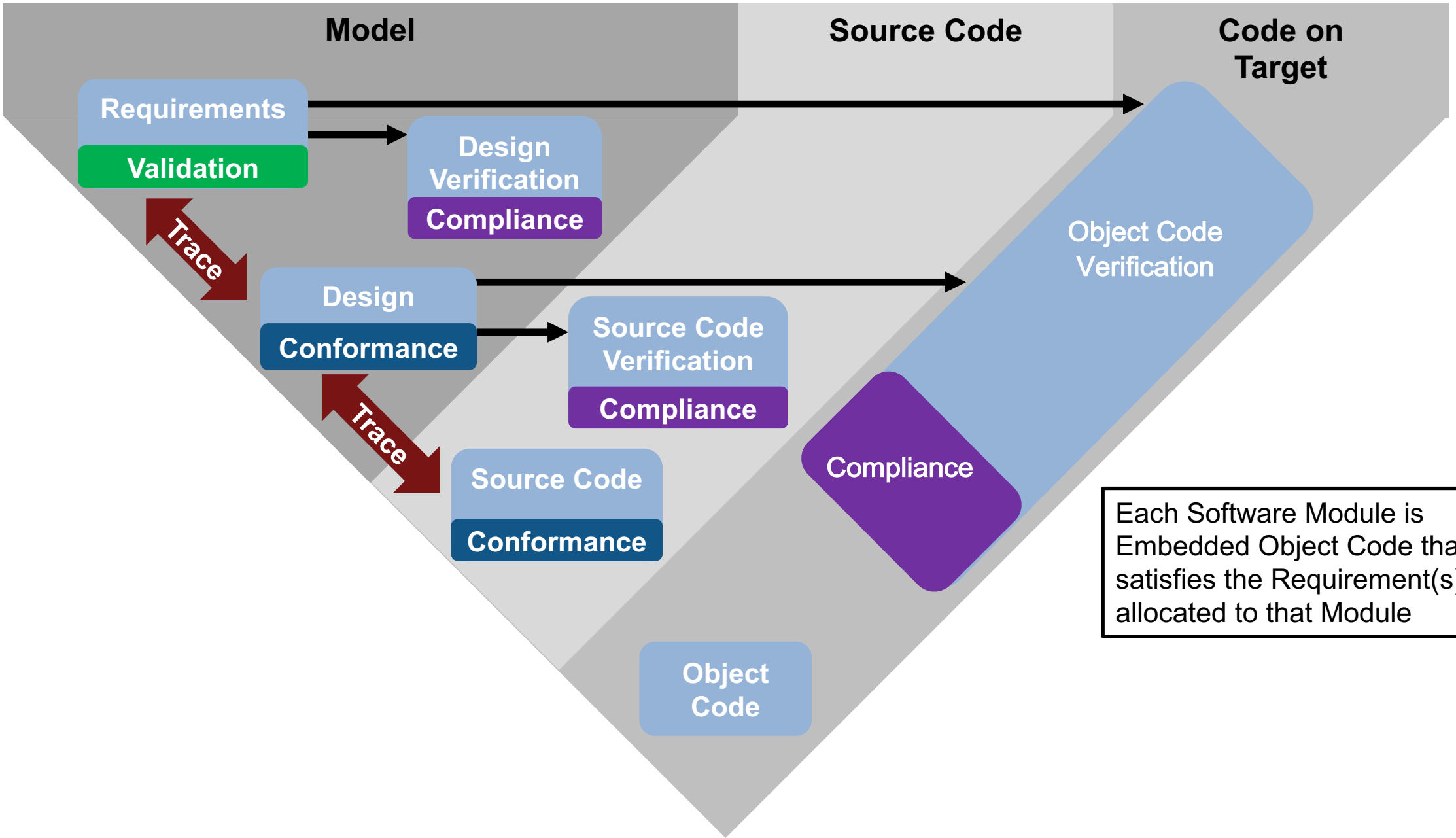
An Operational Entity (e.g. an Aircraft) is comprised of Systems:

- Vehicle Management System
- Payload
- Electrical Power

A System (e.g. Vehicle Management) is comprised of Items:

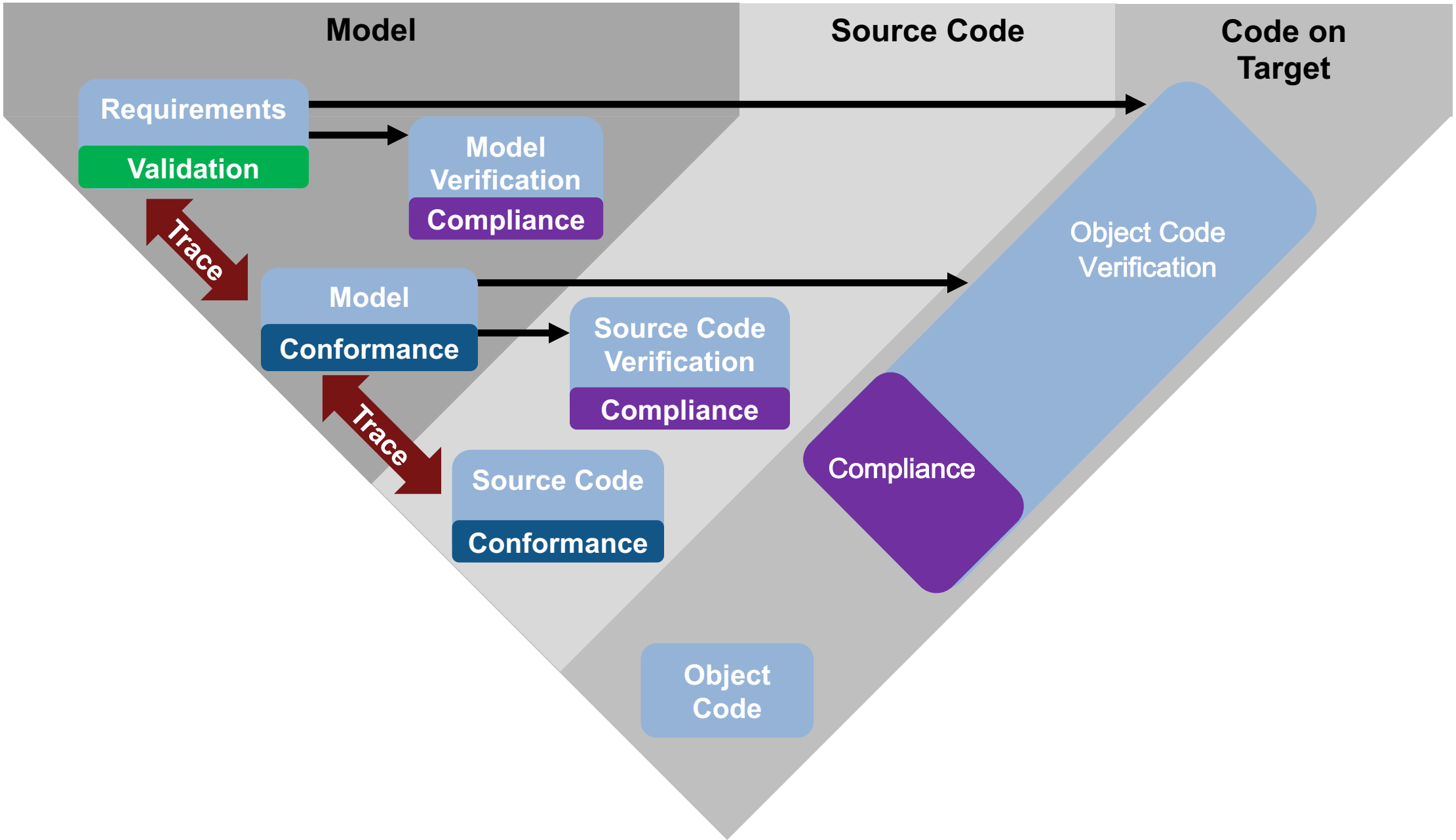
- Flight Control Computer
- Sensors
- Actuators

Software Module Development Workflow (DO-178C)

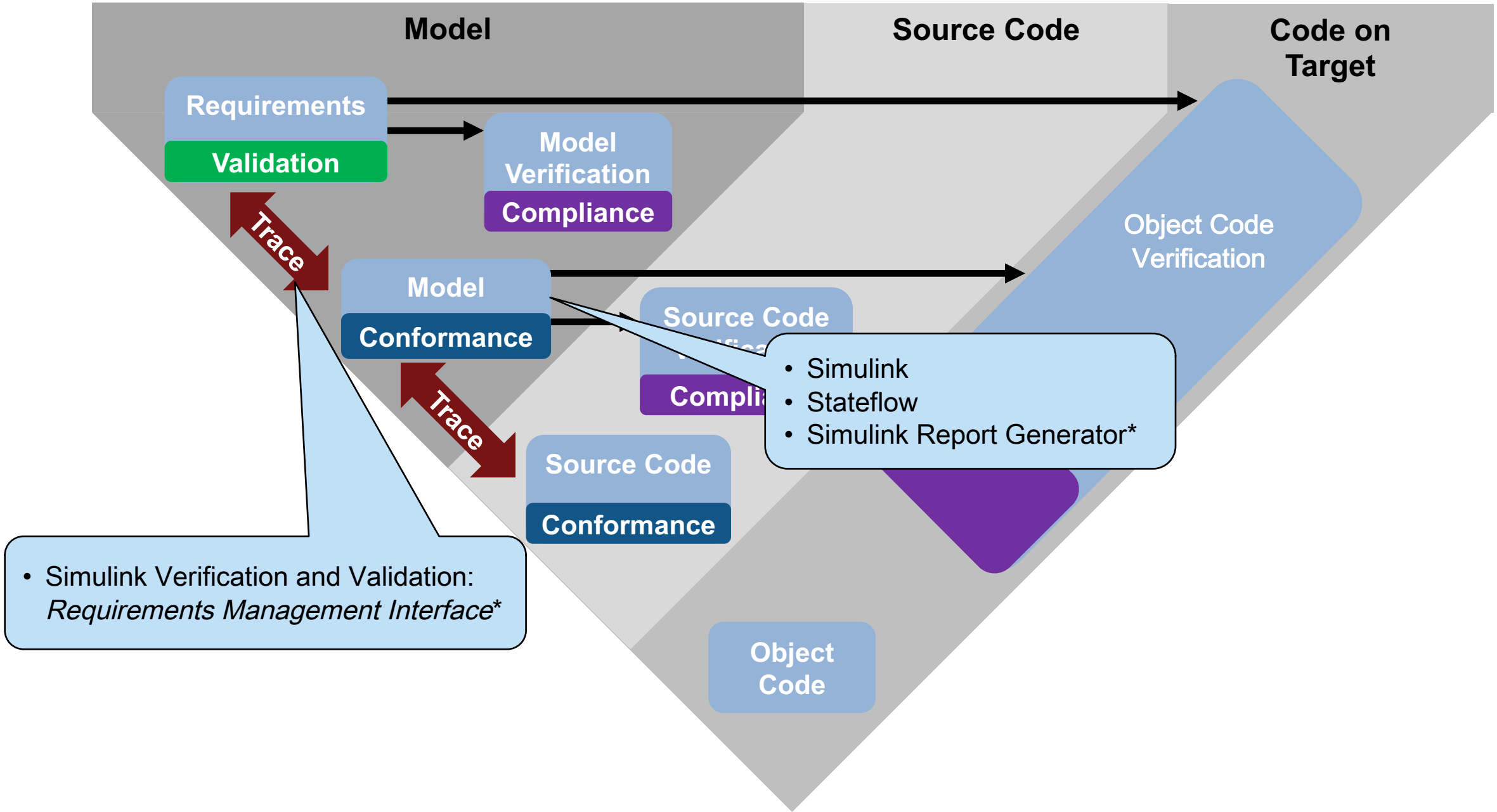


Each Software Module is Embedded Object Code that satisfies the Requirement(s) allocated to that Module

Model-Based Software Development Workflow (DO-178C)

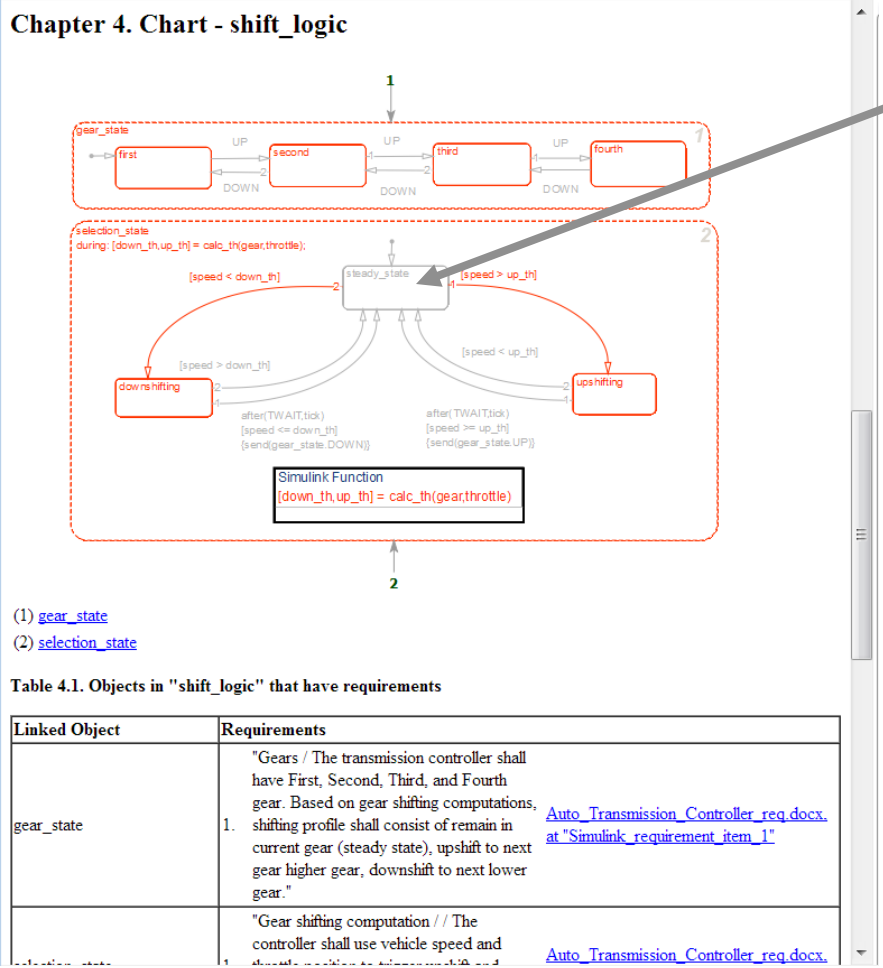


Model-Based Software Development Workflow (DO-178C)



Confirm requirements map to design

Produce requirements trace reports



Chapter 4. Chart - shift_logic

(1) gear_state
(2) selection_state

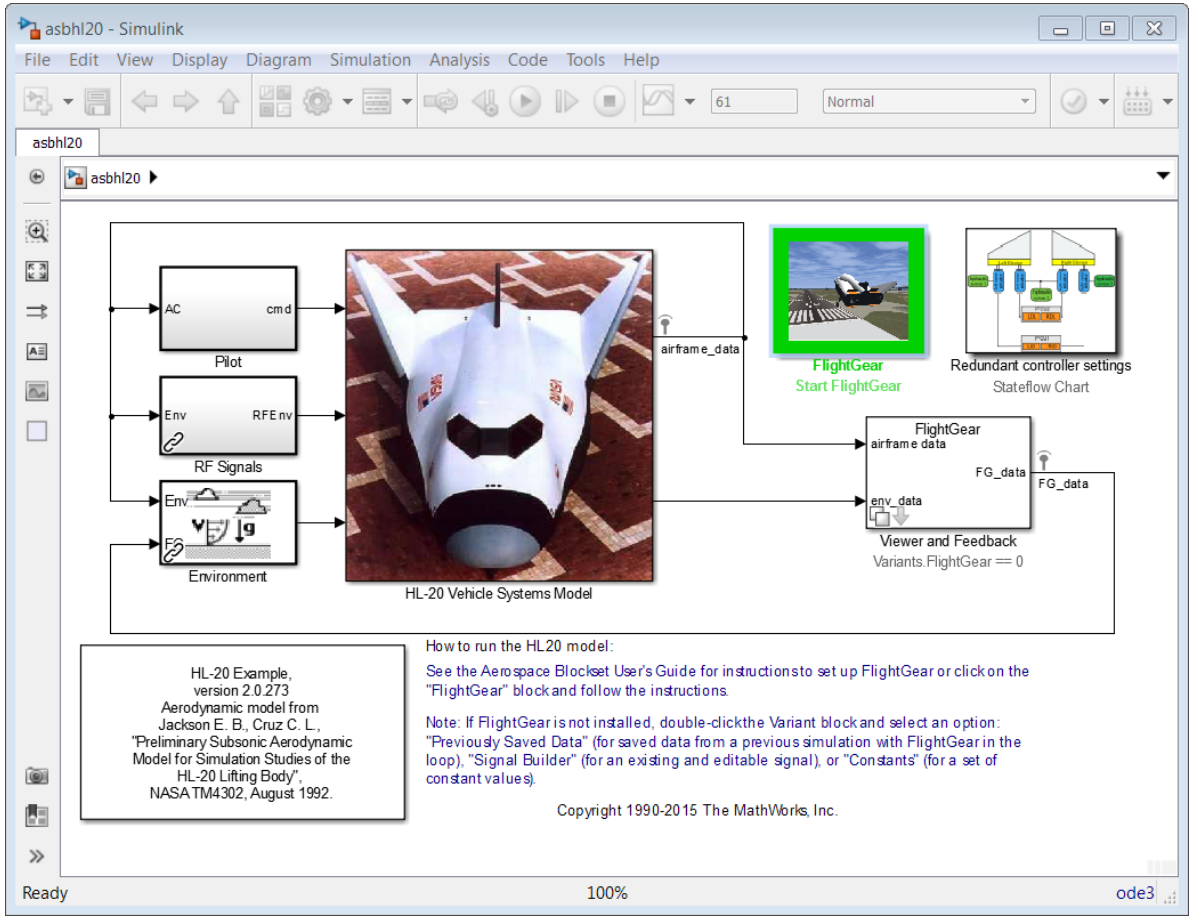
Table 4.1. Objects in "shift_logic" that have requirements

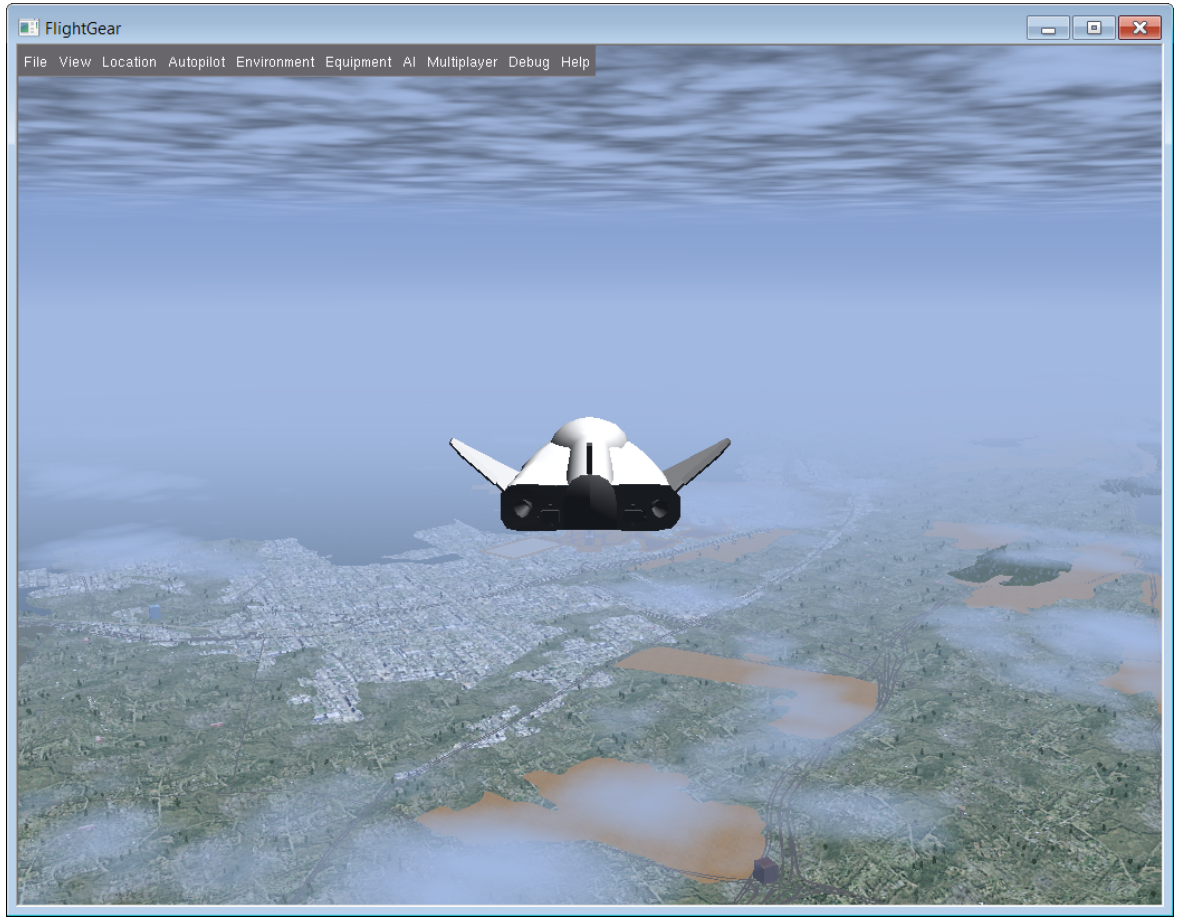
Linked Object	Requirements	
gear_state	1. "Gears / The transmission controller shall have First, Second, Third, and Fourth gear. Based on gear shifting computations, shifting profile shall consist of remain in current gear (steady state), upshift to next gear higher gear, downshift to next lower gear."	Auto_Transmission_Controller_req.docx, at "Simulink_requirement_item_1"
selection_state	1. "Gear shifting computation / / The controller shall use vehicle speed and	Auto_Transmission_Controller_req.docx,

- ***Identify missing requirements!***
- Design components that do not map to a requirement
- Indicative of incorrect design, or misinterpreting requirements
- Avoid finding unintended behaviors late in project phases

Tool: Simulink Verification and Validation

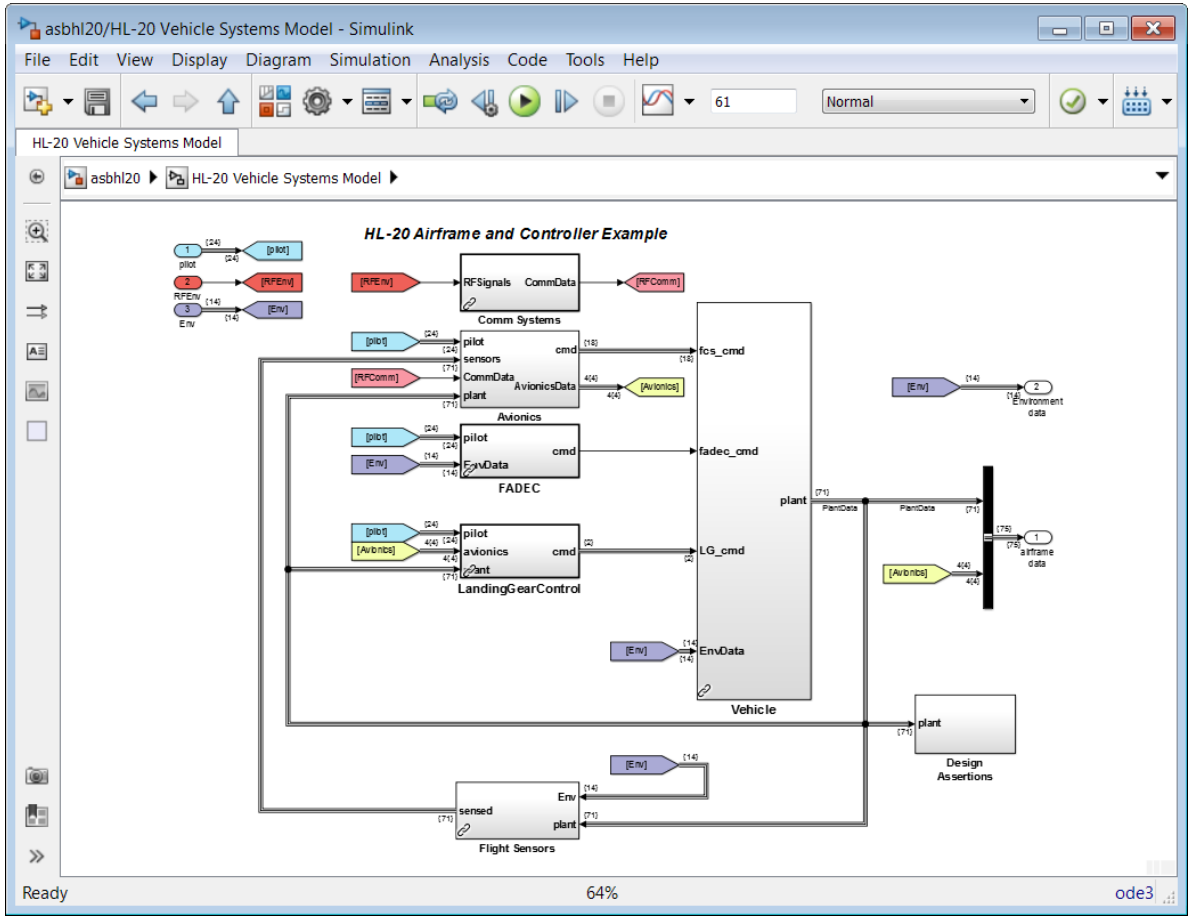
Modeling and simulation with Model-Based Design

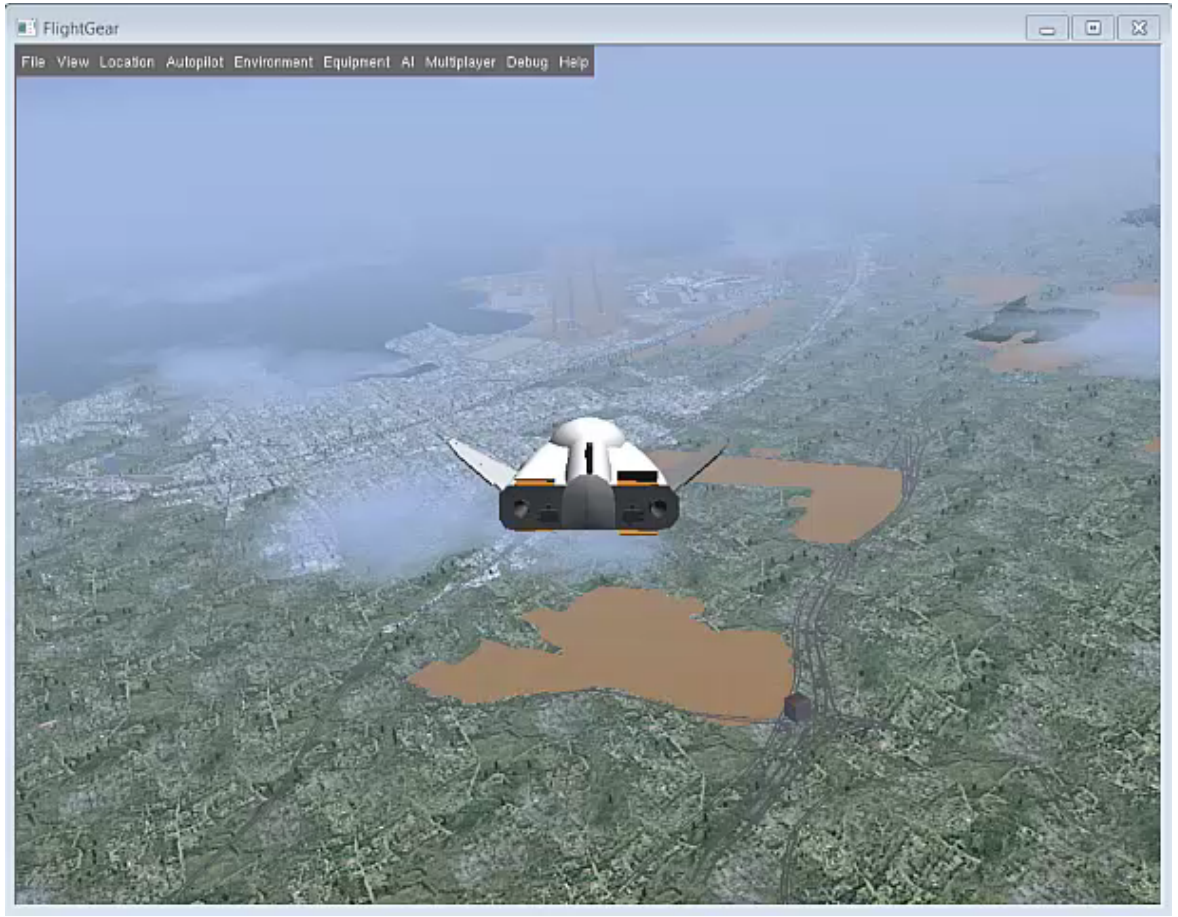




Reference: http://www.flightgear.org/

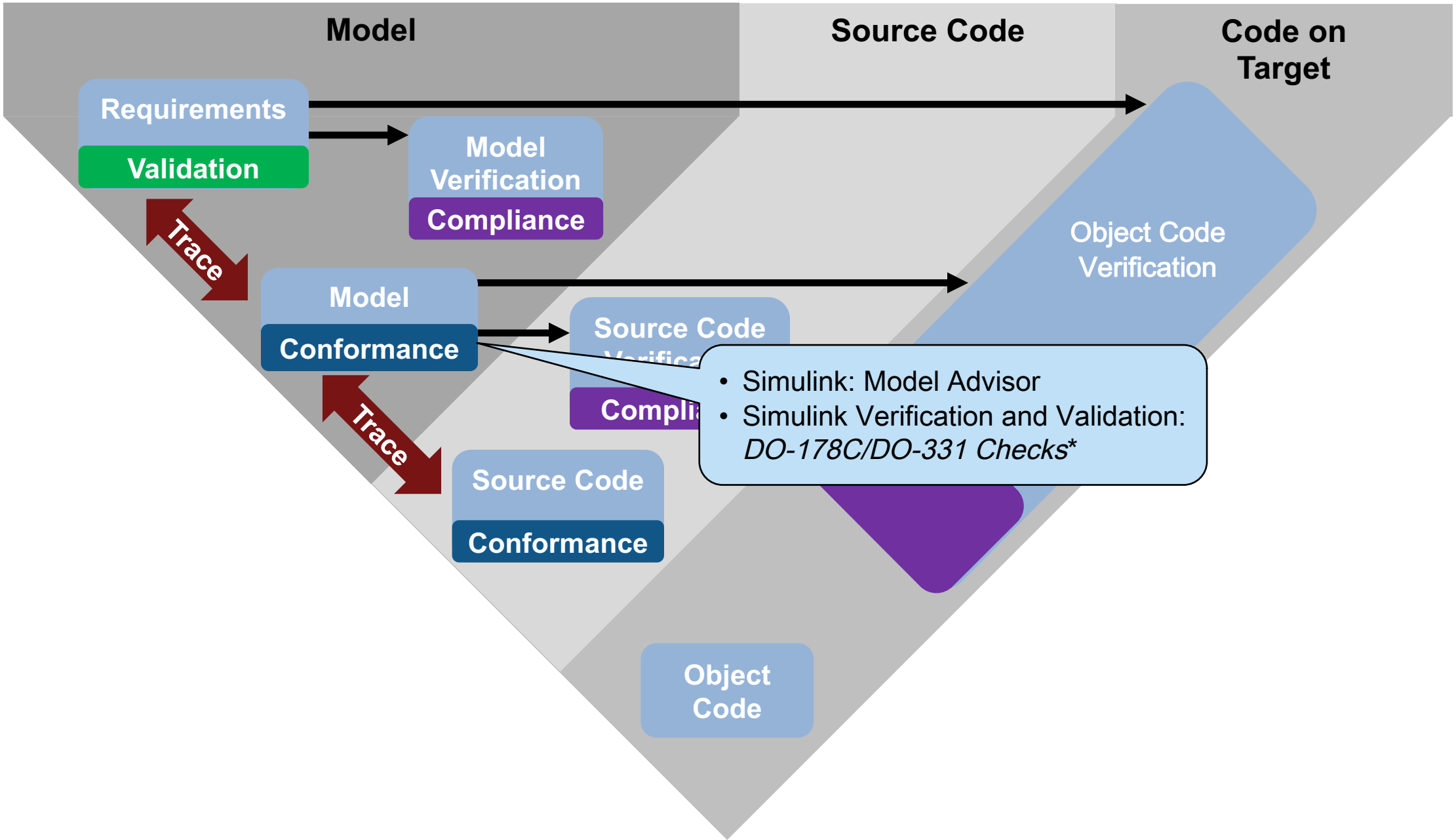
Modeling and simulation with Model-Based Design



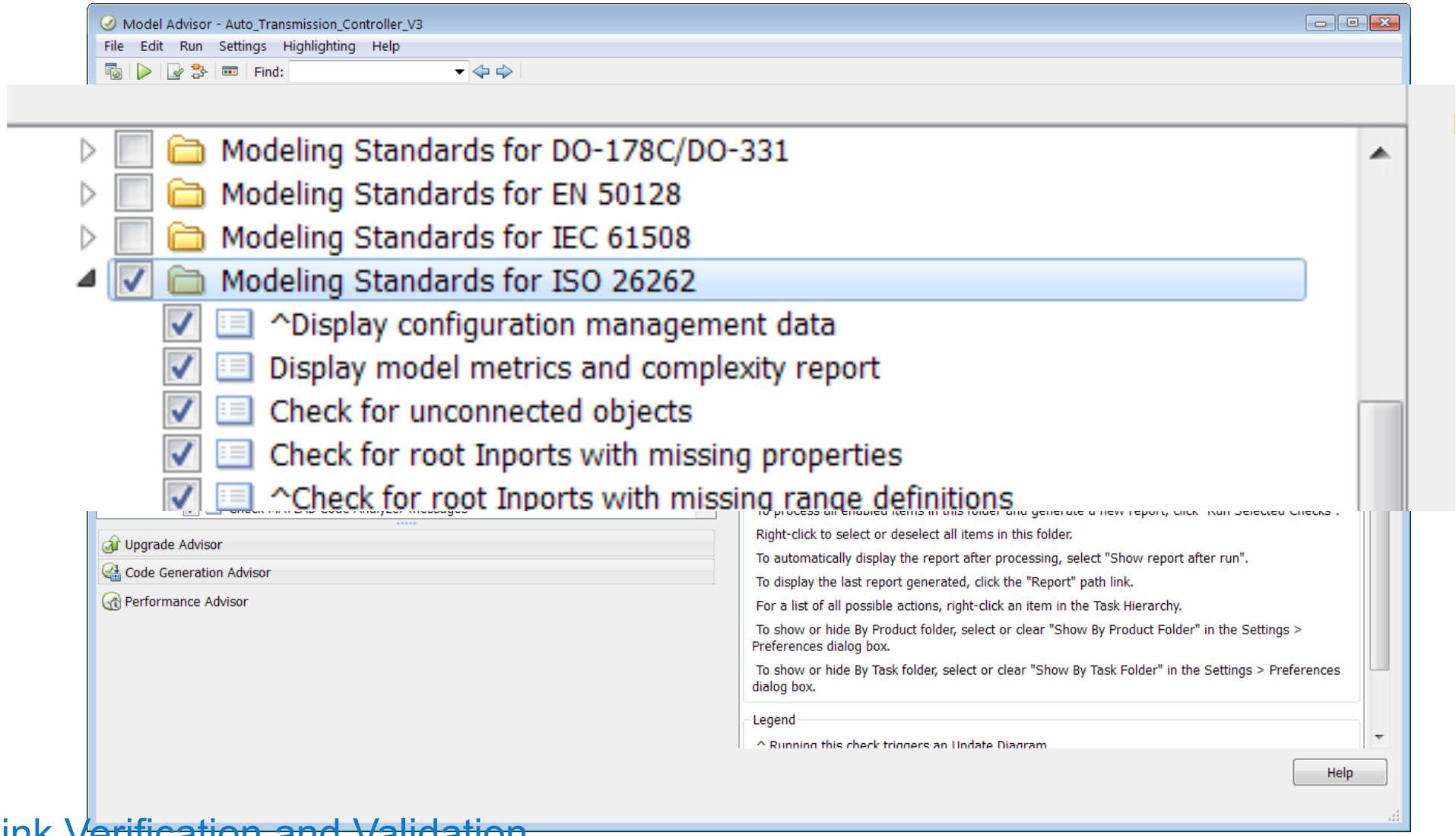


Reference: http://www.flightgear.org/

Model-Based Software Development Workflow (DO-178C)

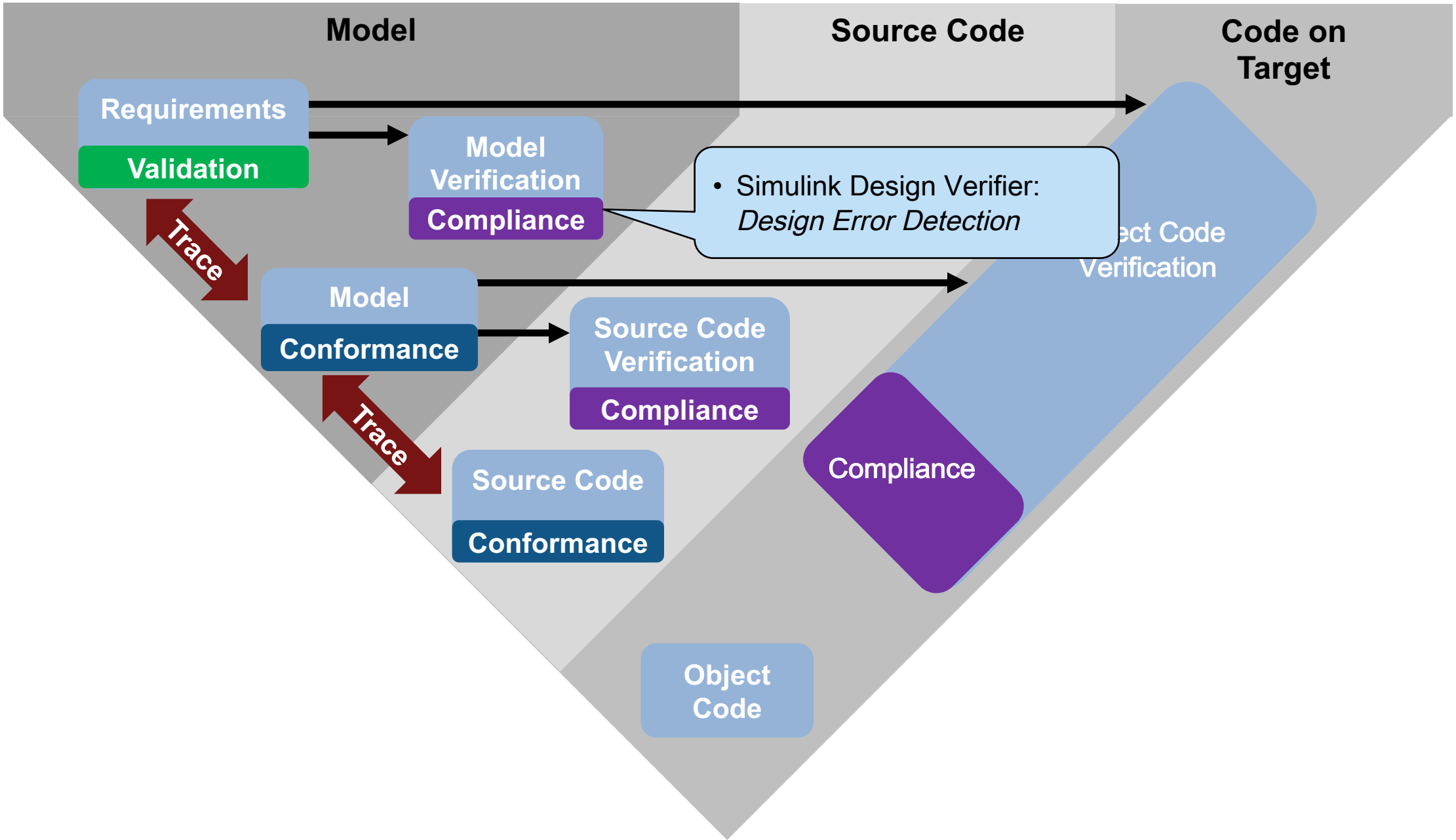


Check model complies with standards



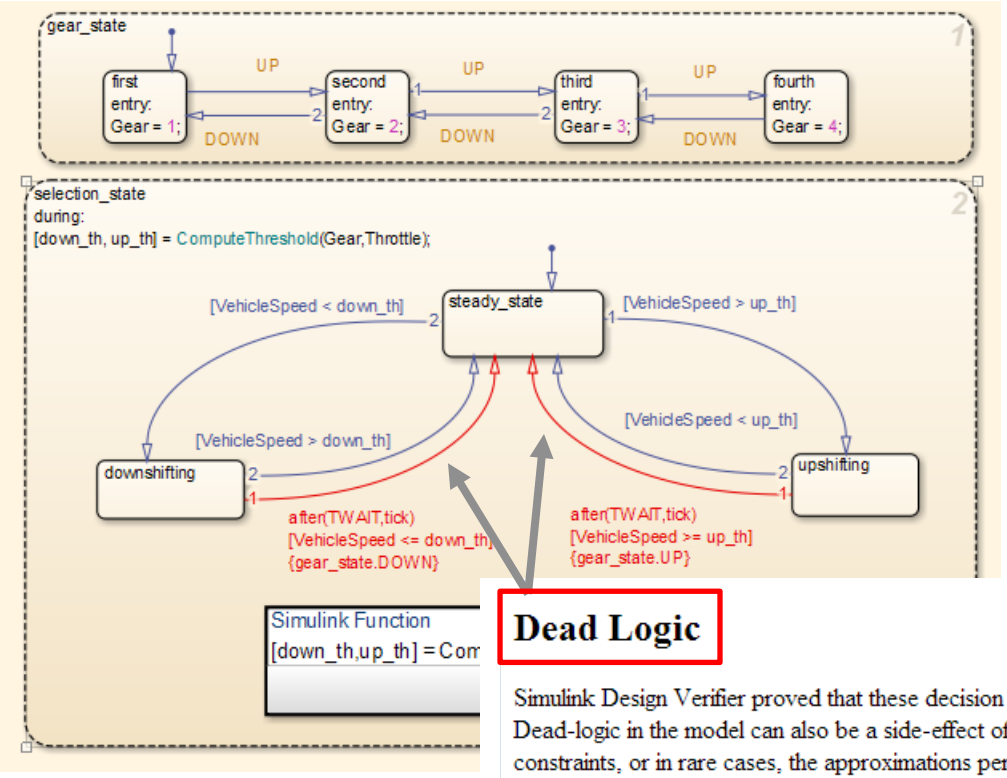
Tool: Simulink Verification and Validation

Model-Based Software Development Workflow (DO-178C)



Confirm design is free of implementation errors

Prove absence of dead logic, unreachable states



Dead Logic

Simulink Design Verifier proved that these decision and condition outcomes cannot occur and are dead-logic in the model. Dead-logic in the model can also be a side-effect of parameter configurations, input specified minimum maximum constraints, or in rare cases, the approximations performed by Simulink Design Verifier.

#	Type	Model Item	Description	Analysis Time (sec)
9	Condition	ShiftLogic.selection_state."after(TWAIT,tick) [VehicleS..."	Transition: Condition 1, "tick" F	34
19	Condition	ShiftLogic.selection_state."after(TWAIT,tick) [VehicleS..."	Transition: Condition 1, "tick" F	34

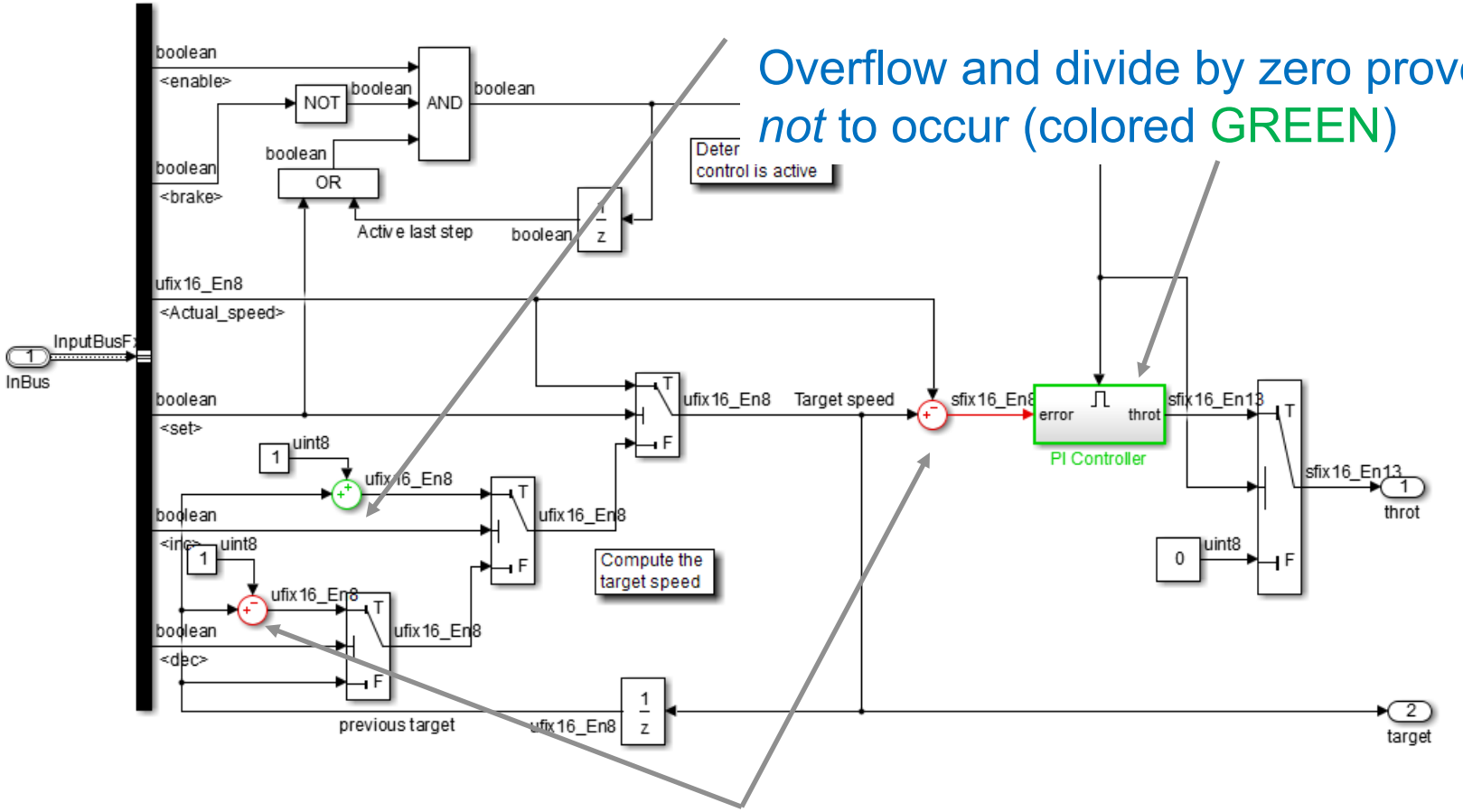
Tool: Simulink Design Verifier

Confirm design is free of implementation errors

Prove absence of overflow, divide by zero, and other design errors

Overflow proven *not* to occur (colored GREEN)

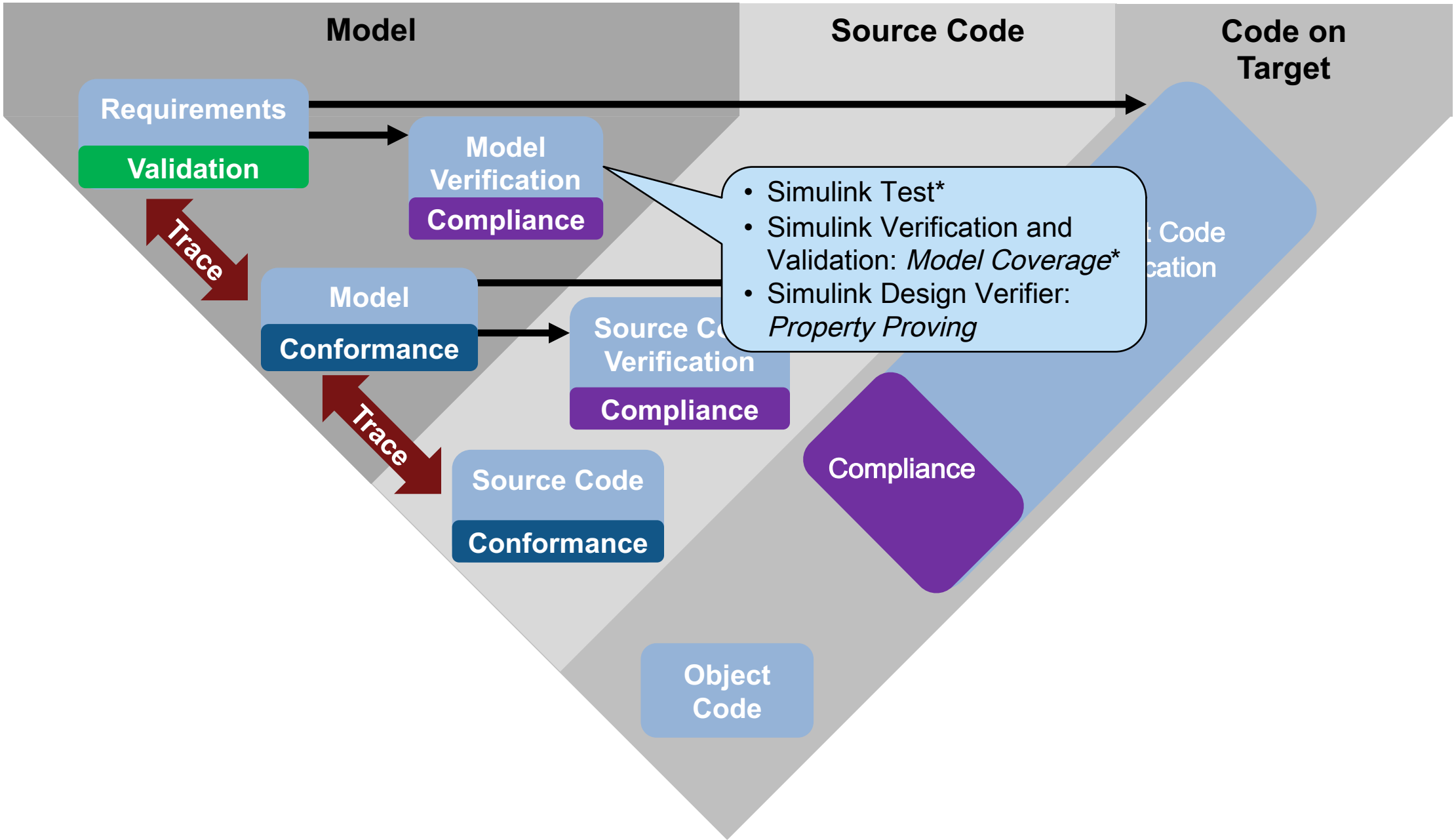
Overflow and divide by zero proven *not* to occur (colored GREEN)



Overflow proven to occur (colored RED)

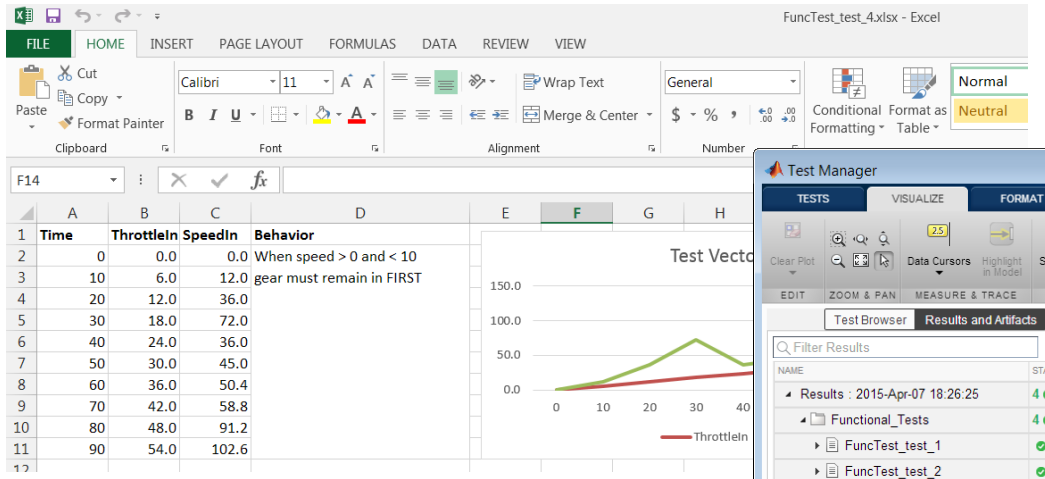
Tool: Simulink Design Verifier

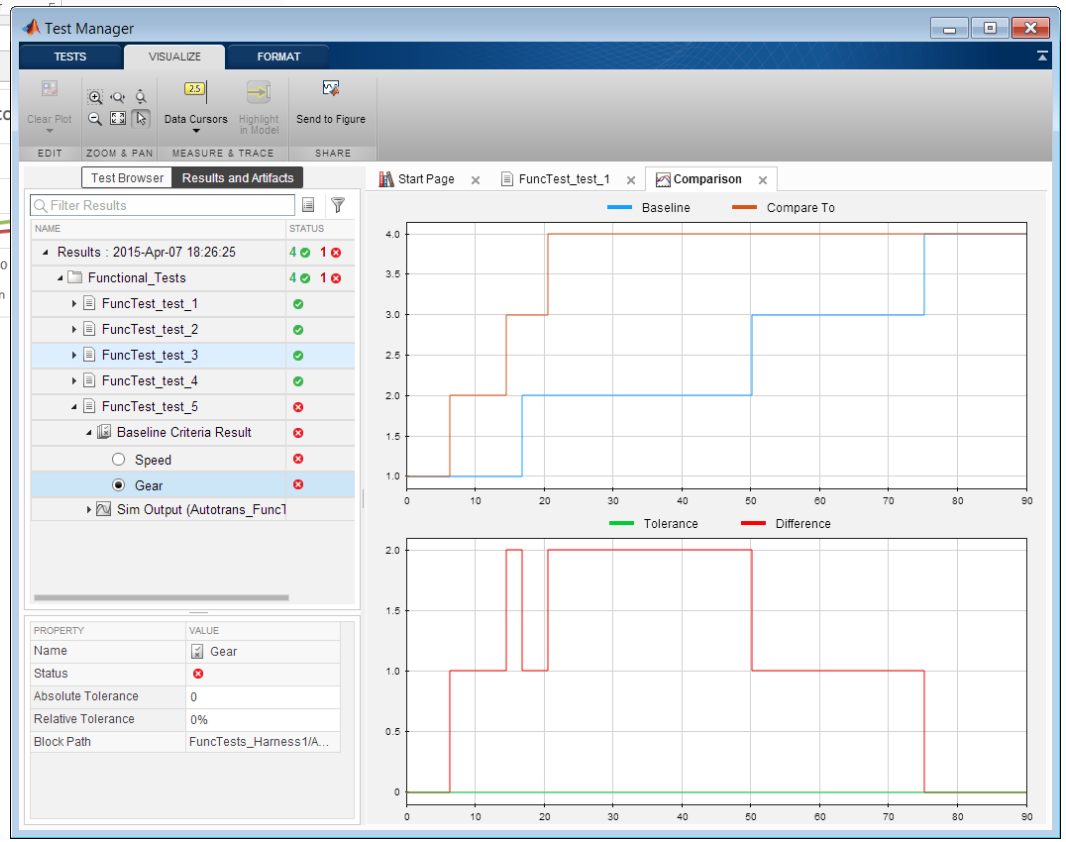
Model-Based Software Development Workflow (DO-178C)



Execute functional tests in simulation

Time series test vectors





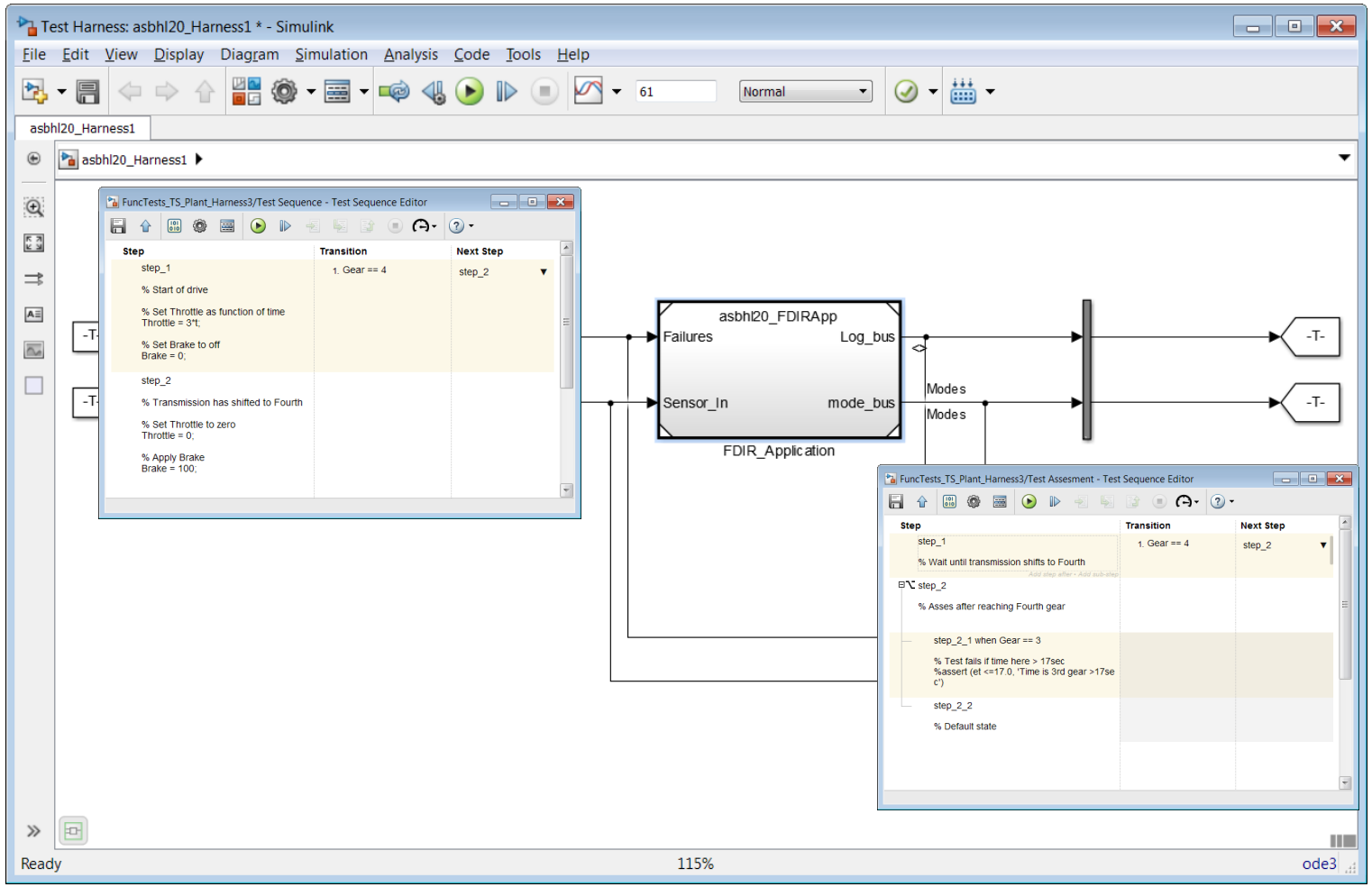
Pass / fail results

Tool: Simulink Test

Execute functional tests in simulation

Sequence based tests and assessments

Requirement → In the event of an failure condition of the position sensor or low hydraulic pressure, the Fault Detection Isolation and Recovery (FDIR) application shall select the primary actuator to isolated mode

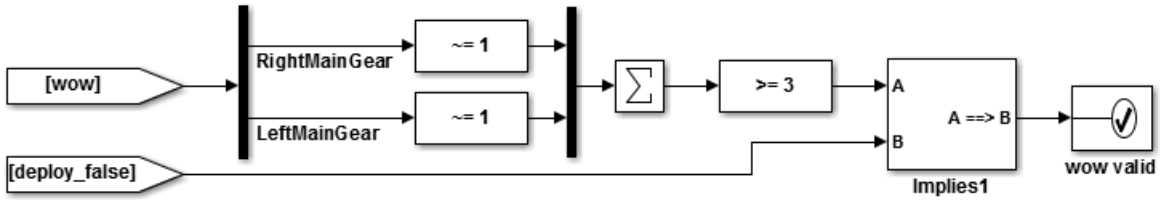


Tool: Simulink Test

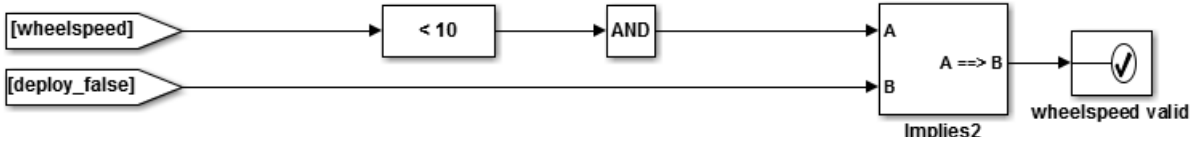
Complement testing with formal methods

Graphical representation of safety requirements for thrust reverser

If average airspeed > 150knots, deploy cannot be true



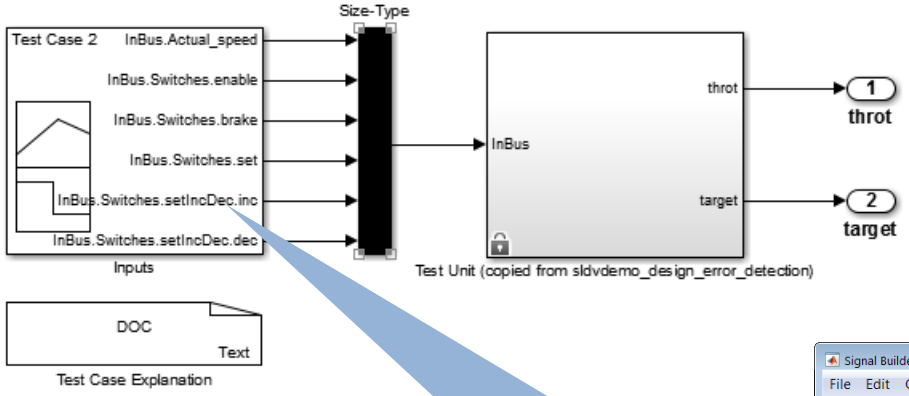
If two WOW sensors are false, deploy cannot be true

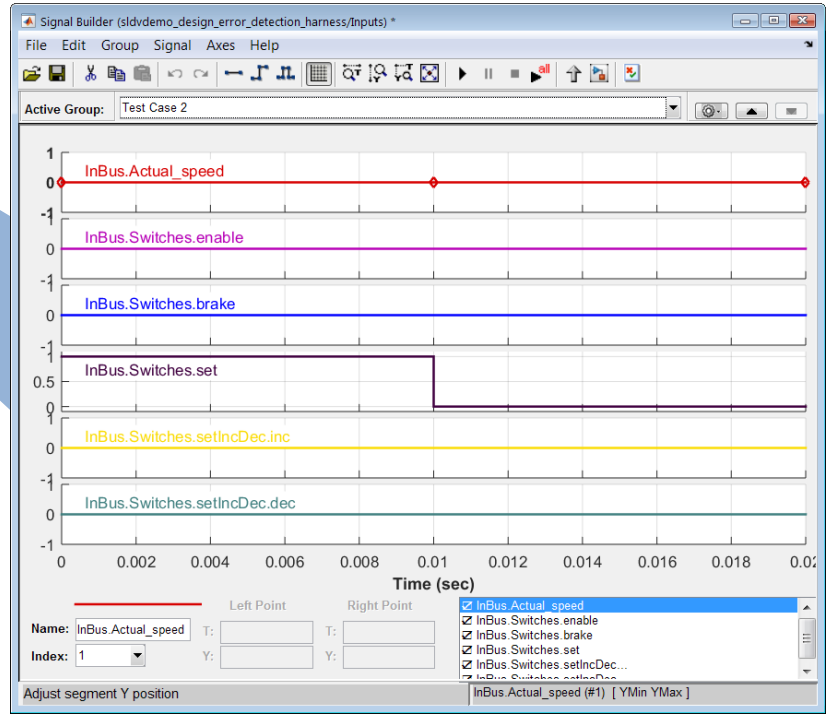


If either wheelspeed sensor < 10 knots, deploy cannot be true

Tool: Simulink Design Verifier

Counter example test generation

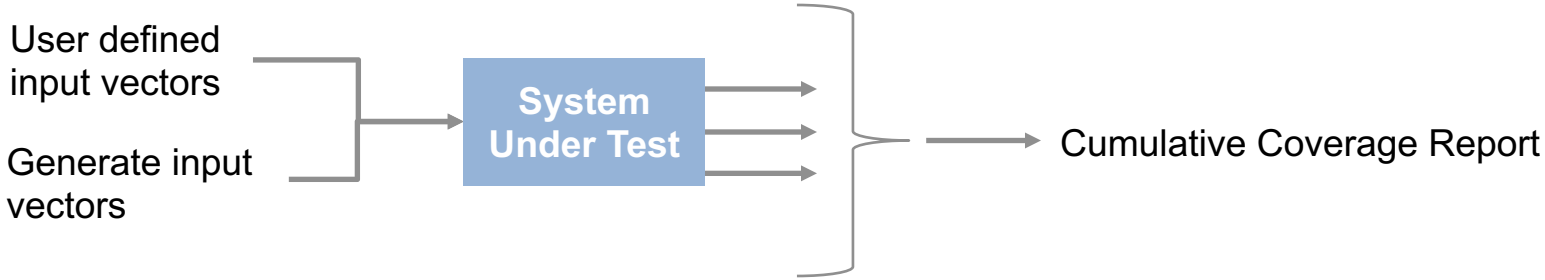




Tool: Simulink Design Verifier

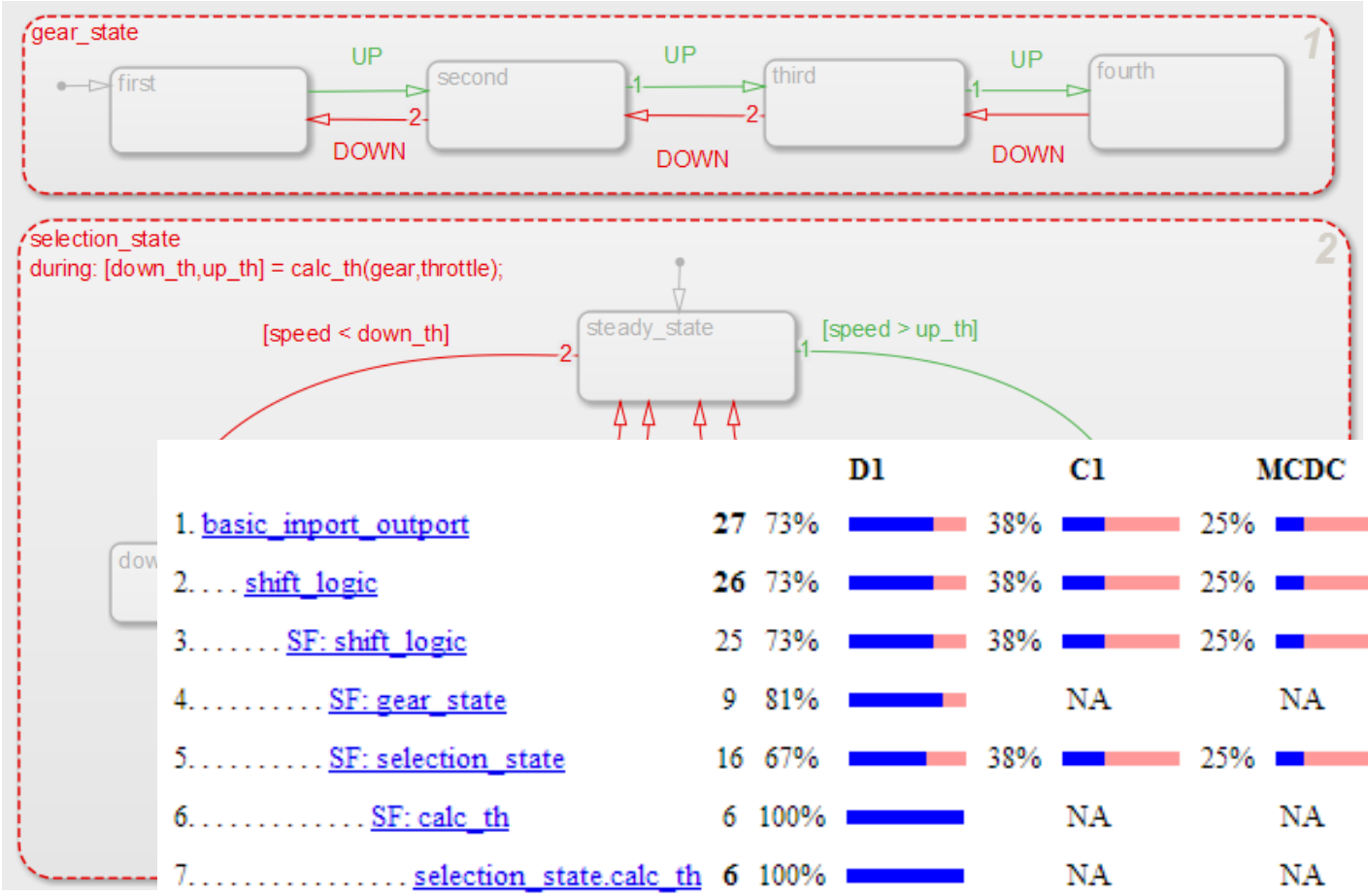
Generate additional tests with formal methods

- Top off functional tests with generated tests
  - Use model checking to explore state space
  - Generate missing test-cases to execute all paths
  - Tests generated for decision, condition, MC/DC, table



Tool: Simulink Design Verifier, Simulink Test

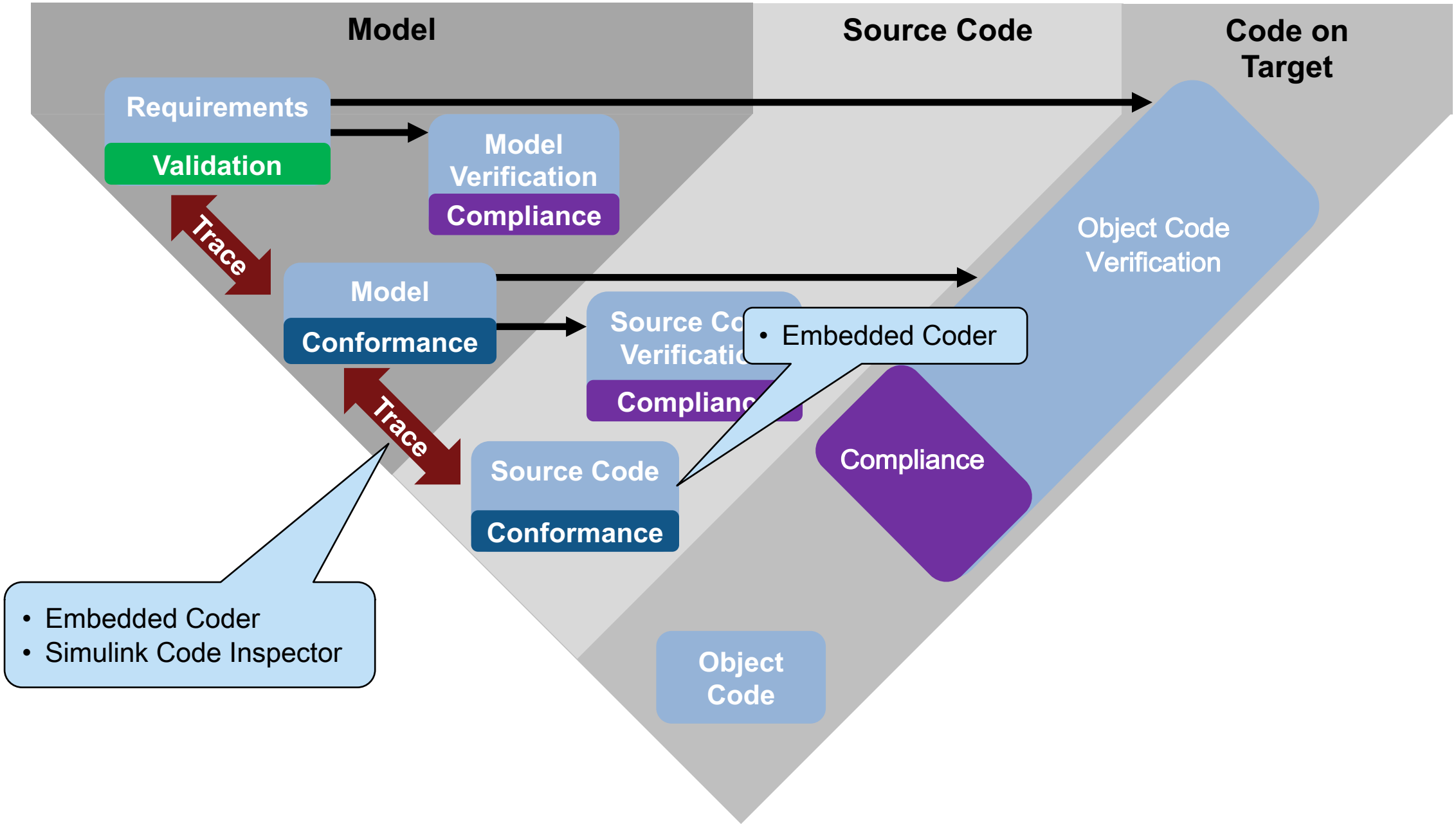
Confirm design is fully tested



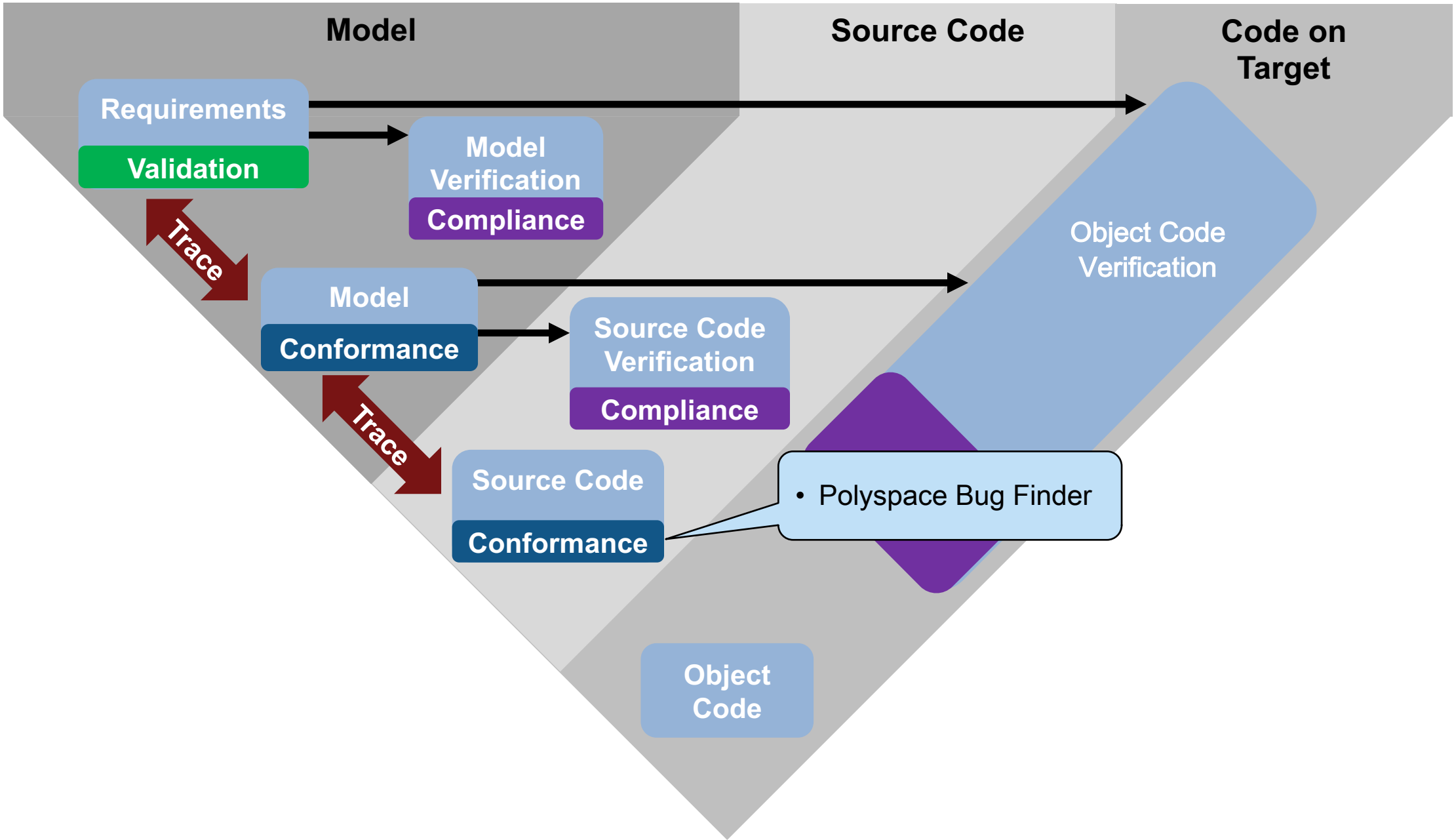
			D1	C1	MCDC	TBL
1. basic_inport_outport	27	73%		38%	25%	30%
2. . . . shift_logic	26	73%		38%	25%	30%
3. . . . . SF: shift_logic	25	73%		38%	25%	30%
4. . . . . SF: gear_state	9	81%		NA	NA	NA
5. . . . . SF: selection_state	16	67%		38%	25%	30%
6. . . . . SF: calc_th	6	100%		NA	NA	30%
7. . . . . selection_state.calc_th	6	100%		NA	NA	30%
8. . . . . Look-Up	5	100%		NA	NA	NA

Tool: Simulink Verification and Validation, Simulink Test

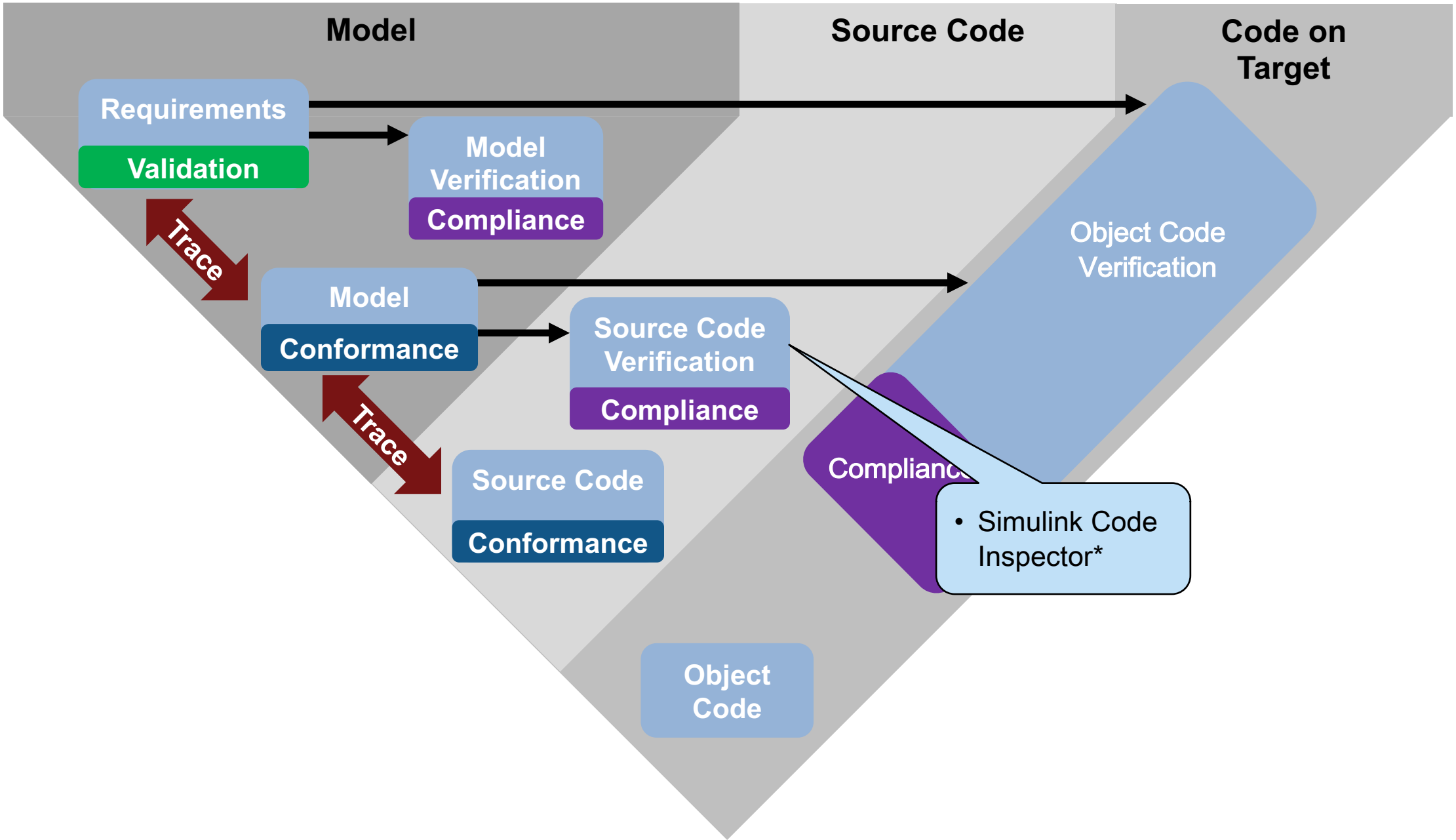
Model-Based Software Development Workflow (DO-178C)



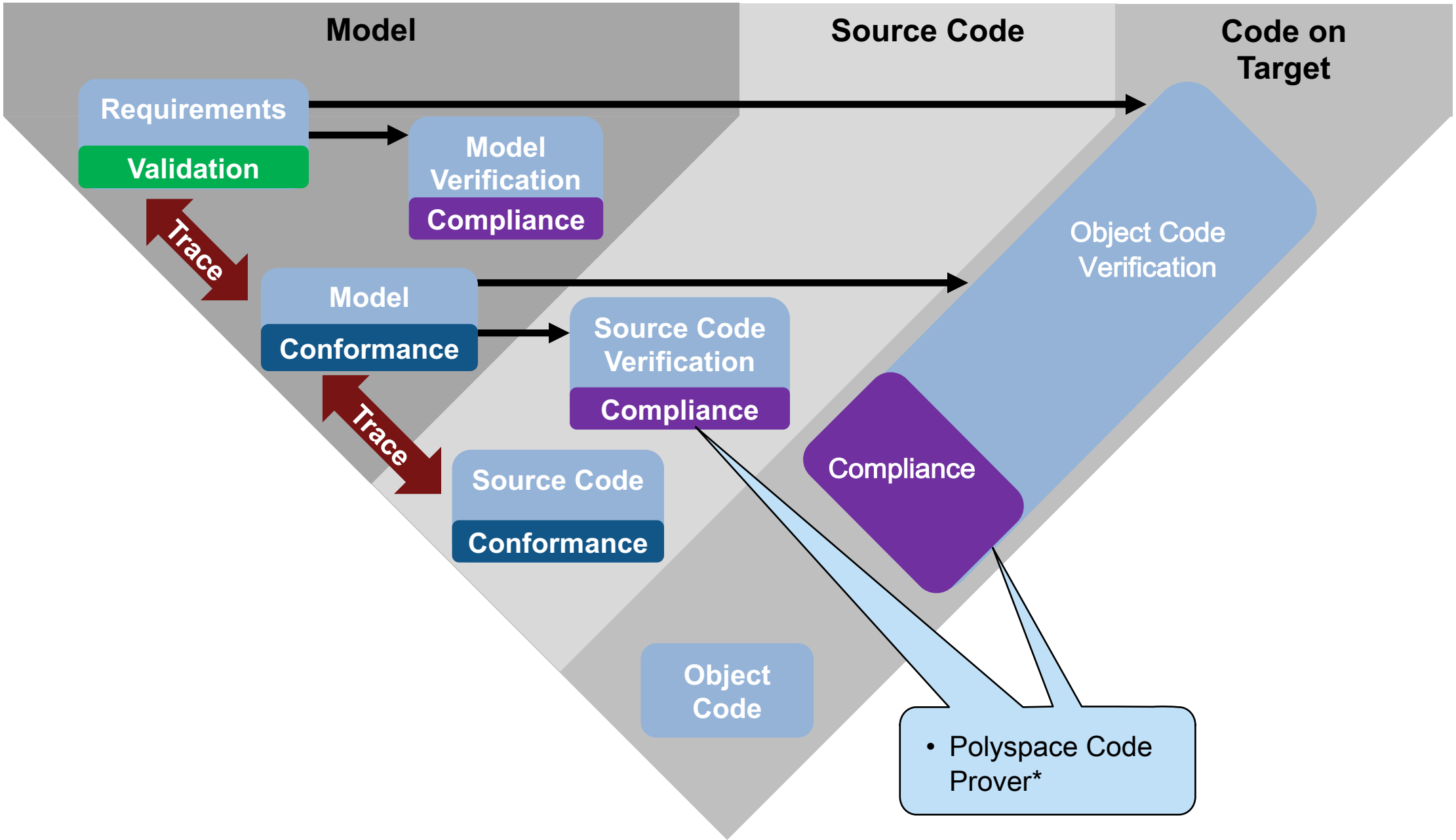
Model-Based Software Development Workflow (DO-178C)



Model-Based Software Development Workflow (DO-178C)



Model-Based Software Development Workflow (DO-178C)



Verify robustness of integrated code

Polyspace formal methods static analysis

Green: reliable
safe pointer access

Red: faulty
out of bounds error

Gray: dead
unreachable code

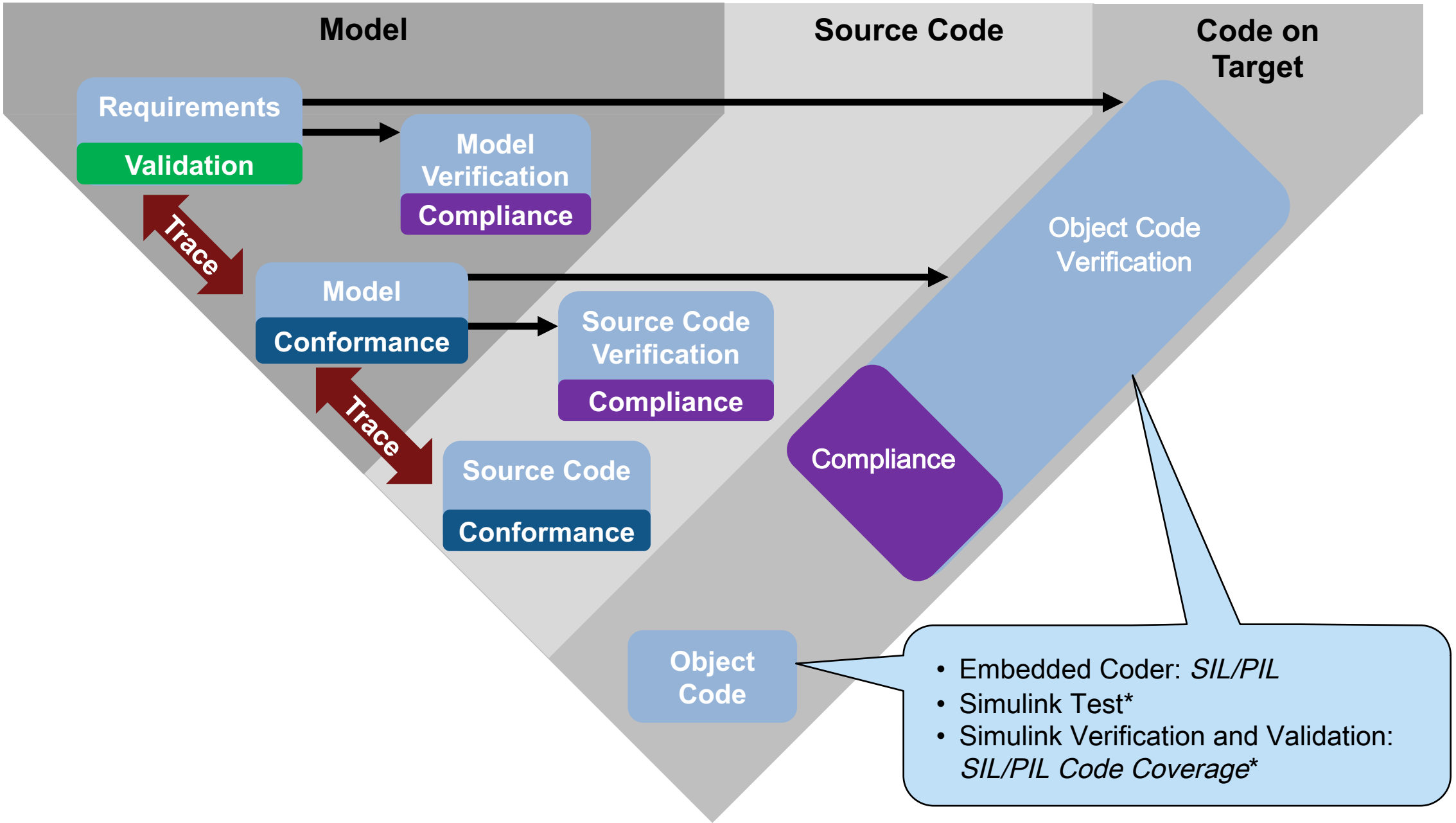
Orange: unproven
may be unsafe for some conditions

Purple: violation
code rules

```
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;
  }
}
```

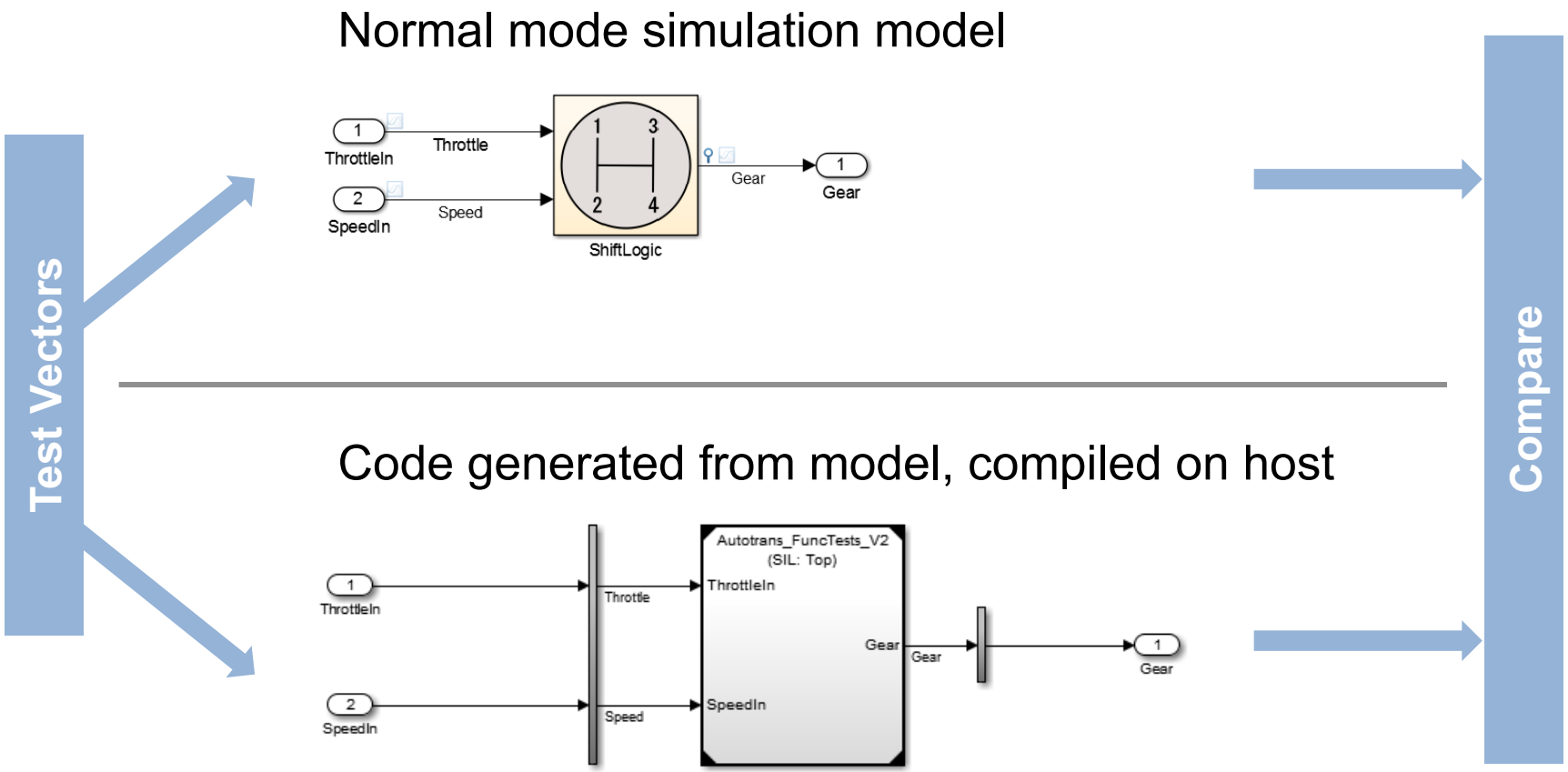
Tool: Polyspace Code Prover

Model-Based Software Development Workflow (DO-178C)



Confirm design and code consistency

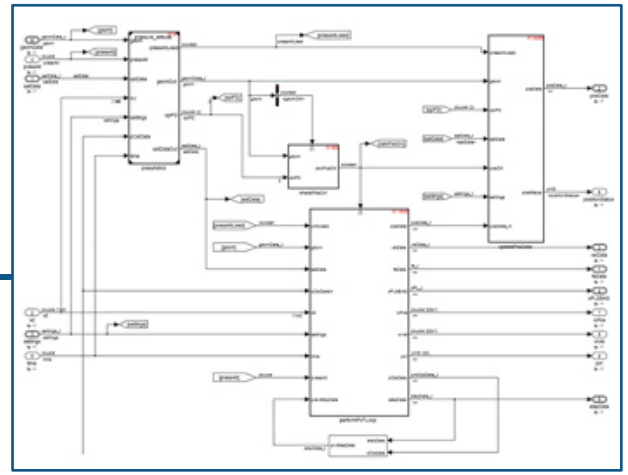
Execute all tests to check equivalence of model to code



Tool: Simulink Test

Examples of Successful Certifications with MathWorks Tools

Septentrio Streamlines DO-178B Certification with MATLAB and Simulink



Simulink model of part of the AiRx2 system.

Challenge

Obtain DO-178B certification for a GNSS-based landing system for precision aviation applications

Solution

Use Model-Based Design with MATLAB and Simulink to trace requirements, architect system components, simulate the design, and generate and verify source code

Results

- Design test cases reused on generated C source code
- Models verified via simulation, ensuring virtually bug-free code
- Key SOI-1 certification milestone achieved

"Model-Based Design enabled us to streamline the certification process by tracking requirements, verifying the design using simulation, and maintaining the system model as the single source of truth throughout development."

Jan D'Espallier
Septentrio

Link to article

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

The Eurocopter EC130 helicopter.

Challenge

Speed the development, validation, and verification of DO-178B certified helicopter flight software

Solution

Use Model-Based Design to model the system design and software design, and to generate flight code

Results

- Software testing time cut by two-thirds
- Requirements stabilized earlier
- Certified flight software automatically generated

"We use our system design model in Simulink for ARP4754 to establish stable, objective requirements. We save time by using the model as the basis for our software design model for DO-178—from which we generate flight code—and reusing validation tests for software verification."

Ronald Blanrue
Eurocopter

Link to user story

Airbus Develops Fuel Management System for the A380 Using Model-Based Design

Airbus A380, the world's largest commercial aircraft.

Challenge

Develop a controller for the Airbus A380 fuel management system

Solution

Use MATLAB, Simulink, and Stateflow for Model-Based Design to model and simulate the control logic, communicate the functional specification, and accelerate the development of simulators

Results

- Months of development time eliminated
- Models reused throughout development
- Additional complexity handled without staff increases

"Model-Based Design gave us advanced visibility into the functional design of the system. We also completed requirements validation earlier than was previously possible and simulated multiple simultaneous component failures, so we know what will happen and have confidence that the control logic will manage it."

Christopher Slack
Airbus

Link to user story

MathWorks can Help with Certification

MathWorks Consulting

Certification Advisory Service Overview

- The Certification Advisory Service will help you to:
  - Leverage Model-Based Design and supporting tools to their fullest extent to maximize ROI
  - Reduce duplicated and manual effort
  - Avoid common pitfalls by providing proven best practices

- Actual customer ROI:
  - Up to 30% time savings of requirements review and verification
  - Reduction of time-to-flight by 20%
  - Automation of certification activities

http://www.mathworks.com/services/consulting/proven-solutions/do178.html

MathWorks Consulting

Certification Advisory Service

Certification Advisory Service

Phase 1:
Coaching and Instruction

Phase 2:
Processes Gap Analysis and Improvement Plan

Phase 3:
Hands-on deployment support

Service Details

- Present the best practices for each verification and traceability activity in the MBD workflow
- Each activity is covered via a PowerPoint slide deck
- Industry-specific demos supplement the slides

- Scope and objectives of Analysis are **jointly defined**
- Onsite interviews scheduled, **pre-analysis survey completed by participants**
- Gap Analysis: **onsite interviews** with participants
- Develop & **deliver gap closure recommendations**

- Help the customer apply the concepts, best practices and recommendations to **their application**
- Examples: modeling, simulation, code generation, establishing a model architecture, developing automation, integration with 3rd party tools, PIL