

The ANSYS logo is centered in the upper half of the slide. It consists of the word "ANSYS" in a bold, sans-serif font. The letters "AN" are white, and "SYS" are yellow. A registered trademark symbol (®) is located to the upper right of the "S". The logo is set against a solid black rectangular background.

**ANSYS®**

# A Certified Ada Toolchain for High-Integrity Application Development

X. Fornari

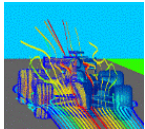
SCADE Suite Product Manager



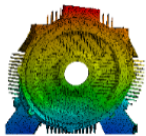


# World Leader in Simulation

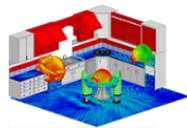
## Leading Disciplines



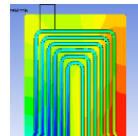
Fluids



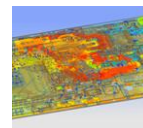
Structures



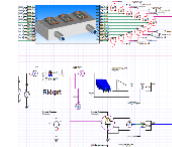
Electromagnetics



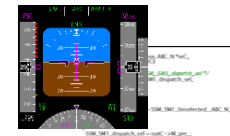
Thermal



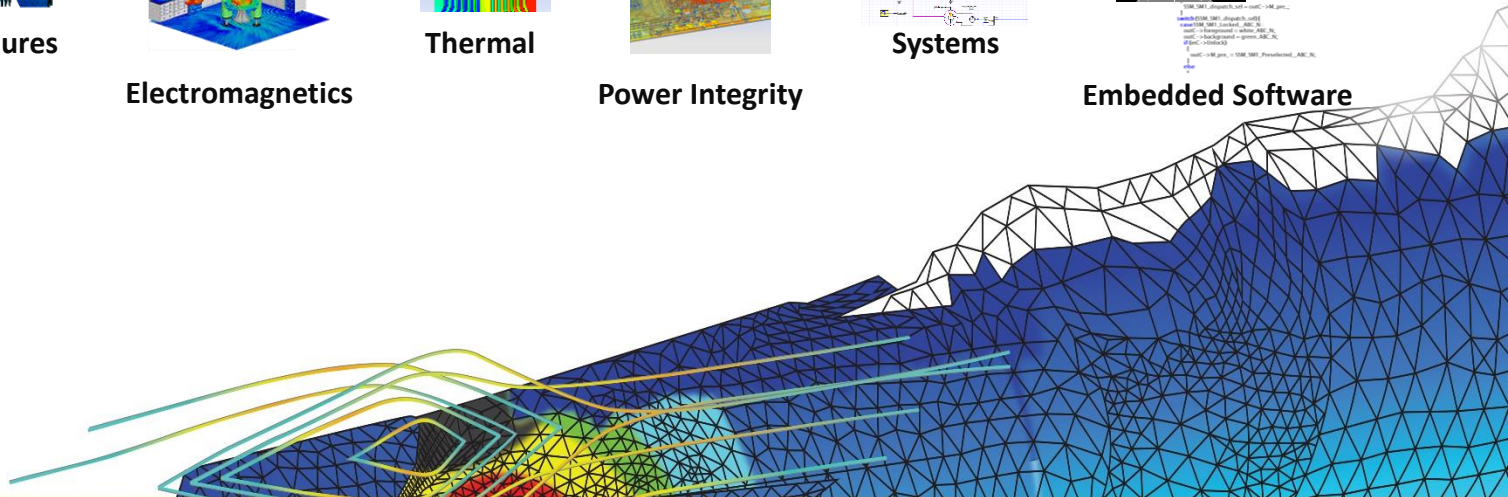
Power Integrity



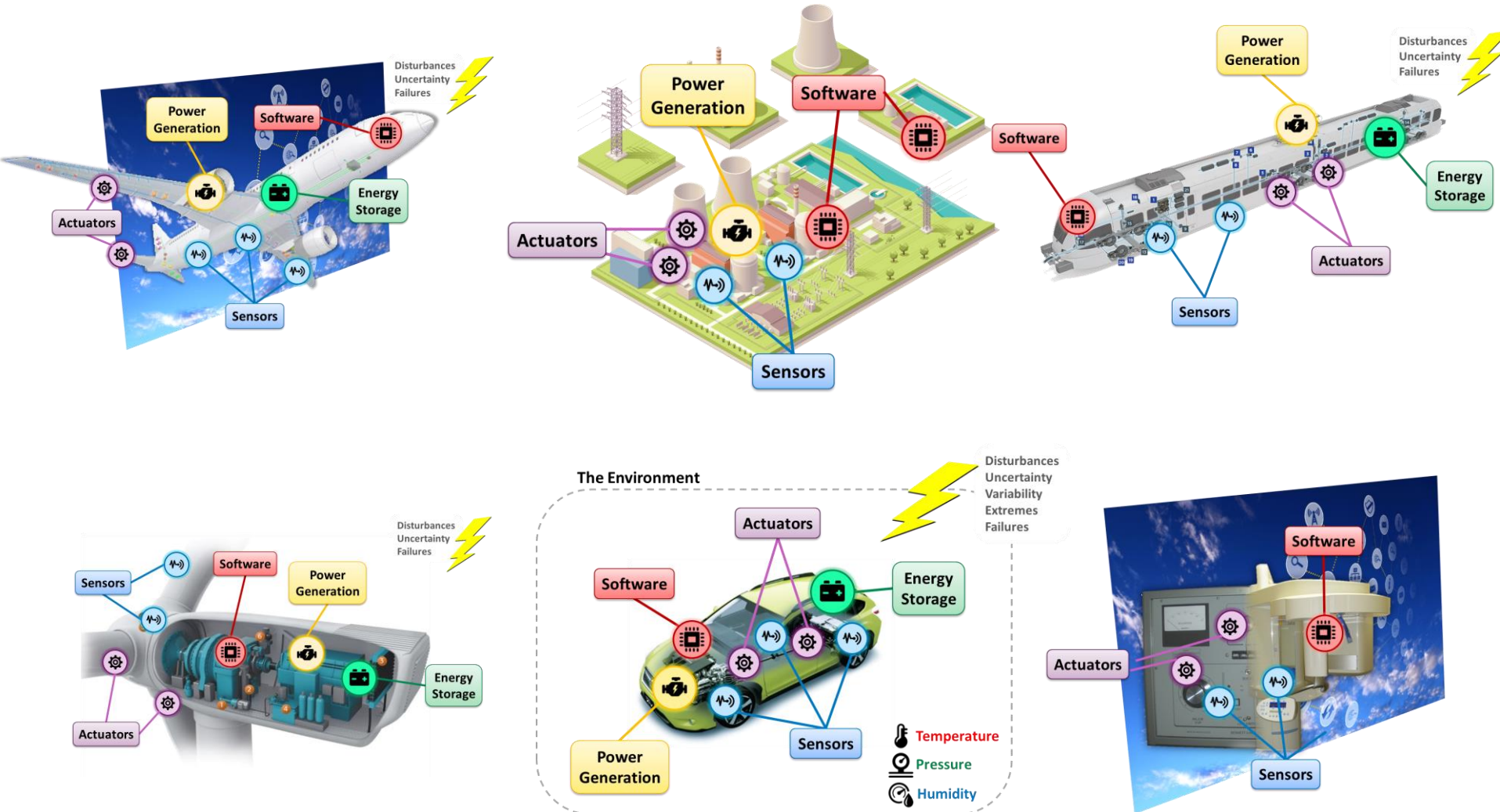
Systems



Embedded Software

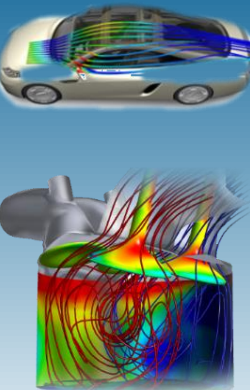


# Systems – They're Everywhere ...

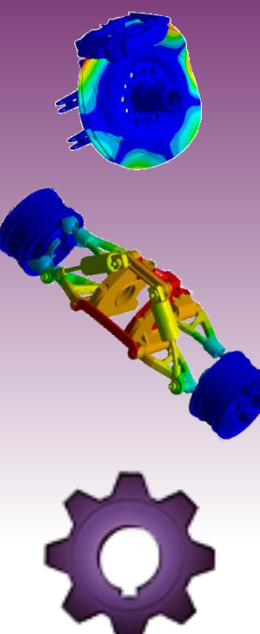


# ANSYS Enables Systems

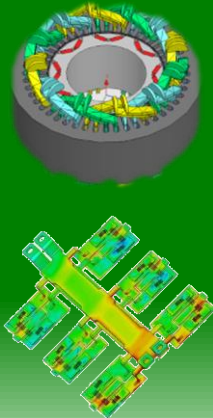
*From Comprehensive Component-Level Design & Simulation ...*



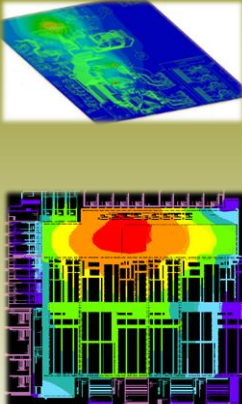
FLUIDS



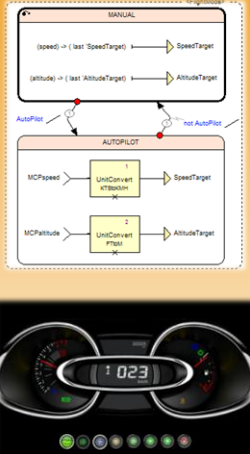
STRUCTURES



ELECTRONICS



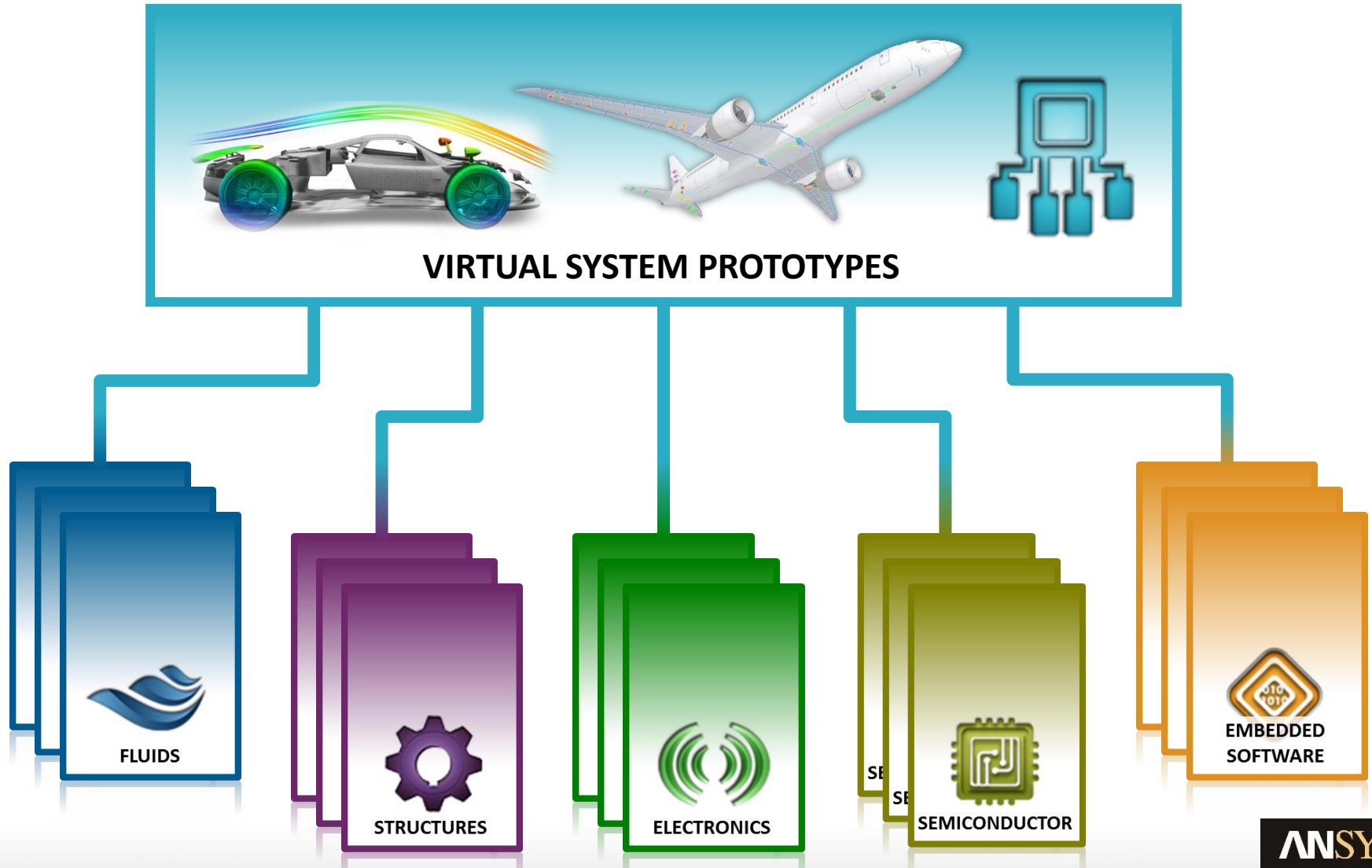
SEMICONDUCTOR



EMBEDDED SOFTWARE

# ANSYS Enables Systems

*... To Complete Systems Simulation*



# Our Vertical Market Focus



## Aerospace & Defense

500% increase in SLOC in aerospace in 10 years



## Automotive

10 M software lines of code (SLOC) in modern vehicles



## Railways

Ever increasing system & software certification costs and project delays/costs overrun



## Industrial Equipment

More than 380K software and system engineers work in the oil and gas industry



## Energy & Nuclear

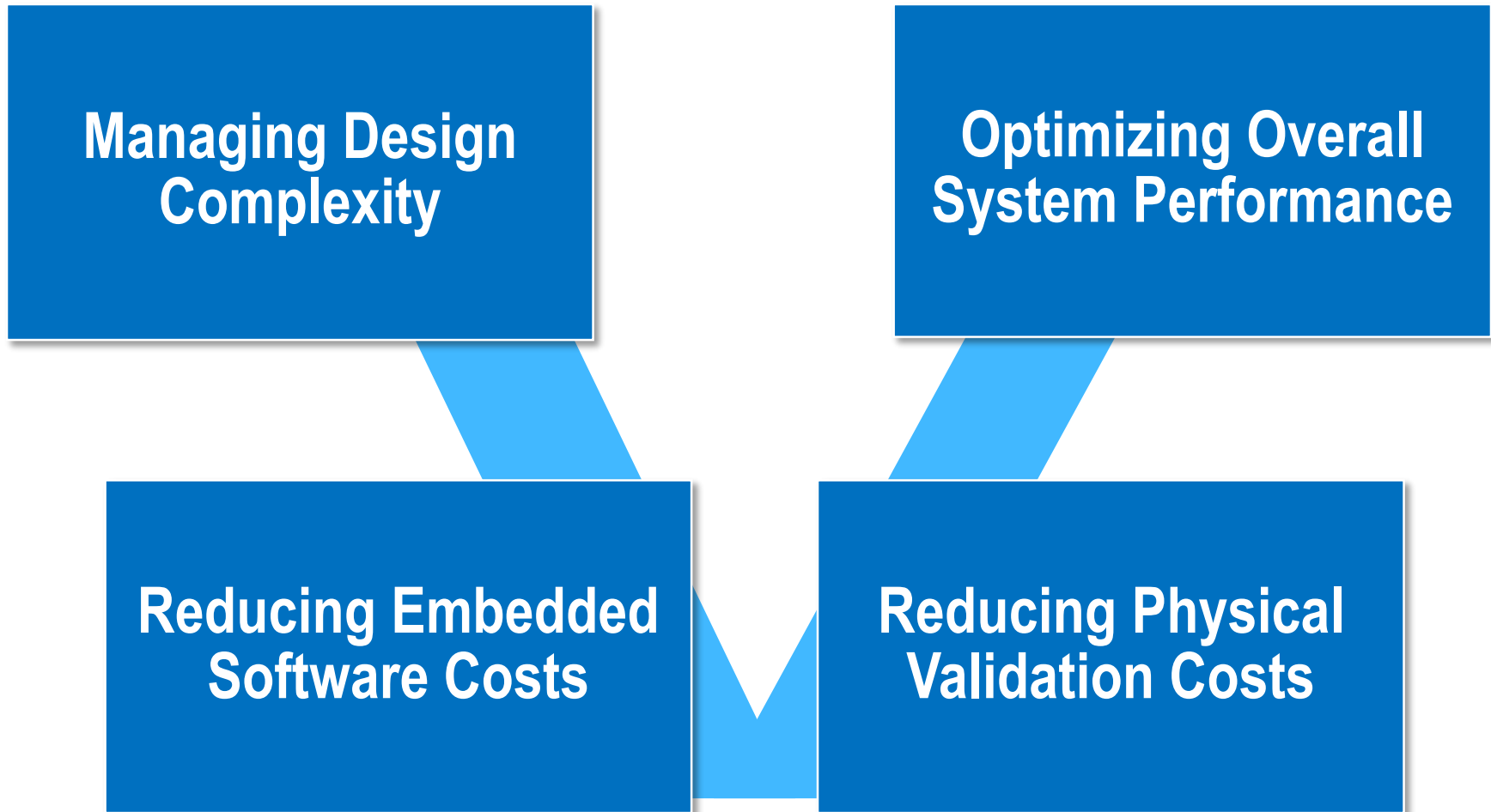
Software-Based Instrumentation and Controls have become state of the art



## Medical

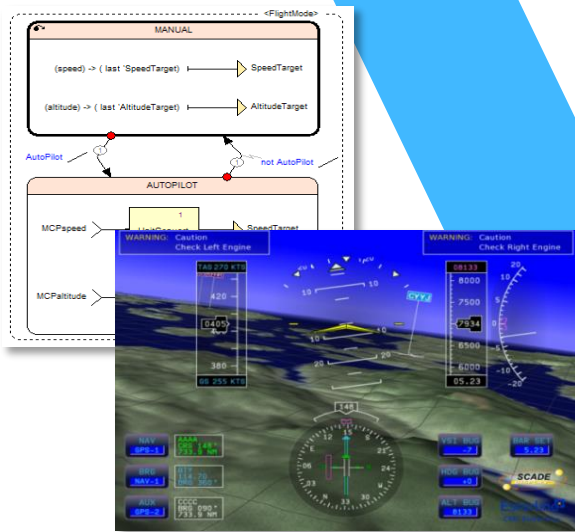
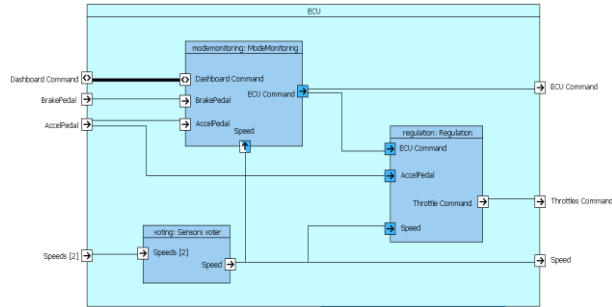
More than 70% of product innovation in medical devices is software driven

# Systems & Software Development Challenges



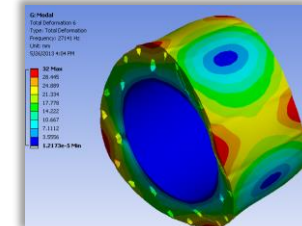
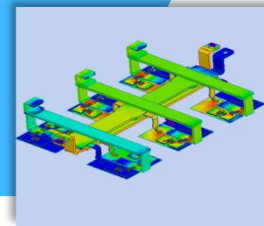
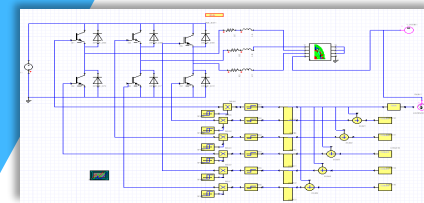
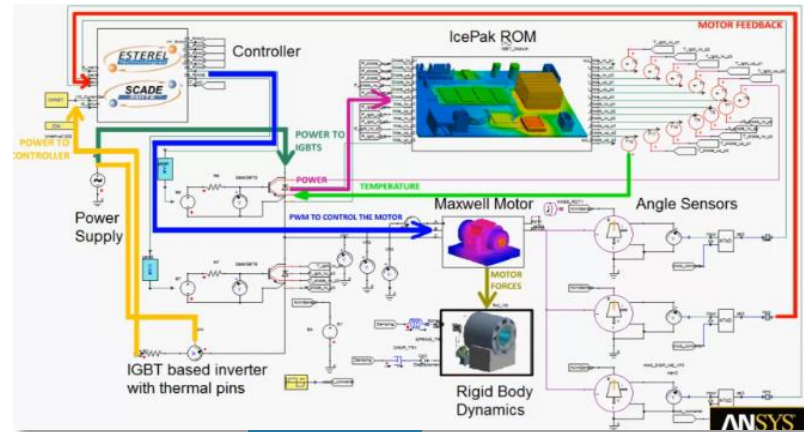
# ANSYS Model-Based Engineering Solutions

## Model-Based Systems Engineering



## Model-Based Software Engineering

## Multi-Physics & System Simulation

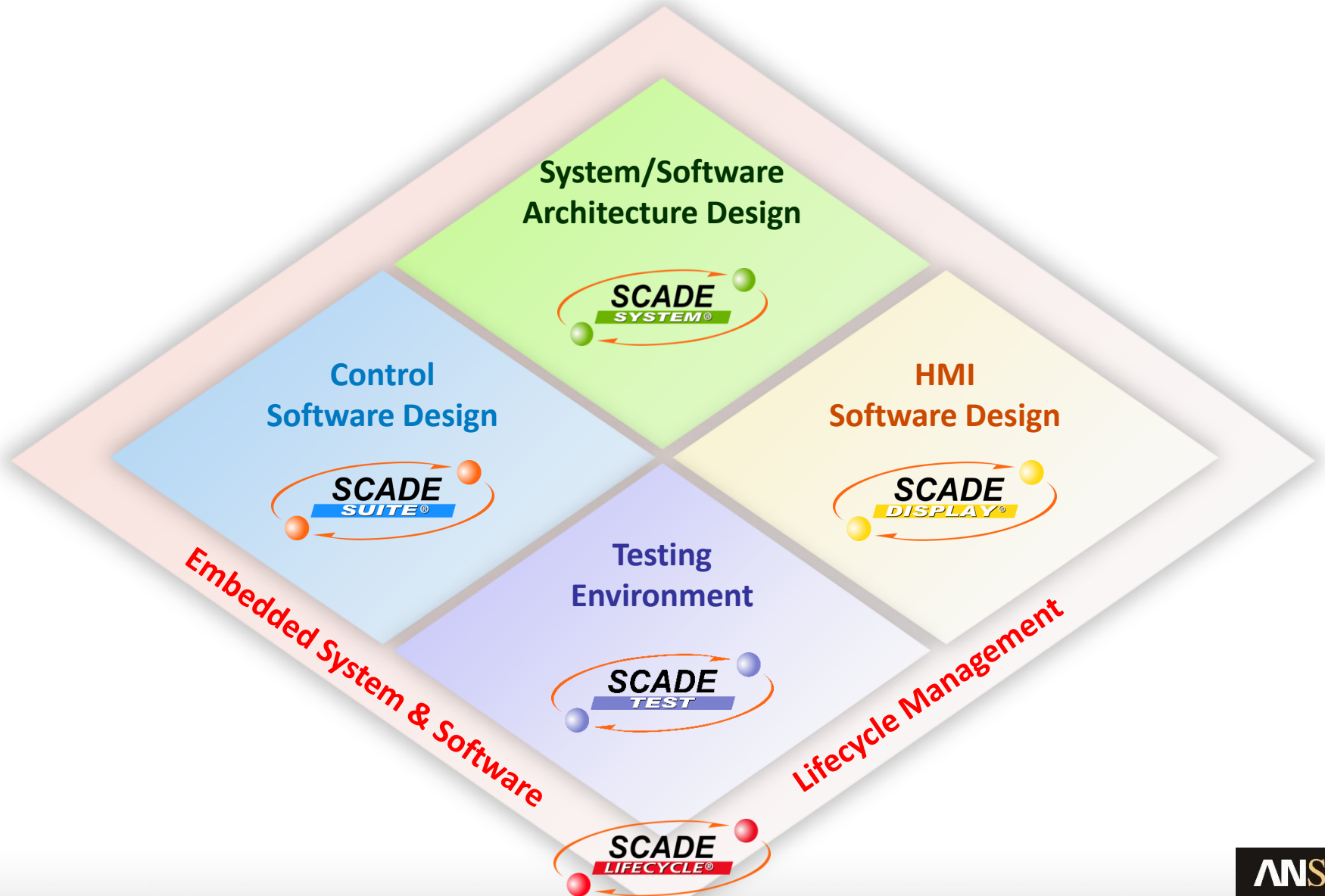


## 3D Physical Simulation

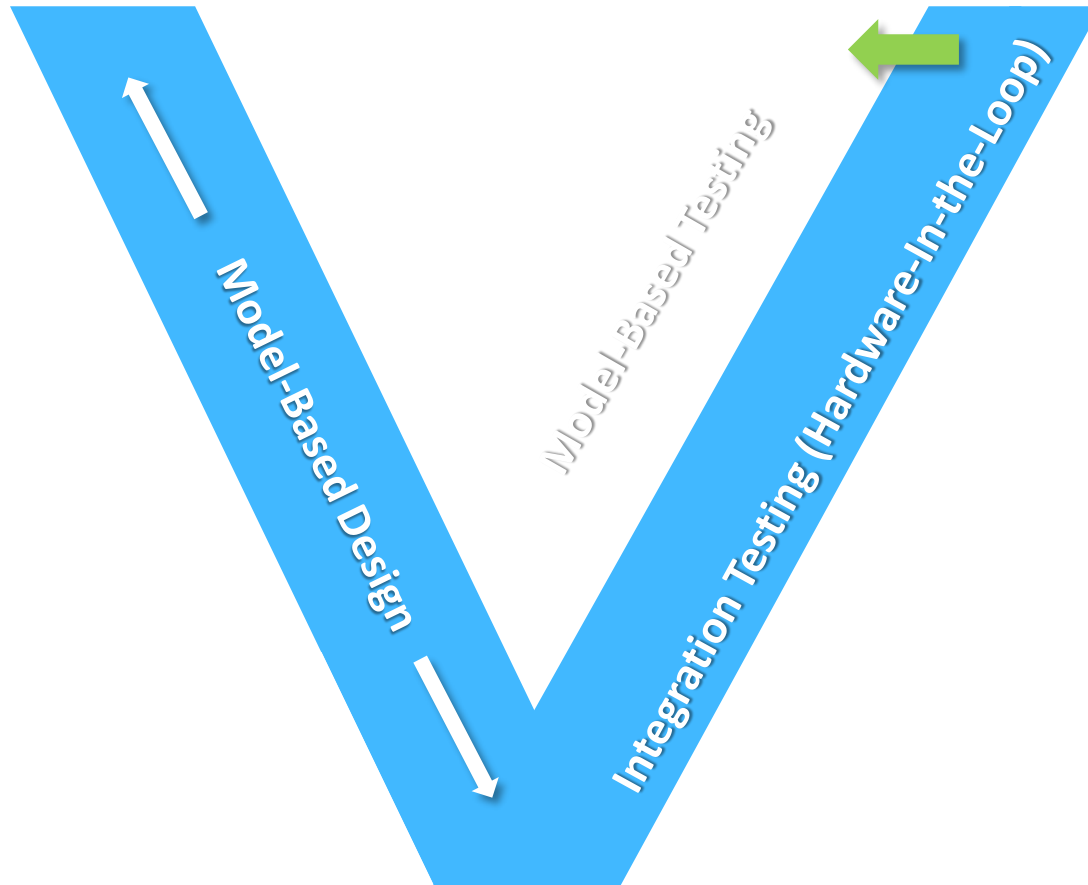




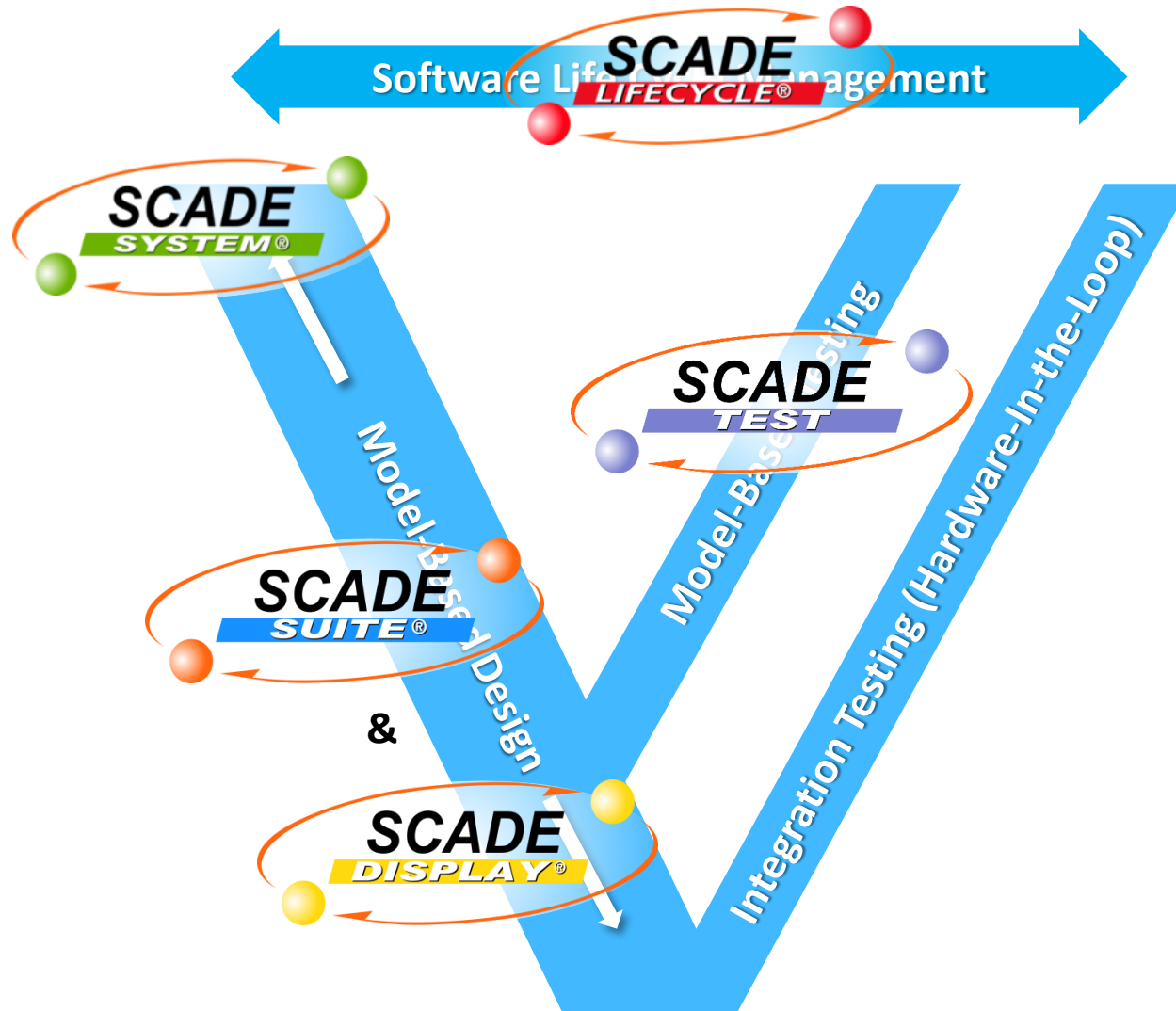
# ANSYS SCADE Products



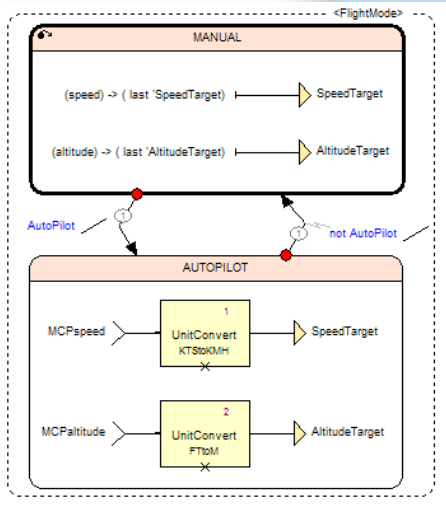
# Software V-Cycle with ANSYS SCADE



# Software V-Cycle with ANSYS SCADE



# ANSYS SCADE Suite



## Control Software Design



Model Checks



Formal Verification



Debug & Simulation



Plant Model Co-simulation (incl. FMI)



Time & Stack Optimization



HIL/SIL/PIL Integration

Calibration

SCADE Suite KCG  
C & Ada

RTOS Adaptors



Object Code & Compiler Verification



DO-178B & C  
IEC 61508  
EN 50128  
ISO 26262  
Certification Kits

PROTOTYPE &  
DESIGN

VERIFY

GENERATE

# ANSYS SCADE Test



**Rapid  
Prototyping**

An icon representing interactive test creation, showing a screenshot of a test case table.

	A	B	C
1	Project	Name	Description
2	Project A	Test Case A	Steps: Do some action. Success: Some expected result. <strong>Do some action.</strong><strong>Do some action.</strong><strong>Success:</strong>Some expected result.
3	Project A	Test Case B	
4			

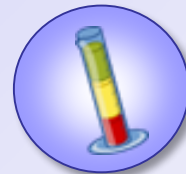
**Interactive  
Test Creation**

**PROTOTYPING &  
TEST CREATION**

**Testing  
Environment**



**Test Execution  
on Host**



**Model Coverage**

**HOST  
EXECUTION**



**Test Execution  
on Target  
(RTRT, LDRA,  
VectorCAST &  
Generic)**

**TARGET  
EXECUTION**

# Certified Ada Toolchain: Modeling

- **A single input language** for two target languages
  - Retarget your existing model to a different language to adapt to various platforms
  - Implement diversity by using two different compilation and execution flow from the same model
  - Supports different generated object names for C and Ada
  - Propagate Ada pragmas to generated code
- **Define imported operators** in Ada
  - Reuse your Ada legacy
  - Use Ada code to extend SCADE expressiveness
  - SCADE libraries using imported code now also have an Ada implementation

# Certified Ada Toolchain: V&V

- **Native Ada Simulator**
  - Supporting Ada imported code
  - Generates Simulation Ada Code
  - Exactly the same principles as C-based simulation
- **SCADE Test fully supports Ada V&V activities**
  - SCADE Test Model Coverage
  - SCADE Test Execution Environment and Test Harness Generation support Ada generated code
  - Validation teams can develop their **functional tests independently** from the actual code generation
- **SCADE Test is qualified DO-330 TQL-5 for C and Ada**

# Certified Ada Toolchain: Code Generation

- **SCADE Suite KCG for Ada generates SPARK-compliant code, compatible with any Ada 95 compiler**
  - **Flexibility:** separate bodies
  - **Readability:** named parameters
  - **Efficiency:** procedures or functions
  - **Improved traceability file** also for Ada
- **SCADE Suite KCG Ada code generator certification**
  - **Full DO-178C/DO-330 Certification Kit**
  - **ISO 26262, EN 50128, IEC 61508**



# SCADE Suite IDE Overview

The screenshot displays the SCADE Suite IDE interface for a project named "CruiseControl.vsw". The main workspace shows a state machine diagram titled "Top Level of the Cruise Control application". The diagram is structured as follows:

- Off State:** A state with an initial transition to the "On" state triggered by the event "On".
- On State (SM2):** Contains a "CruiseRegulation" block (SM3) that takes "local\_CruiseSpeed" and "Speed" as inputs and outputs "ThrottleCmd". It transitions to the "StandBy" state when "StbyCondi on" occurs.
- StandBy State (SM3):** Contains an "STDBY" block that outputs "CruiseState". It transitions back to the "On" state when "not StbyCondition" occurs.
- Interrupt State (SM1):** Contains an "Interrupt" block that outputs "CruiseState". It is triggered by "Brake > PedalsMin" and transitions back to the "StandBy" state when "Resume" occurs.
- Logic:** A block labeled "1" (CruiseSpeedMgt) takes "Set", "QuickAccel", "QuickDecel", and "Speed" as inputs and outputs "local\_CruiseSpeed" and "CruiseSpeed". A block labeled "2" (CruiseRegulation) is also present.

The IDE interface includes a menu bar (File, Edit, View, Operator, Insert, Layout, Project, Tools, Navigate, Window, Help), a toolbar, a workspace with a project tree on the left, and a right-hand sidebar with a "Shortcuts" panel and a "Mathematical" panel containing operators like Plus, Minus, Multiplication, etc. The bottom of the window features an "Output" window showing "Loading project CruiseControl.etp... Successfully loaded project CruiseControl.etp" and a "Properties" panel for the selected element.

# SCADE System - SCADE Suite Integration

## An Integrated Workflow for SW-intensive Systems

The screenshot displays the SCADE Suite Advanced Modeler interface for a project named "Fighter.vsw - SCADE - IBD\_FighterFunction". The interface is divided into several panes:

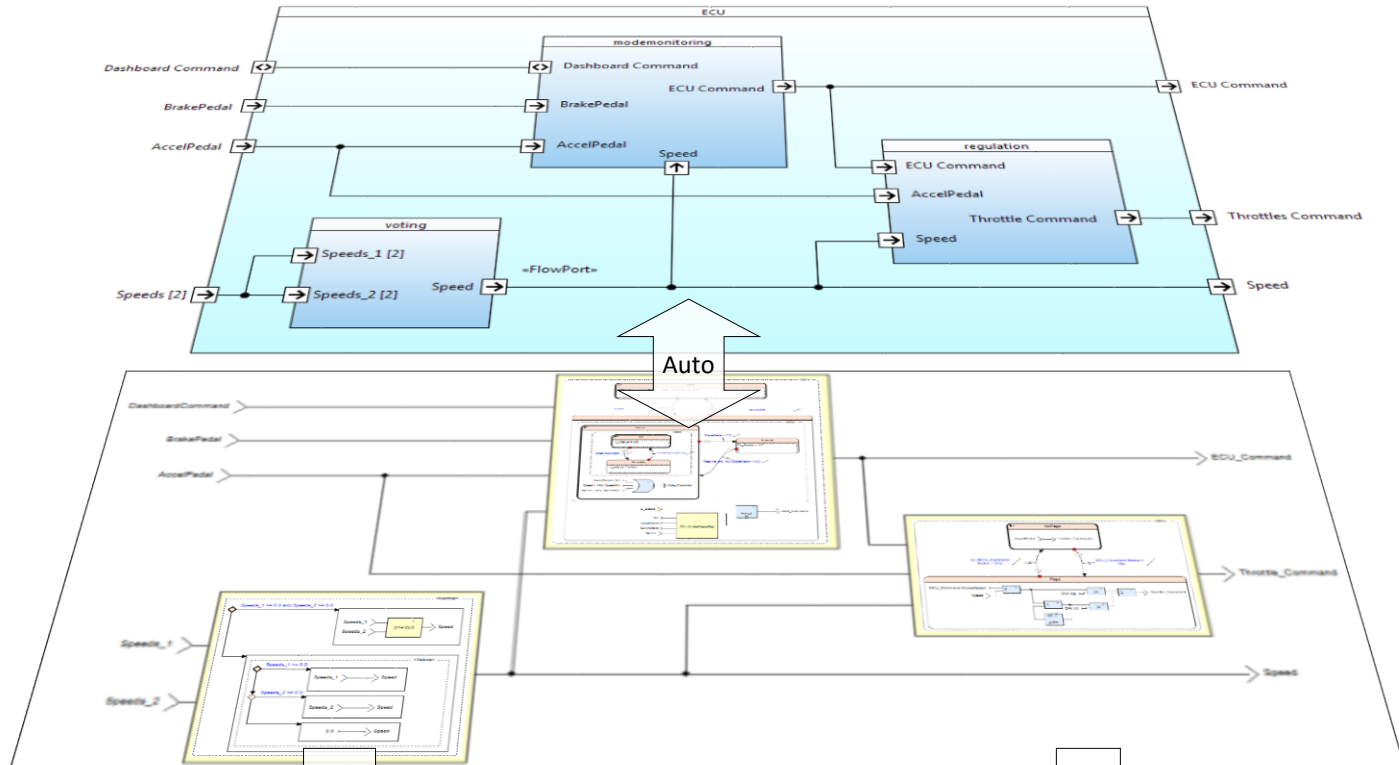
- Workspace:** Shows a hierarchical tree of components. The "MC" component is expanded to show sub-components like "Operators", "MC\_ManageTracks", "RADAR", and "IFF".
- Requirements:** Lists requirements for the "MC\_ManageTracks" component, such as "The MC system shall be able to manage in the MC\_ManageTracks component, tracks data coming from the radar and IFF: - Receiving tracks data from the radar and IFF: - Sending the index of highest priority tracks to the MC\_ManageGun component - Sending position / speed vector of the".
- Diagram:** A block diagram of the "Fighter" system. It shows a "Mission Computer" block containing "DetectTracks", "GenerateTracks", "ManageTracks", and "DisplayTracks" sub-blocks. A "ManageTracks" block is connected to a "ManageGun" block. A large green arrow points from the "ManageTracks" block to the "ManageGun" block, and a blue arrow points from the "ManageGun" block to the "ManageTracks" block, indicating a bidirectional flow of information.
- Logic Diagram:** A detailed logic diagram for the "MC\_ManageGun" component. It shows inputs like "GunShotButton", "IndexMissionTrackHighestPriority", and "MissionTracks". The logic involves several logic gates (AND, OR, NOT) and a "FindIndexTrackNumber" block. The output is "AllowGunLock".
- Properties:** Shows the properties of the selected component, including "KCG Pragn", "Code Integ", "Coverage", and "Traceability".
- Output:** Displays the output of the simulation, including "Messages", "Problems", "Error Log", "Coverage", "Dump", "Build", "Simulator", and "Script".

# Integrated Workflow for SW-intensive Systems

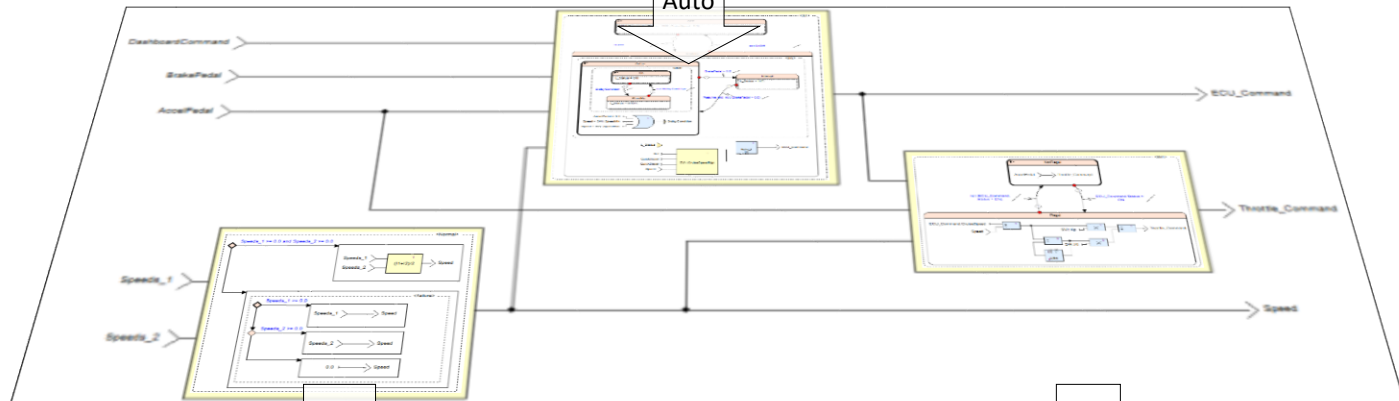
## Systems



## SW Architecture



## SW Design



## SW Coding

```

/* Architecture::Regulation/
void Regulation_Architecture(
/* ECU_Command/ */
tECU_cmd_Architecture *ECU_Command,
/* AccelPedal/ */
tPercent_Architecture AccelPedal,
/* Speed/ */
tVehicleSpeed_Architecture Speed,
outC_Regulation_Architecture *outC)
{
  kcg_float32 tmp;
  /* SMI:Regul: L3/ */
  kcg_float32_L3_Regul_SMI;
  /* SMI: */
  SSM_ST_SMI SM1_state_act;
  /* SMI: */
  kcg_bool SM1_reset_act;
  /* SMI: */
  switch (outC->SM1_state_nxt) {
  case SSM_st_NotRegul_SMI :
    SM1_reset_act = (*ECU_Command).Status == ON_Architecture;
    if (SM1_reset_act) {
      SM1_state_act = SSM_st_Regul_SMI;
    }
  } else {
    SM1_state_act = SSM_st_NotRegul_SMI;
  }
  break;
}
    
```

```

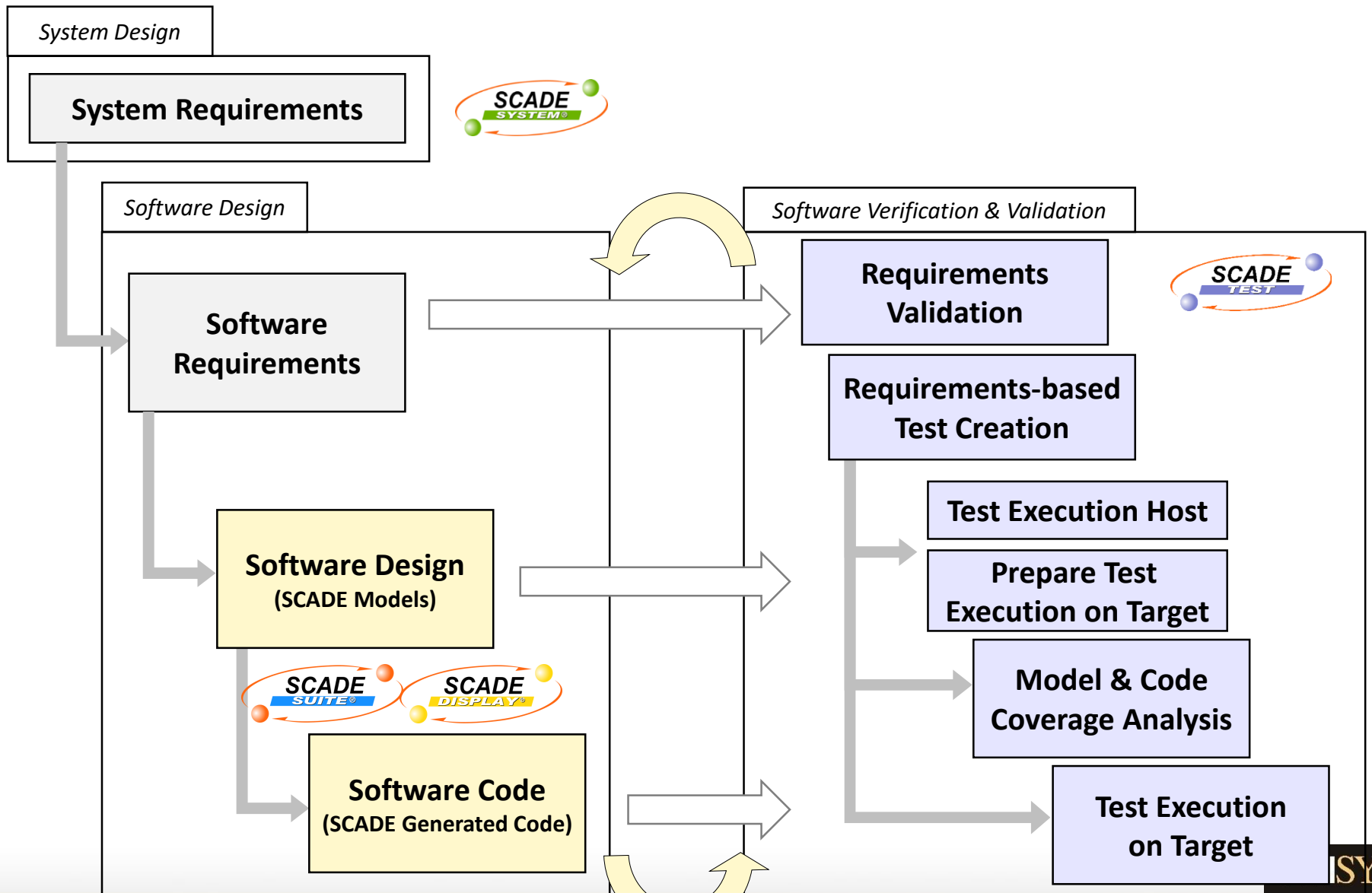
-- Architecture::Regulation/
procedure Regulation(
-- ECU_Command/
ECU_Command : in tECU_cmd;
-- AccelPedal/
AccelPedal : in tPercent;
-- Speed/
Speed : in tVehicleSpeed;
Ctx : in out Context_Regulation)
is
  -- SMI:
  SMI_state_act : Kcg_Types.SSM_ST_SMI;
  -- SMI:
  SMI_reset_act : Boolean;
  -- SMI:Regul: L3/
  L3 : Kcg_Config.Kcg_Float32;
  tmp : Kcg_Config.Kcg_Float32;
begin
  case (Ctx.SMI_state_nxt) is
  when Kcg_Types.SSM_st_NotRegul =>
    SMI_reset_act := ECU_Command.Status = Kcg_Types.ON;
    if(SMI_reset_act) then
      SMI_state_act := Kcg_Types.SSM_st_Regul;
    else
      SMI_state_act := Kcg_Types.SSM_st_NotRegul;
    end if;
  end if;
end if;
    
```



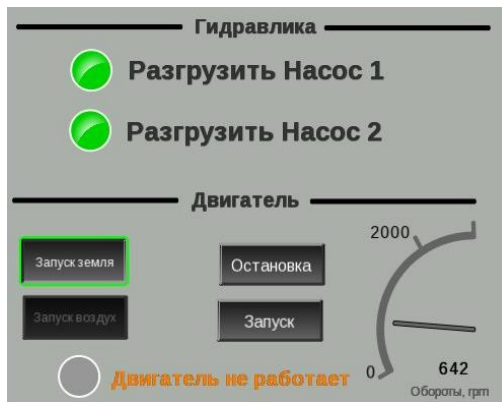
# Addressing Main Application Software Testing Challenges

- Software requirements are **often inadequate to meet user needs** or incomplete
- Test creation and maintenance is **very time consuming**
- Late testing activities generate **expensive design rework**
- Test execution and results analysis are **often very manual activities**
- Testing effort is sometime **not sufficient and software errors remains**. How to know if testing effort is adequate?
- Test execution infrastructure has **to adapt to a variety of hardware targets**

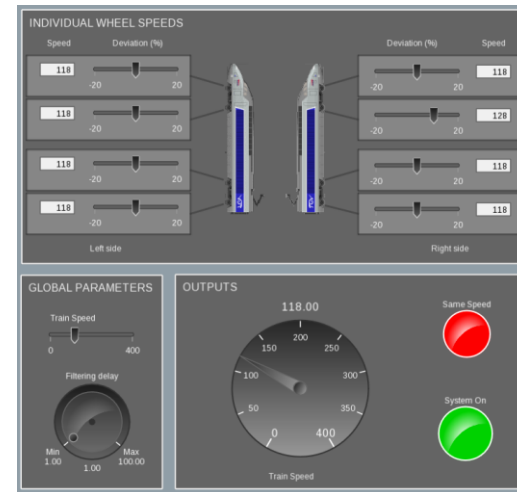
# SCADE Test Workflow Overview



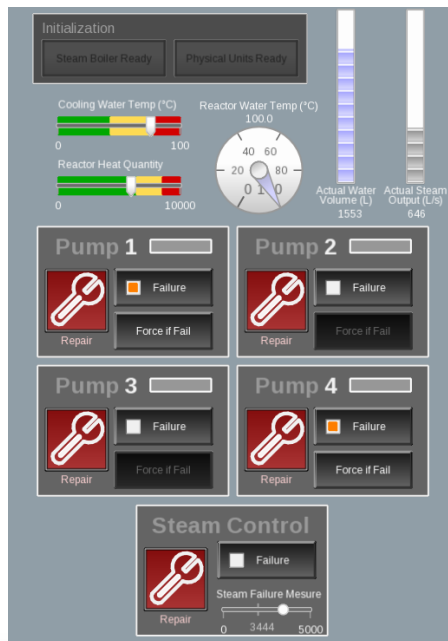
# Examples of Rapid Prototyper Panels



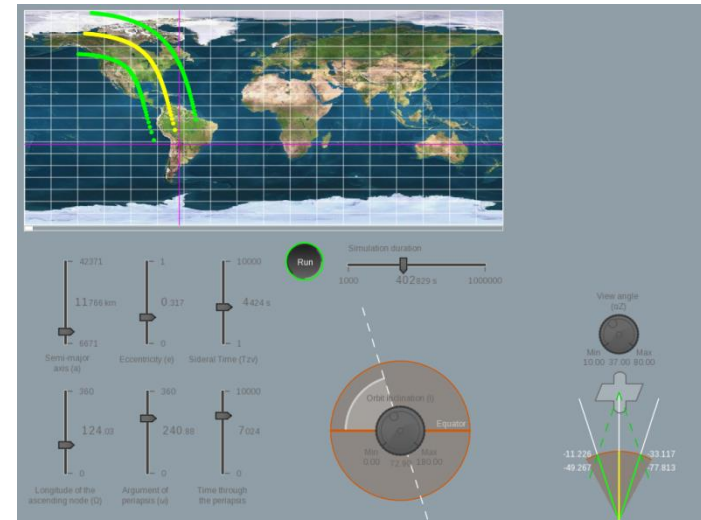
**Aerospace - Hydraulic Pump Control Panel**



**Rail - Communication Based Train Control Panel**



**Energy - Steam Boiler Control Panel**



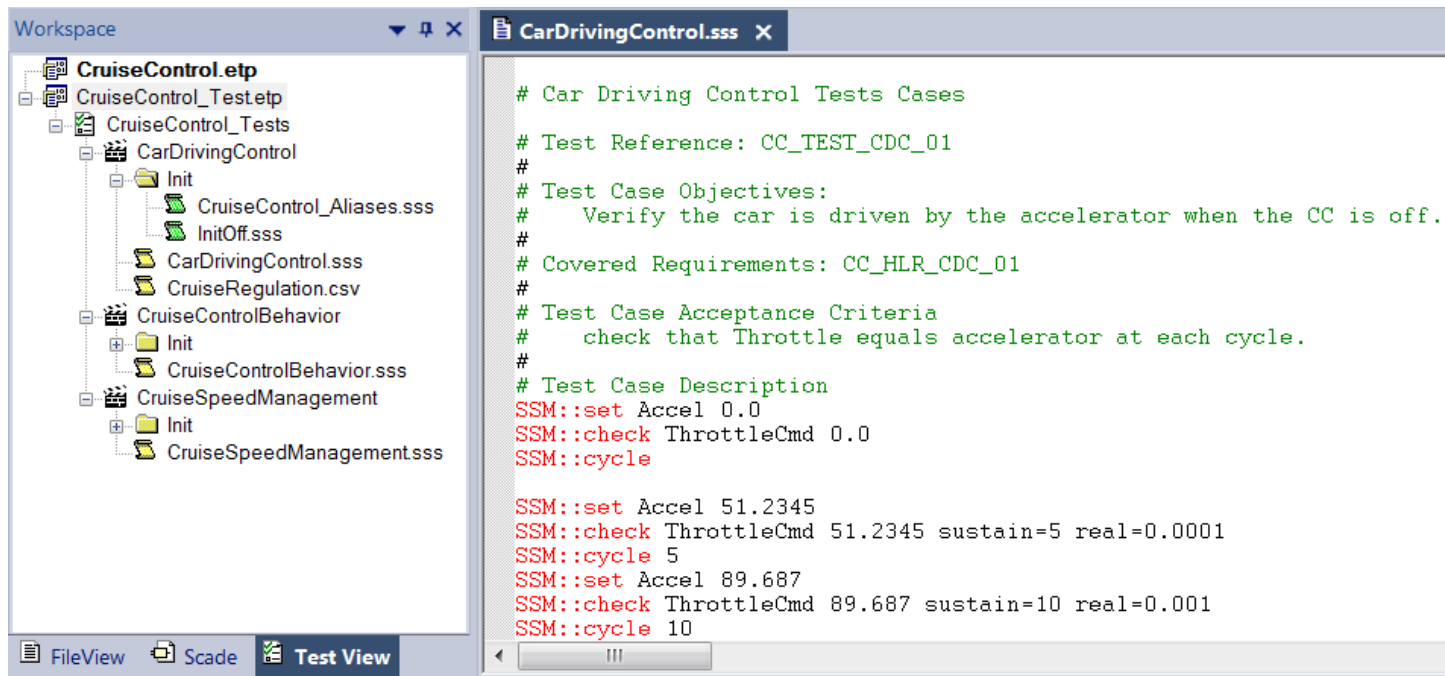
**Space - Orbital Simulation Cockpit**

← BACK

ANSYS®

# Test Case Creation & Management

- SCADE Test provides effective means to create requirements based Test cases
- Intuitive **graphical user-interface** enabling managing these test cases and Test Results.



# Test Execution on Host

Software requirements based tests can be executed on Host and expected results compared to actual results

Status	Step	Name	Actual Value	Expected Value	Tolerance	NBW	File:line
✘	1	CruiseSpeed	89.2900000000	89.2	0.001		..\TestScenarios\CruiseSpeedManagement.sss:line31
✔	3	CruiseSpeed	143.3300000000	143.33	0.1		..\TestScenarios\CruiseSpeedManagement.sss:line49
✔	5	CruiseSpeed	145.8300000000	145.83	0.1		..\TestScenarios\CruiseSpeedManagement.sss:line54
✔	7	CruiseSpeed	150.0	150.0		⚠	..\TestScenarios\CruiseSpeedManagement.sss:line56
✔	11	CruiseSpeed	39.4900000000	39.49	0.1		..\TestScenarios\CruiseSpeedManagement.sss:line78
✔	12	CruiseSpeed	36.9900000000	36.99	0.1		..\TestScenarios\CruiseSpeedManagement.sss:line81
✔	13	CruiseSpeed	34.4900000000	34.49	0.1		..\TestScenarios\CruiseSpeedManagement.sss:line84
✔	14	CruiseSpeed	31.9900000000	31.99	0.1		..\TestScenarios\CruiseSpeedManagement.sss:line87
✔	15	CruiseSpeed	30.0	30.0		⚠	..\TestScenarios\CruiseSpeedManagement.sss:line90
✔	16	CruiseSpeed	30.0	30.0		⚠	..\TestScenarios\CruiseSpeedManagement.sss:line90

3. Test Summary

Operator	Status	#OK/#Total
CruiseControl::CruiseControl	Failed	176/177

4. Detailed Test Results

4.1. CruiseControl::CruiseControl: Failed (#OK: 176/177)

<CarDrivingControl> Passed (#OK: 125/125)

Step	Data item	Tol.	Expected	Actual	Status	NBW	Location	File:line
1	ThrottleCmd		0.0	0.0	OK			1:16
2	ThrottleCmd	0.0001	51.2345	51.2344999999999997	OK			1:20
3	ThrottleCmd	0.0001	51.2345	51.2344999999999997	OK			1:20
4	ThrottleCmd	0.0001	51.2345	51.2344999999999997	OK			1:20
5	ThrottleCmd	0.0001	51.2345	51.2344999999999997	OK			1:20



# Model Coverage Analysis

- Capability to **merge results**, supporting modular model-based testing methodology
- Automatic **synthesis report generation**, for both model and code coverage
- The report gathers all configuration elements (SCADE model, test scenarios, test and coverage results)

The screenshot displays the SCADE software interface for model coverage analysis. The workspace tree on the left shows a test scenario 'test.etp' with several coverage cases, including 'A -> 0 (1/2)' which is highlighted with a red box. The main window shows a detailed view of the 'Root' node, displaying a table of activation conditions (AC) and coverage cases. A red box highlights a specific row in this table, which is annotated with a callout: 'Detailed view with activation conditions and coverage cases'. Below this, a logic diagram shows the internal structure of the model, with a red arrow pointing to the location of the highlighted case: 'Localisation in the model'. The output window at the bottom shows the following table:

Root/A -> 0		
	True	False
Overall		X

# Test Re-run & Execution on Target

- **Automatic generation of test harnesses**, from the same set of model-based test cases, for specific target test environments:

- LDRA TestBed®



- IBM RTRT®

- Vector Software VectorCAST®

- Interface driver to adapt to custom-made test infrastructures

- **Generated test harnesses consist in:**

- For LDRA TestBed : one TBrn file per Test Cases

- For IBM RTRT : one PTU file per Test Cases

- For VectorCAST : one VectorCAST file per Test Cases



# Summary of Benefits

- **SCADE 's Model-Based** approach of allows meeting **efficiently** all requirements of **high-integrity application**
- **Qualified Code Generation** of SCADE Suite and SCADE Display KCG is based on using the safety standards (DO-178C/DO-330, ISO 26262, EN 50128, IEC 61508, ...)
  - Only **COTS code generators** developed following the standards
  - Provides unique certification benefits (no code review, no low-level testing are needed, etc.)
- **Complete integration** with ANSYS 3D tools **for full virtual system simulation**
- **Bottom line is reduction of cost and time** to certification at any integrity level compared to manual or non-certified model-based approaches