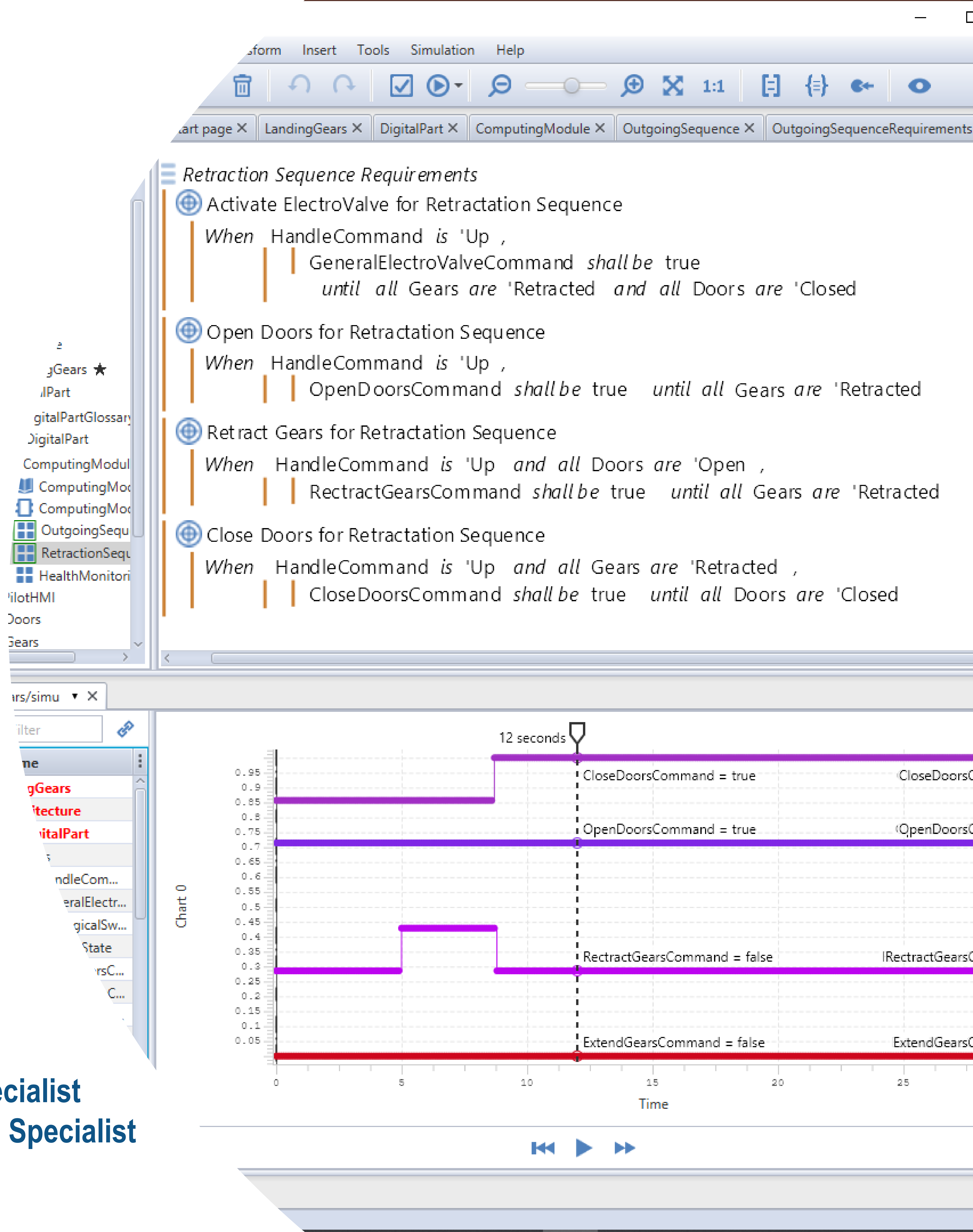
3DEXPERIENCE®

EXECUTABLE REQUIREMENTS AS THE CONDUCTOR OF SYSTEMS ENGINEERING



DASSAULT SYSTEMES | The 3DEXPERIENCE® Company

David Diaz – WW CATIA Systems, E/E and Software Engineering, Industry Process Specialist
Arshita Paliwal – WW CATIA Systems, E/E and Software Engineering, Industry Process Specialist

AGENDA

Use Case Description

End-To-End Process Scenario

Executable Requirements with RIL

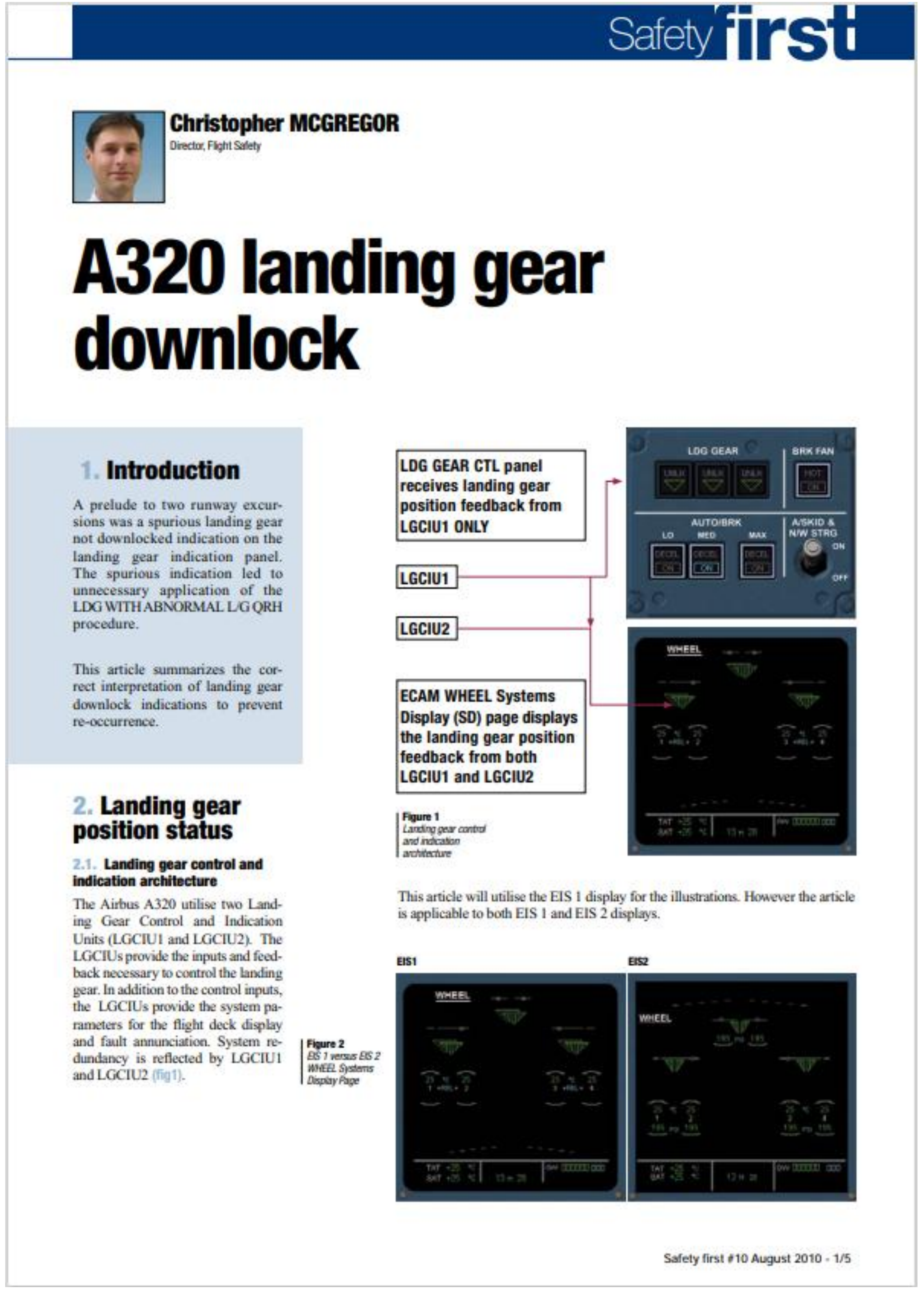
Software-In-The-Loop (SIL)

Hardware-In-The-Loop (HIL)

Conclusion

USE CASE DESCRIPTION- LANDING GEAR CASE STUDY

"A320 landing gear downlock- *Christopher GREGOR Director, Flight Safety*"



Safety first

Christopher MCGREGOR
Director, Flight Safety

A320 landing gear downlock

1. Introduction

A prelude to two runway excursions was a spurious landing gear not downlocked indication on the landing gear indication panel. The spurious indication led to unnecessary application of the LDG WITH ABNORMAL L/G QRH procedure.

This article summarizes the correct interpretation of landing gear downlock indications to prevent re-occurrence.

2. Landing gear position status

2.1. Landing gear control and indication architecture

The Airbus A320 utilise two Landing Gear Control and Indication Units (LGCIU1 and LGCIU2). The LGCIUs provide the inputs and feedback necessary to control the landing gear. In addition to the control inputs, the LGCIUs provide the system parameters for the flight deck display and fault annunciation. System redundancy is reflected by LGCIU1 and LGCIU2 (fig1).

LDG GEAR CTL panel receives landing gear position feedback from LGCIU1 ONLY

LGCIU1

LGCIU2

ECAM WHEEL Systems Display (SD) page displays the landing gear position feedback from both LGCIU1 and LGCIU2

Figure 1
Landing gear control and indication architecture

This article will utilise the EIS 1 display for the illustrations. However the article is applicable to both EIS 1 and EIS 2 displays.

EIS1 EIS2

Figure 2
EIS 1 versus EIS 2 WHEEL Systems Display Page

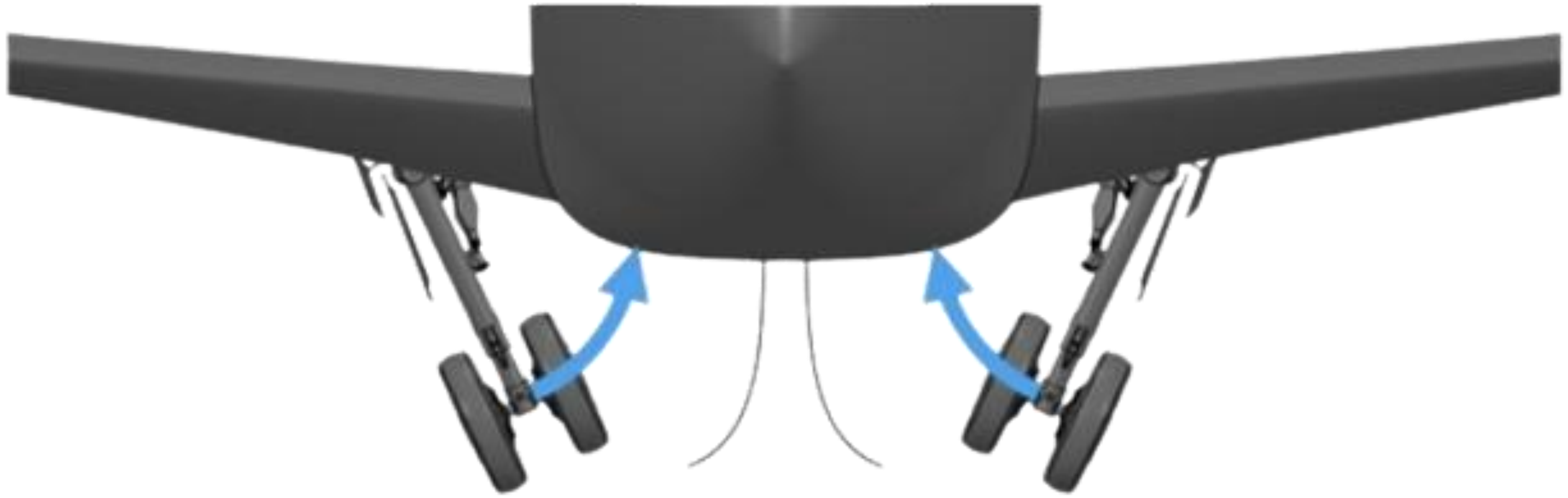
Safety first #10 August 2010 - 1/5

Airbus A320 Landing gear using 1 Locked sensor

Spurious landing gear not downlocked indications with one sensor

- *Need for a redundant sensor for more accuracy*

Aircraft system architecture update to incorporate 2 locking sensors for deployment and retraction



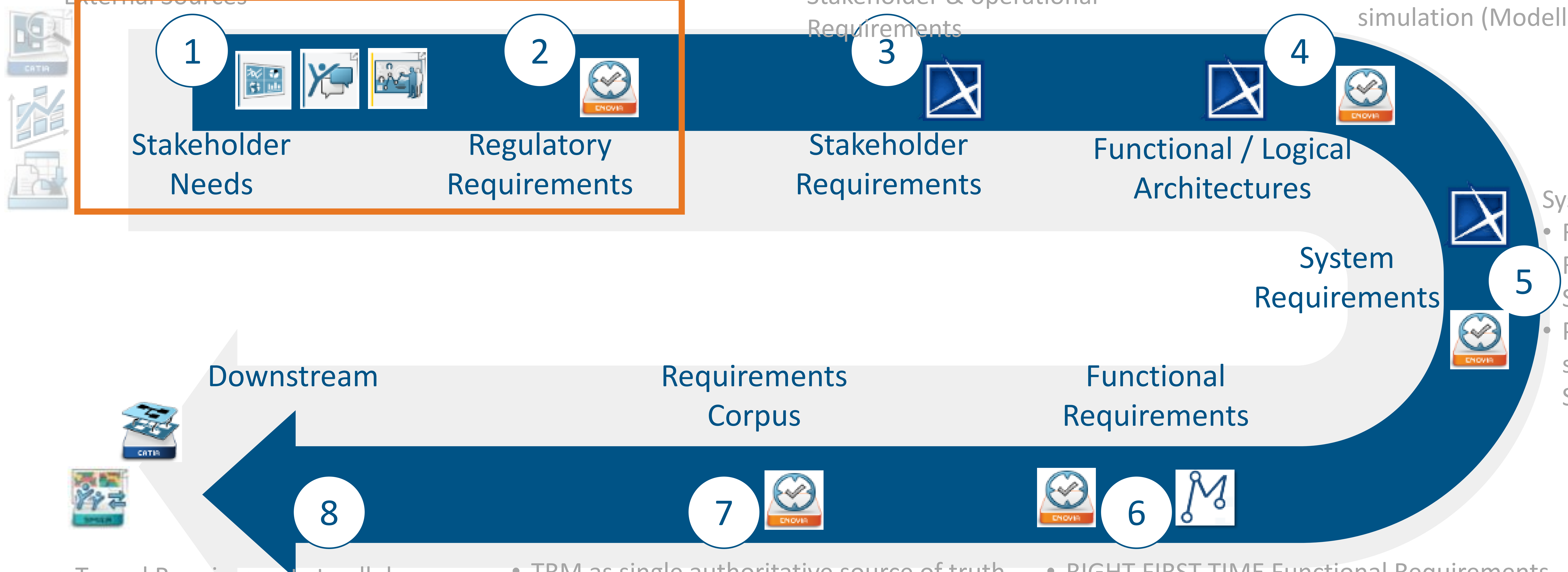
WORKFLOW

- Elicit Stakeholder Needs
- Market-driven Needs
- Capture & reference Ideas
- External Sources

- Consider regulatory requirements impacts (Safety, Environmental …)

Operational Requirements
- Mission modelling & simulation
- Decompose Needs to Stakeholder & operational Requirements

- Activity Diagrams
- Internal block diagrams describing architectures
- State machines
- Requirements Diagrams
- Conceptual behaviour modelling & simulation (Modellica / SIMULINK,…)



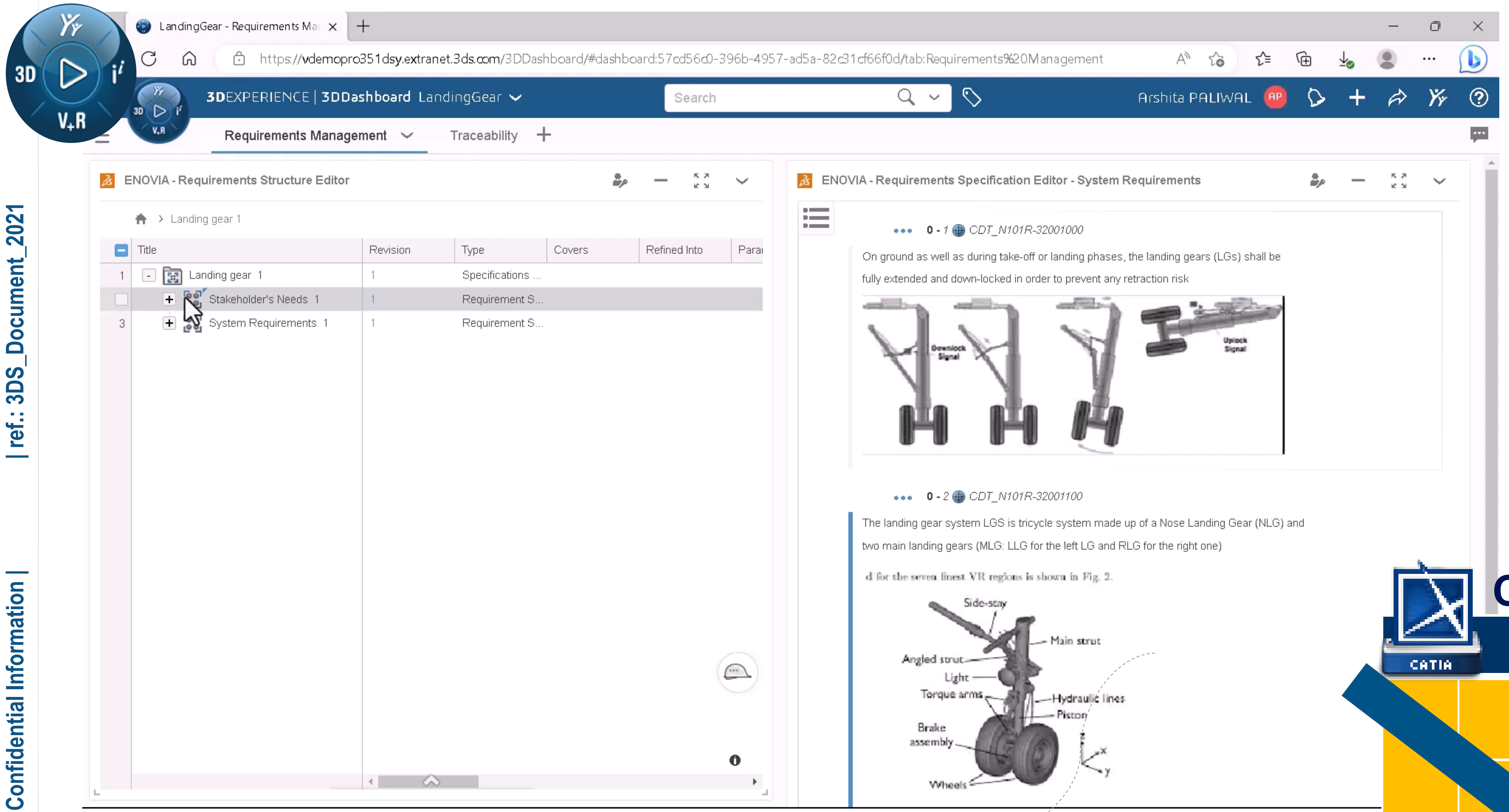
System Requirements
- Refine Stakeholder Requirements into System / Sub-System Requirements.
- Physics behaviours modelling & simulation (Modellica / SIMULINK,…)

- Traced Requirements to all down stream disciplines

- TRM as single authoritative source of truth for all requirements
- Enable downstream disciplines to trace, reference, validate against requirements for all activities.
- Manage test strategy, cases, executions against requirements.

- RIGHT FIRST TIME Functional Requirements authored in STIMULUS
- Simulate for conflicting, ambiguous, missing requirements.
- Push robust requirements back to TRM

PROCESS FLOW-> 3DX- SYSTEM SPECIFICATION



Export of Requirements from Enovia Requirements Management (TRM) to CATIA Magic via DataHub plugin

CATIA Magic

Domain			Pillar				
			Requirements	Structure	Behavior	Parameters	Safety & Reliability
Domain	Problem	Black Box	Stakeholder Needs	System Context	Use Cases	Measures of Effectiveness	Conceptual and Functional FMEA
		White Box		Conceptual Subsystems	Functional Analysis	MoEs for Subsystems	Conceptual Subsystems FMEA
	Solution		System Requirements	System Structure	System Behavior	System Parameters	System S&R
			Subsystem Requirements	Subsystem Structure	Subsyste ı Behavior	Subsystem Parameters	Subsystem S&R
			Component Requirements	Component Structure	Component Behavior	Component Parameters	Component S&R
	Implementation						

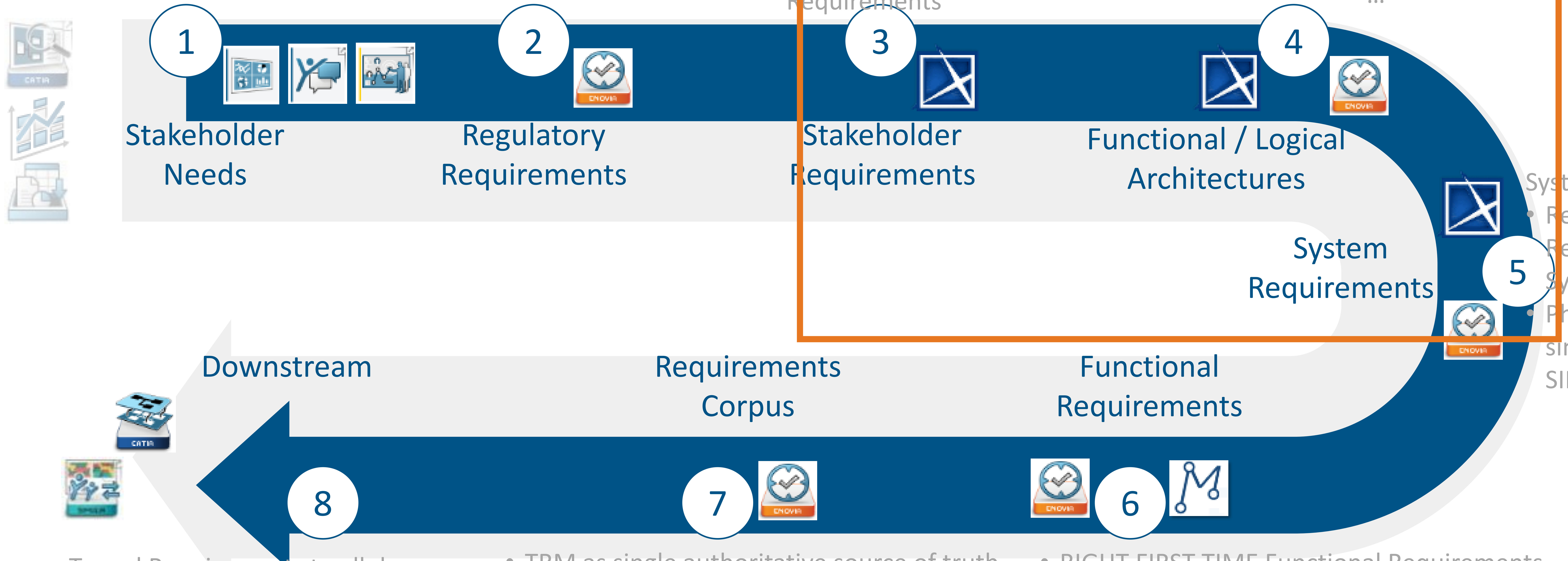
WORKFLOW

- Elicit Stakeholder Needs
- Market-driven Needs
- Capture & reference Ideas
- External Sources

- Consider regulatory requirements impacts (Safety, Environmental …)

Operational Requirements
- Mission modelling & simulation
- Decompose Needs to Stakeholder & operational Requirements

- Activity Diagrams
- Internal block diagrams describing architectures
- State machines
- Requirements Diagrams
- Conceptual behaviour modelling & simulation (Modellica / SIMULINK,…)
- …



System Requirements
- Refine Stakeholder Requirements into System / Sub-System Requirements.
- Physics behaviours modelling & simulation (Modellica / SIMULINK,…)

- Traced Requirements to all down stream disciplines

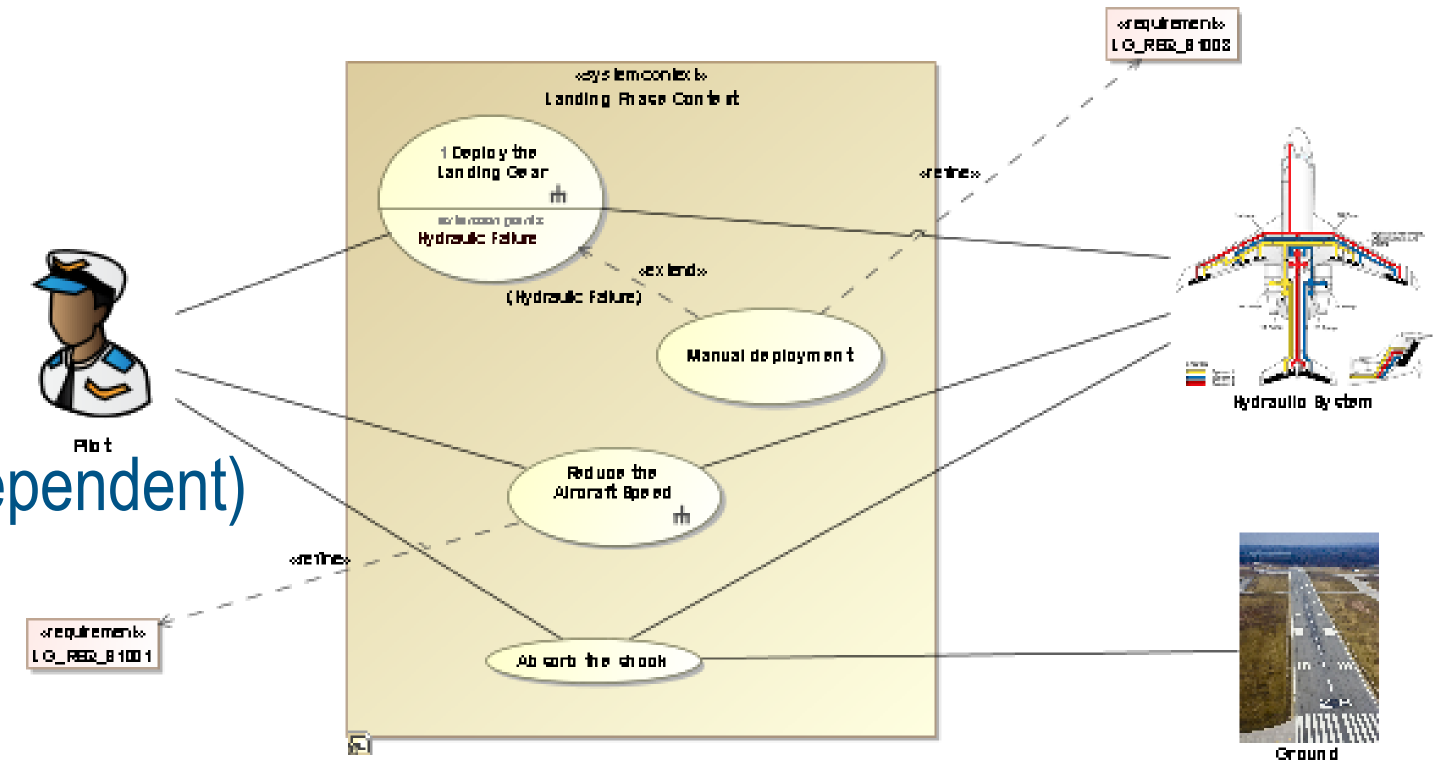
- TRM as single authoritative source of truth for all requirements
- Enable downstream disciplines to trace, reference, validate against requirements for all activities.
- Manage test strategy, cases, executions against requirements.

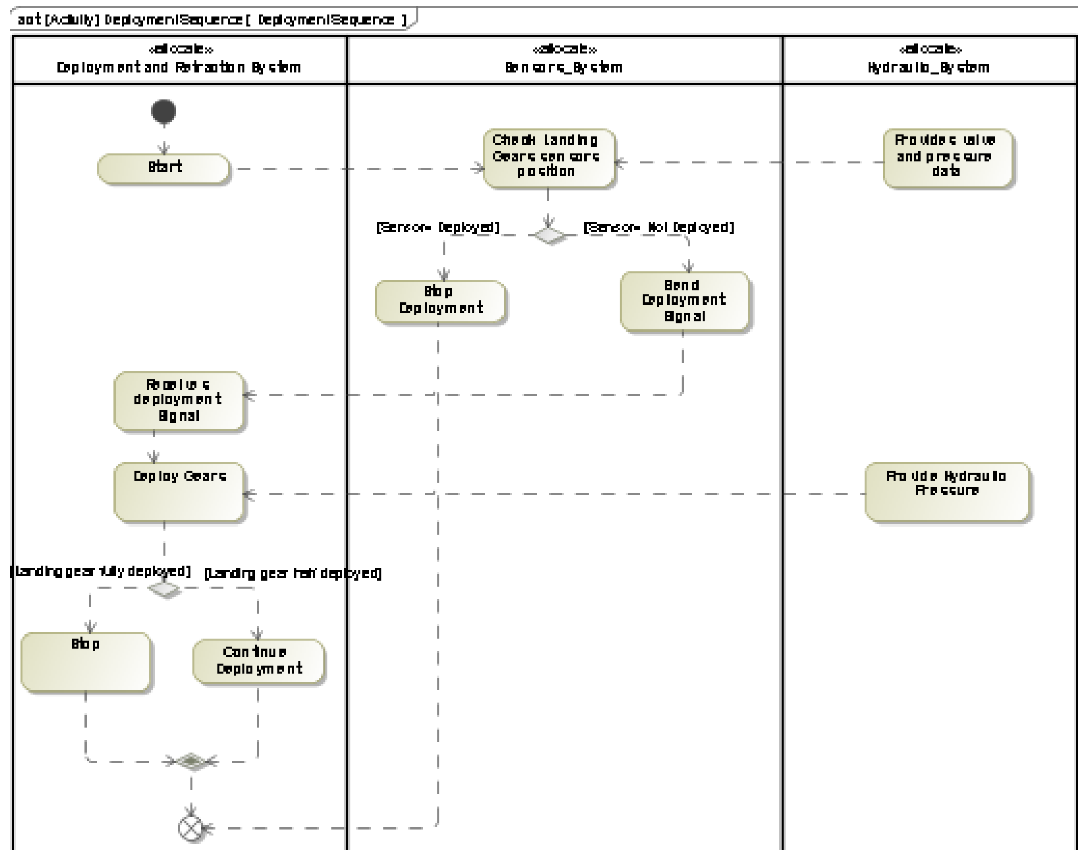
- RIGHT FIRST TIME Functional Requirements authored in STIMULUS
- Simulate for conflicting, ambiguous, missing requirements.
- Push robust requirements back to TRM

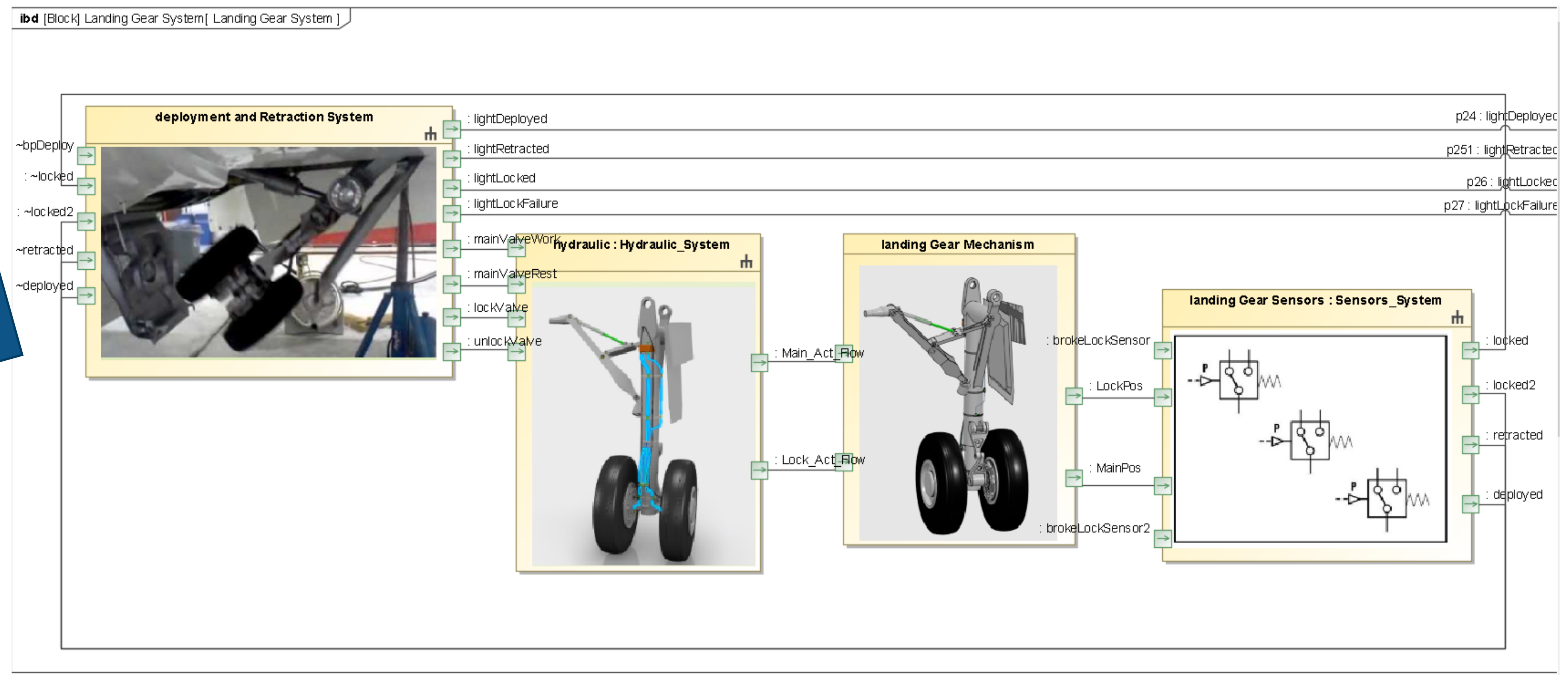
SYSTEM ARCHITECTURE WITH SYSML

- The Cyber MagicGrid methodology separates the problem (implementation free) and the solution (technology dependent)
- It is composed of 4 pillars
- Landing Gear System has been defined using the MagicGrid approach
- Starting from Stakeholder needs down to the subsystem-level architecture.

Domain			Pillar				
			Requirements	Structure	Behavior	Parameters	Safety & Reliability
Domain	Problem	Black Box	Stakeholder Needs	System Context	Use Cases	Measures of Effectiveness	Conceptual and Functional FMEA
Domain	Problem	White Box	Stakeholder Needs	Conceptual Subsystems	Functional Analysis	MoEs for Subsystems	Conceptual Subsystems FMEA
Domain	Solution		System Requirements	System Structure	System Behavior	System Parameters	System S&R
Domain	Solution		Subsystem Requirements	Subsystem Structure	Subsystem Behavior	Subsystem Parameters	Subsystem S&R
Domain	Solution		Component Requirements	Component Structure	Component Behavior	Component Parameters	Component S&R
Domain	Imple-menta-tion						

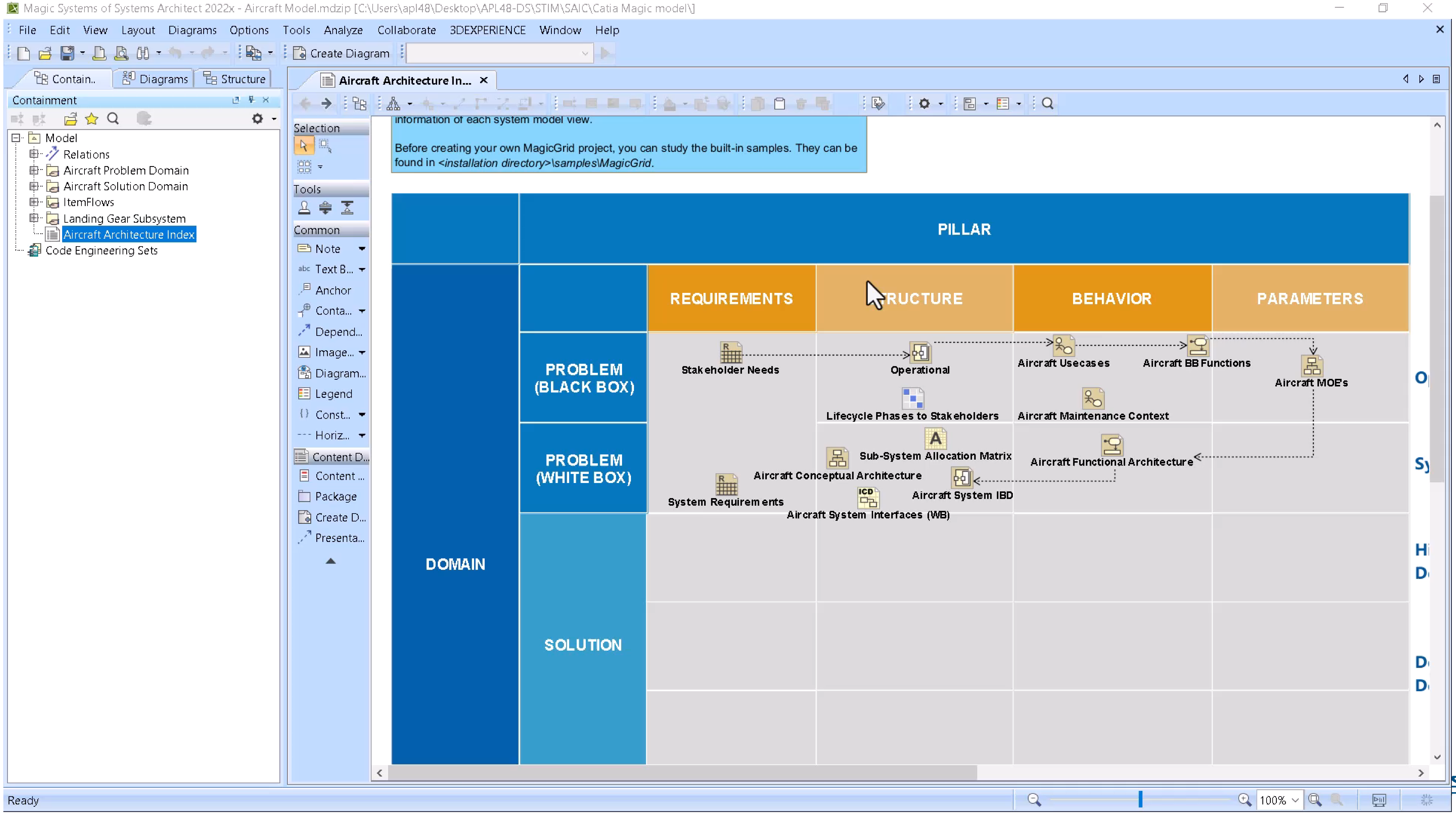






DASSAULT SYSTEMES

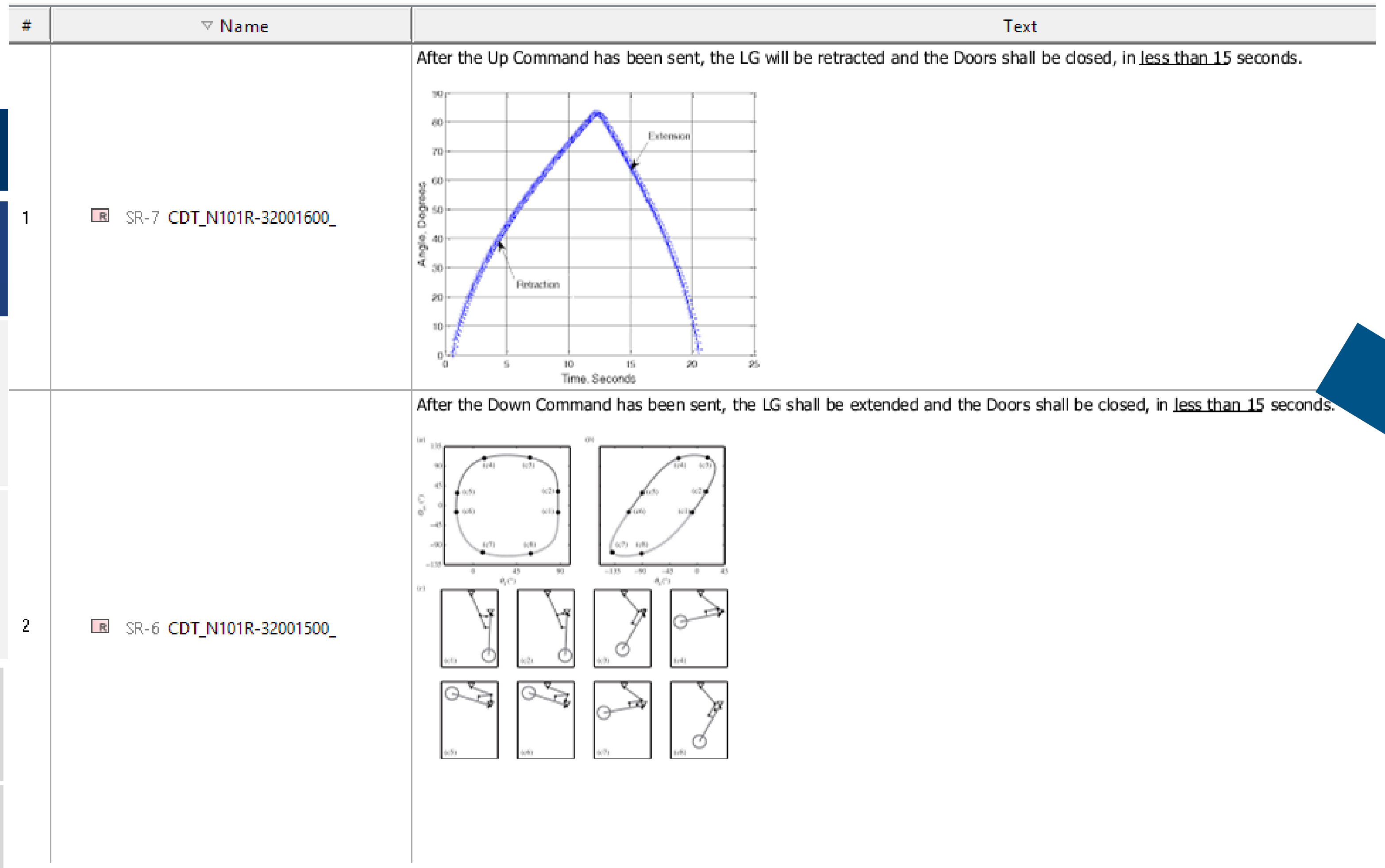
LANDING GEAR SYSTEM MODELLING



PROCESS FLOW-> SYSTEM SPEC- DETAILED SPEC

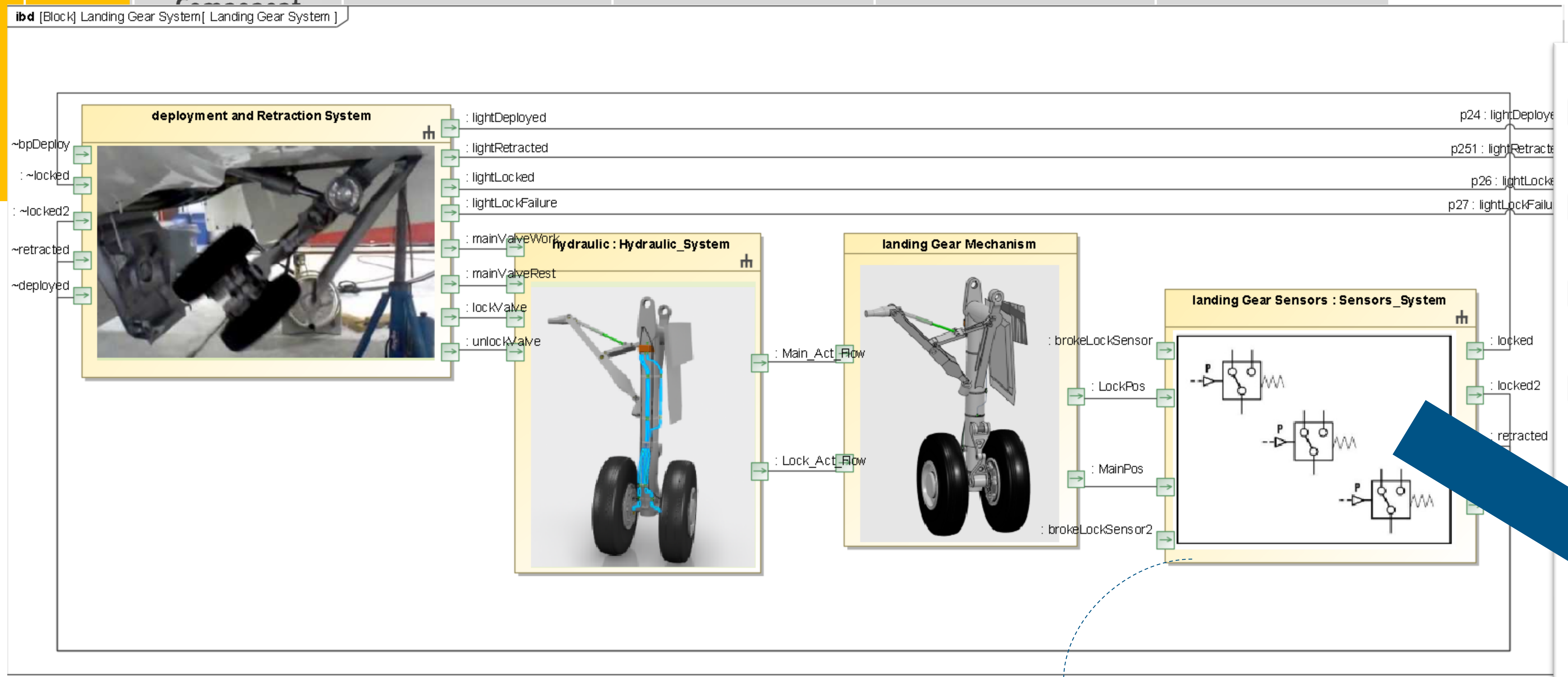
CATIA Magic

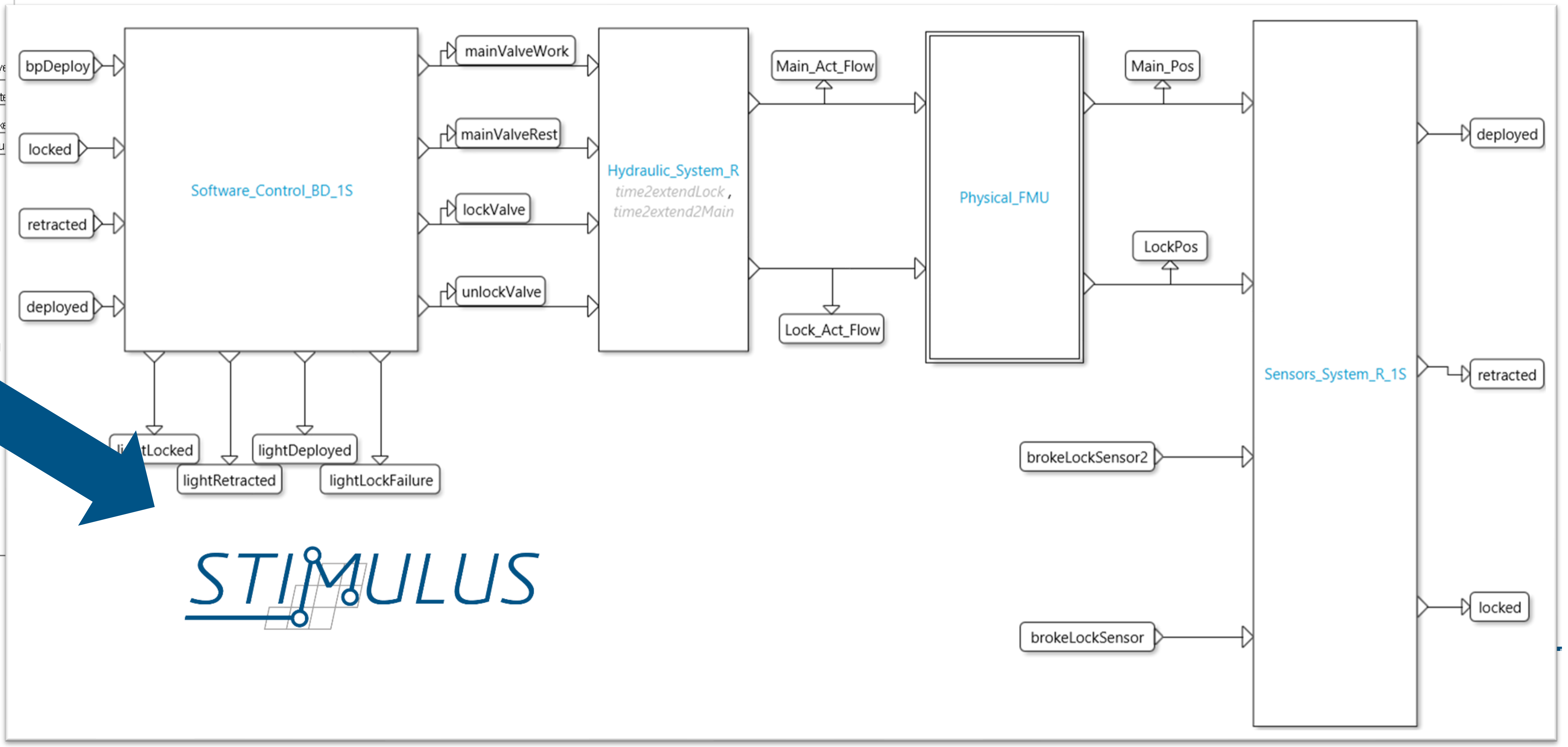
Domain		Pillar				
		Requirements	Structure	Behavior	Parameters	Safety & Reliability
Problem	Black Box	Stakeholder Needs	System Context	Use Cases	Measures of Effectiveness	Conceptual and Functional FMEA
	White Box		Conceptual Subsystems	Functional Analysis	MoEs for Subsystems	Conceptual Subsystems FMEA
Solution		System Requirements	System Structure	System Behavior	System Parameters	System S&R
		Subsystem Requirements	Subsystem Structure	Subsyste ı Behavior	Subsystem Parameters	Subsystem S&R



1

After the Up Command has been sent, the LG will be retracted and the Doors shall be closed, in less than 15 seconds.





Export of Internal block diagram from CATIA Magic to STIMULUS thanks to a plugin

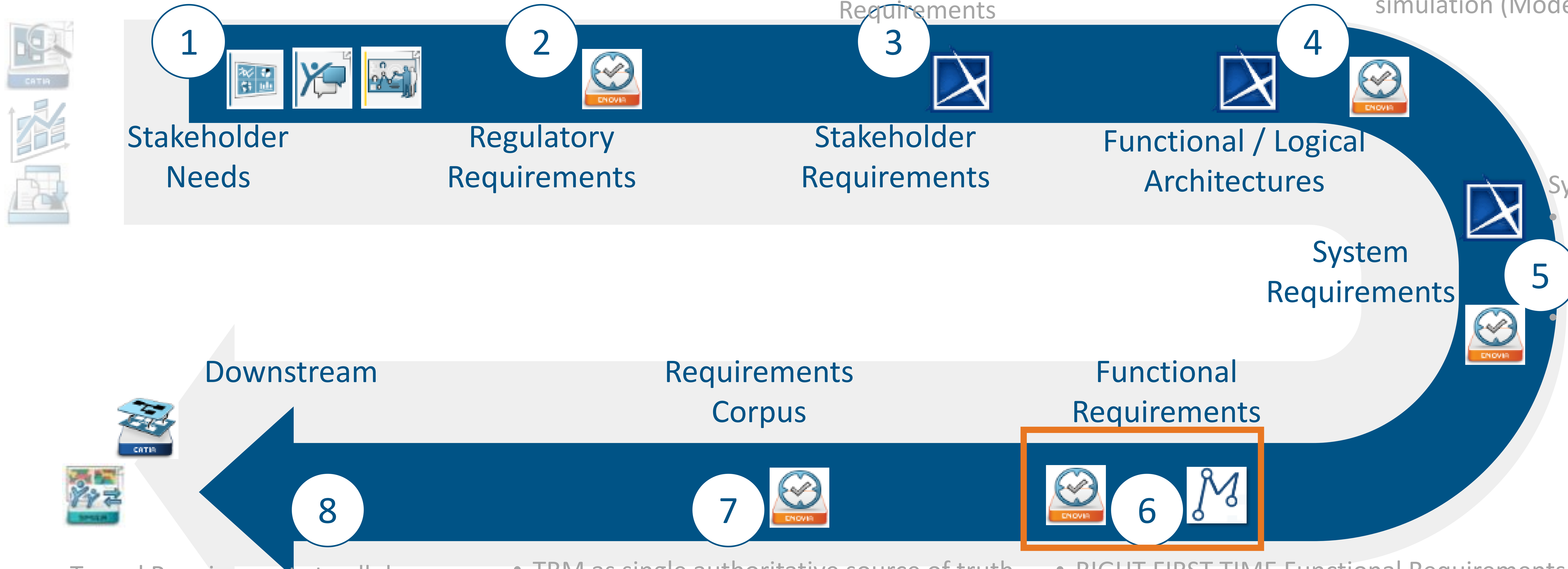
WORKFLOW

- Elicit Stakeholder Needs
- Market-driven Needs
- Capture & reference Ideas
- External Sources

- Consider regulatory requirements impacts (Safety, Environmental …)

Operational Requirements
- Mission modelling & simulation
- Decompose Needs to Stakeholder & operational Requirements

- Activity Diagrams
- Internal block diagrams describing architectures
- State machines
- Requirements Diagrams
- Conceptual behaviour modelling & simulation (Modellica / SIMULINK,…)



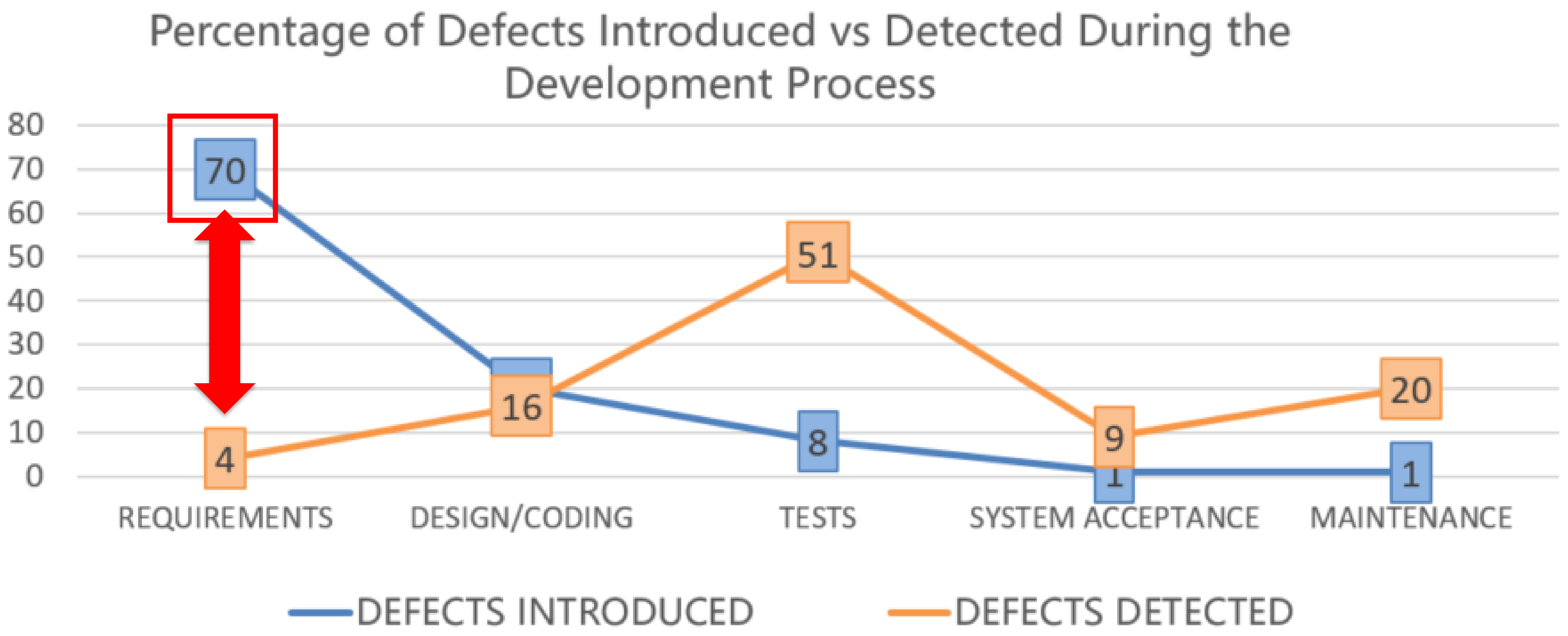
System Requirements
- Refine Stakeholder Requirements into System / Sub-System Requirements.
- Physics behaviours modelling & simulation (Modellica / SIMULINK,…)

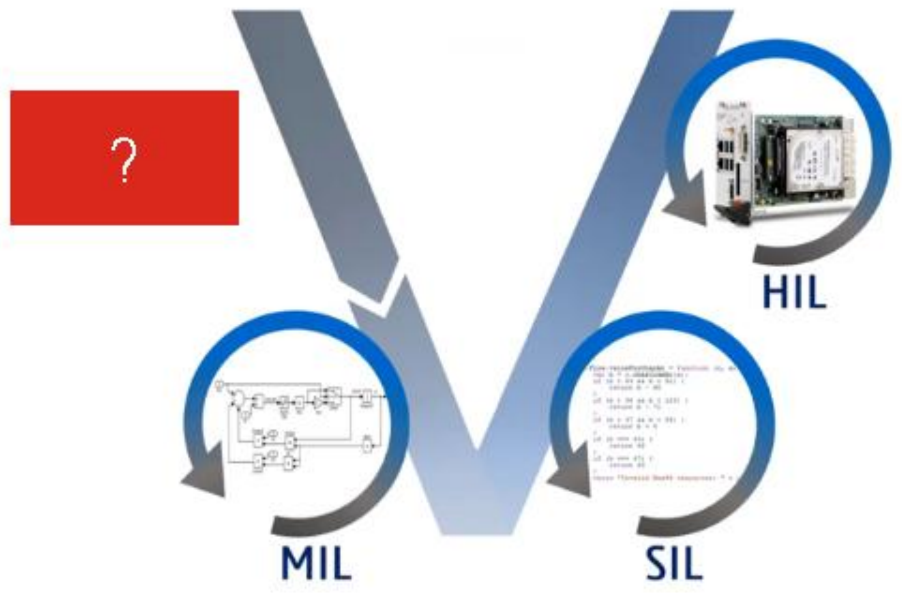
- Traced Requirements to all down stream disciplines

- TRM as single authoritative source of truth for all requirements
- Enable downstream disciplines to trace, reference, validate against requirements for all activities.
- Manage test strategy, cases, executions against requirements.

- RIGHT FIRST TIME Functional Requirements authored in STIMULUS
- Simulate for conflicting, ambiguous, missing requirements.
- Push robust requirements back to TRM

REQUIREMENTS SIMULATION



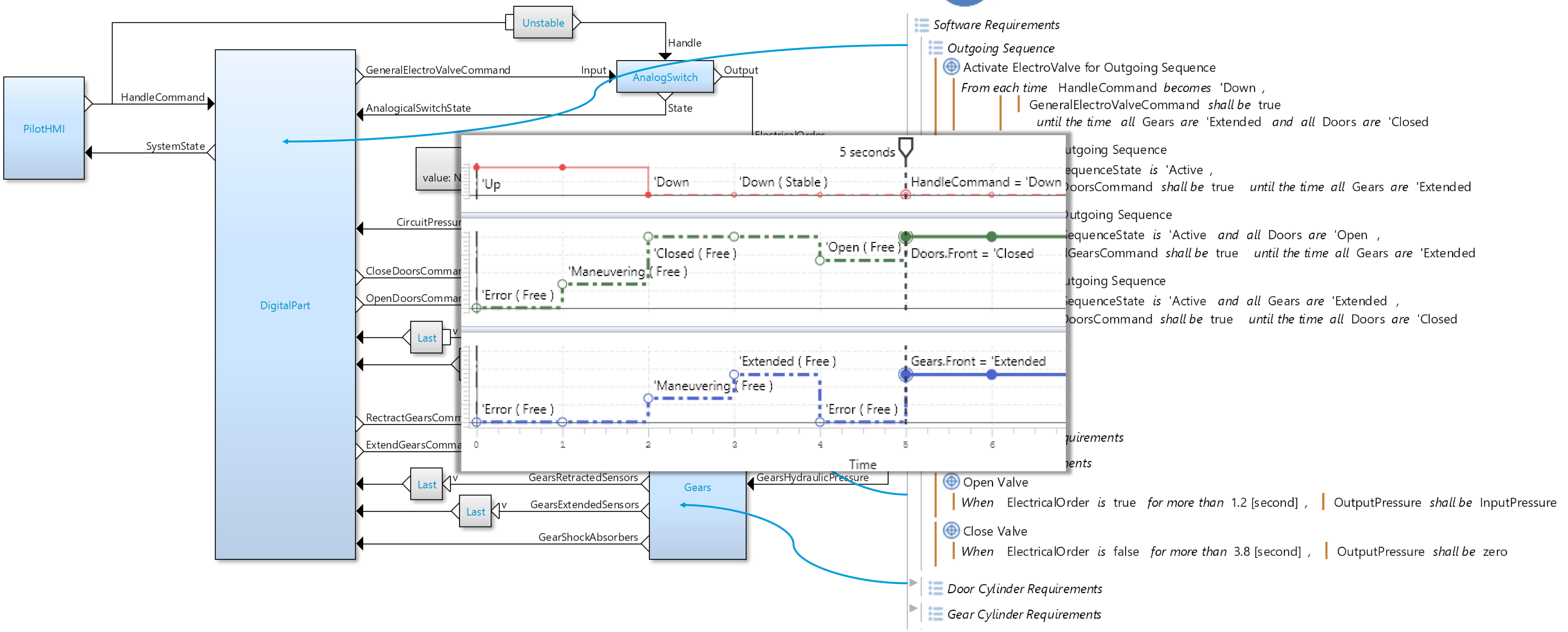


(source: IBM Systems Sciences Institute)

- Quality criteria for functional requirements (what the system shall do):
- Correctness: requirements capture the system behaviour
- Consistency: no conflicting requirements
- Completeness: no missing requirements
- Testability: test scenarios can be written

REQUIREMENTS-IN-THE-LOOP FUNDAMENTALS

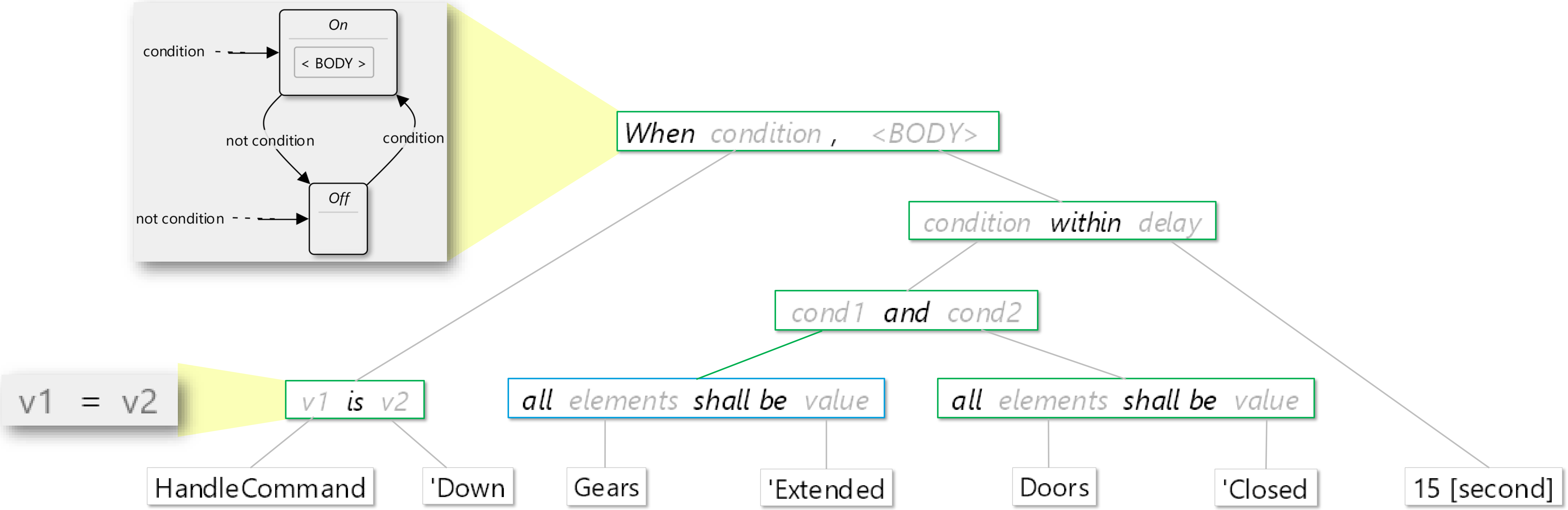
Allocate formalized requirements to SYSTEM architecture … AND SIMULATE!



▶System Architecture ▶Formalized Requirements

FORMALIZING TEXTUAL REQUIREMENTS

Write unambiguous requirements by composing templates that have clear semantics.

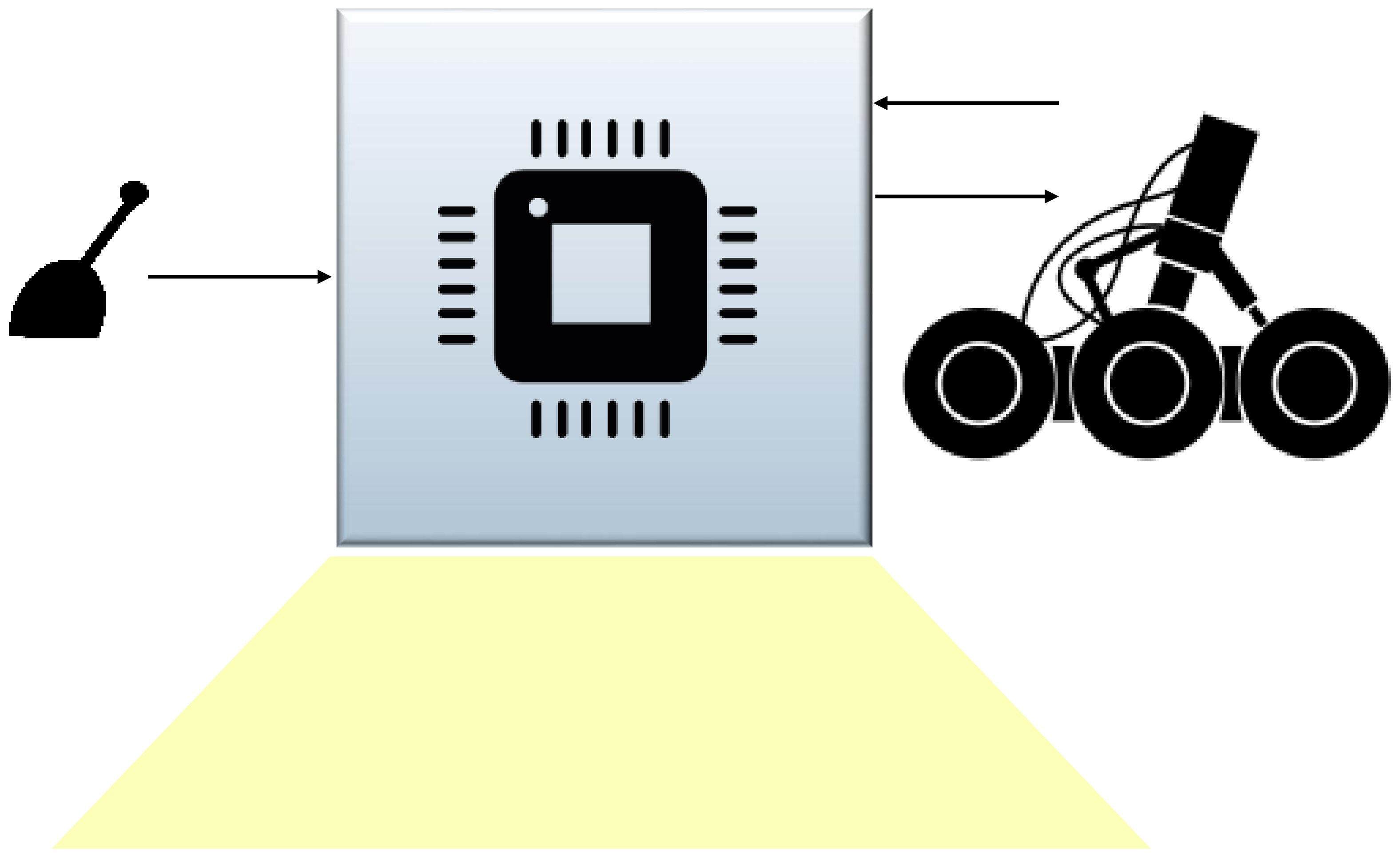


When **HandleCommand** ***is*** **‘Down*****,*** ***all*** **Gears** ***shall be*** **‘Extended** ***and all*** **Doors** ***shall be*** **‘Closed** ***within*** **15 [second]**

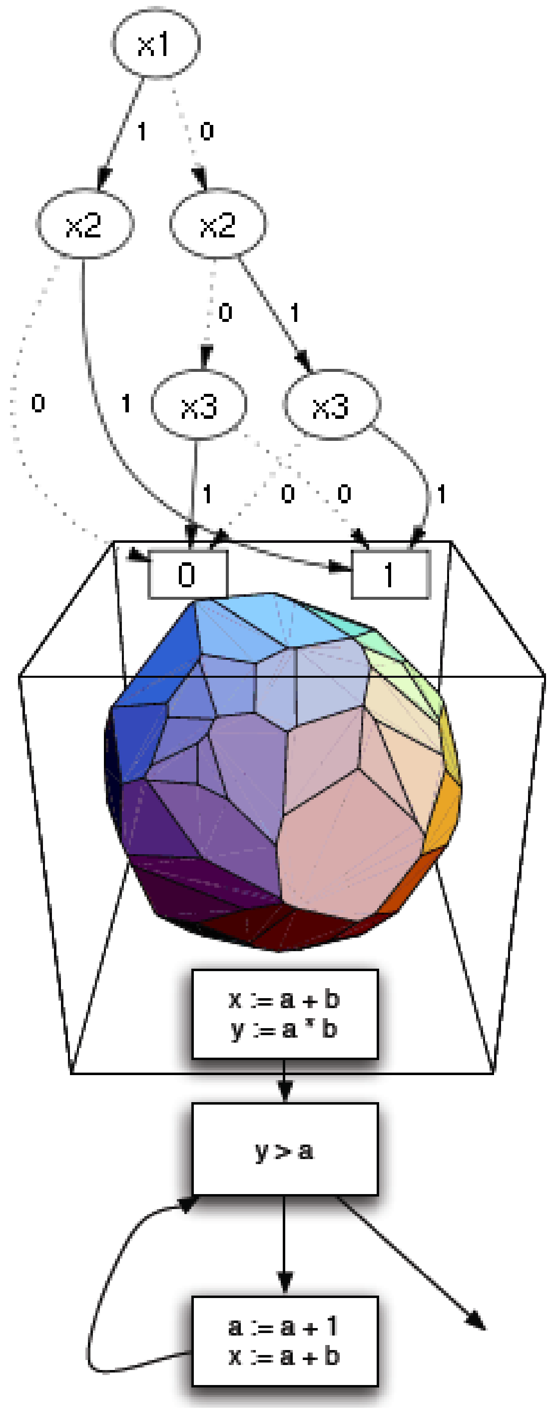
▶Resulting requirements can be simulated, but what does SIMULATION mean?

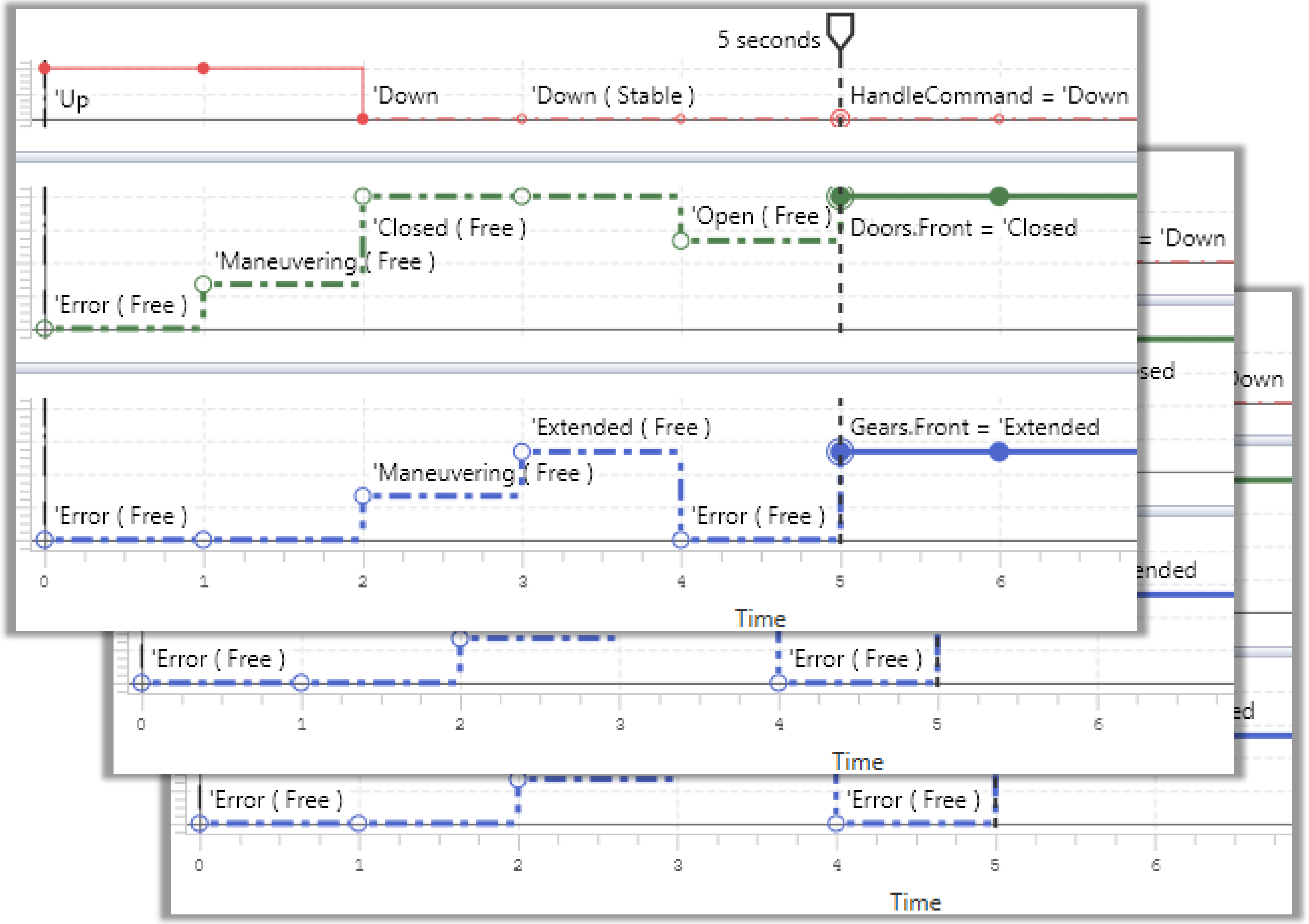
SIMULATE FORMALIZED REQUIREMENTS

Generate possible behaviors that satisfy requirements



When **HandleCommand** ***is*** **'Down,** ***all*** **Gears** ***shall be*** **'Extended** ***and all*** **Doors** ***shall be*** **'Closed** ***within*** **15 [second]**





- ▶Compiler transforms requirements into clocked equations
- ▶Simulator solves logical, numerical and temporal constraints
- ▶Debugger detects incorrect, missing or conflicting requirements

DASSAULT SYSTEMES

LANDING GEAR DIGITAL SUBSYSTEM SPECIFICATION

▶Original informal requirements

Outgoing sequence. The outgoing of gears is decomposed in a sequence of elementary actions. When the gears are locked in retracted position, and the doors are locked in closed position, if the pilot sets the handle to "Down", then the software should have the following sequence of actions:

1. stimulate the general electro-valve isolating the command unit in order to send hydraulic pressure to the maneuvering electro-valves,
2. stimulate the door opening electro-valve,
3. once the three doors are in the open position, stimulate the gear outgoing electro-valve,
4. once the three gears are locked down, stop the stimulation of the gear outgoing electro-valve,
5. stop the stimulation of the door opening electro-valve,
6. stimulate the door closure electro-valve,
7. once the three doors are locked in the closed position, stop the stimulation of the door closure electro-valve,
8. and finally stop stimulating the general electro-valve.

The previous sequences should be interruptible by counter orders (a retraction order occurs during the let down sequence and conversely) at any time. In that case, the scenario continues from the point where it was interrupted.

- ▶Monolithic specification as related requirements cannot be considered independently
- ▶Vague and non testable specification of counter-orders

▶CATIA Stimulus requirements

Outgoing Sequence Requirements

Activate ElectroValve for Outgoing Sequence

When HandleCommand *is* 'Down ,
GeneralElectroValveCommand *shall be* true
until all Gears *are* 'Extended *and all* Doors *are* 'Closed

Open Doors for Outgoing Sequence

When HandleCommand *is* 'Down ,
OpenDoorsCommand *shall be* true *until all* Gears *are* 'Extended

Extend Gears for Outgoing Sequence

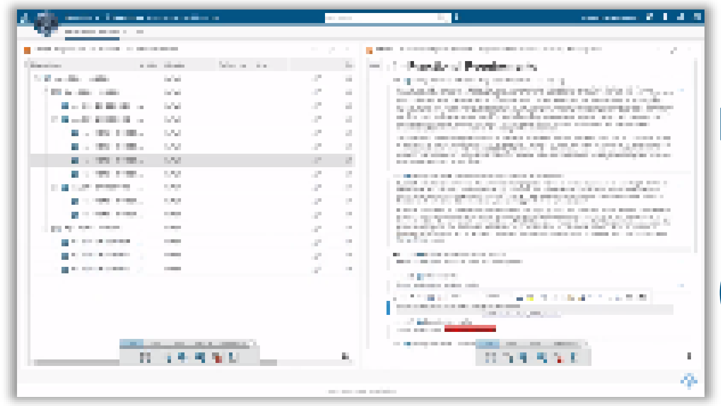
When HandleCommand *is* 'Down *and all* Doors *are* 'Open ,
ExtendGearsCommand *shall be* true *until all* Gears *are* 'Extended

Close Doors for Outgoing Sequence

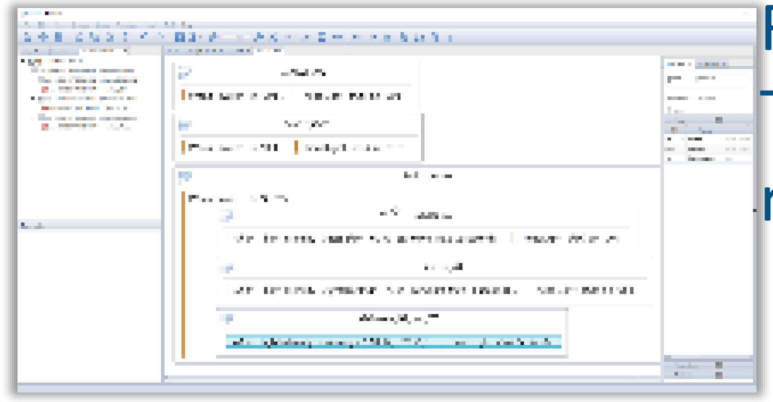
When HandleCommand *is* 'Down *and all* Gears *are* 'Extended ,
CloseDoorsCommand *shall be* true *until all* Doors *are* 'Closed

- ▶Independent requirements which shall always be true whatever the state of the system
- ▶Robust to counter-orders, testable
- ▶Debugged through trials & errors thanks to simulation

REQUIREMENTS ENGINEERING AND VERIFICATION WORKFLOW

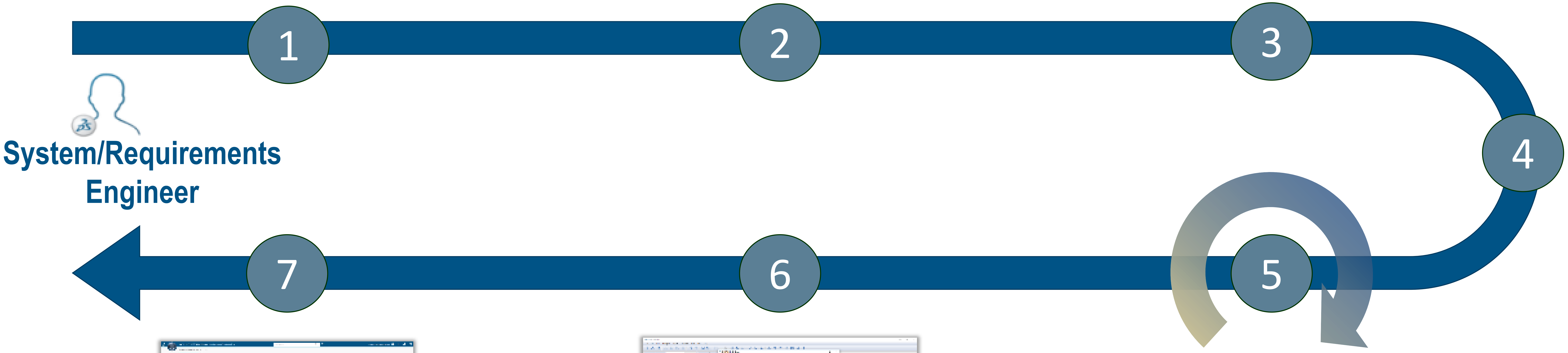


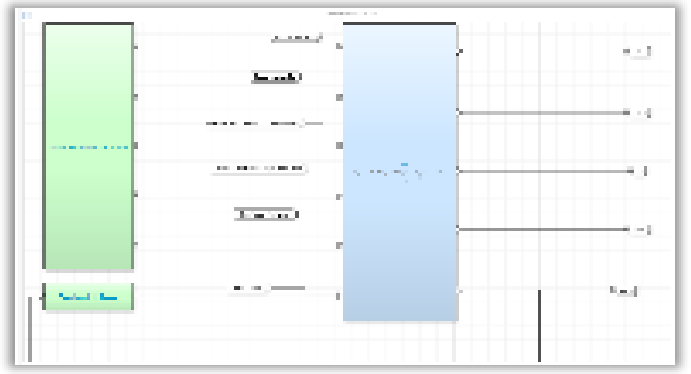
Identify functional requirements inside Requirement Manager (TRM)



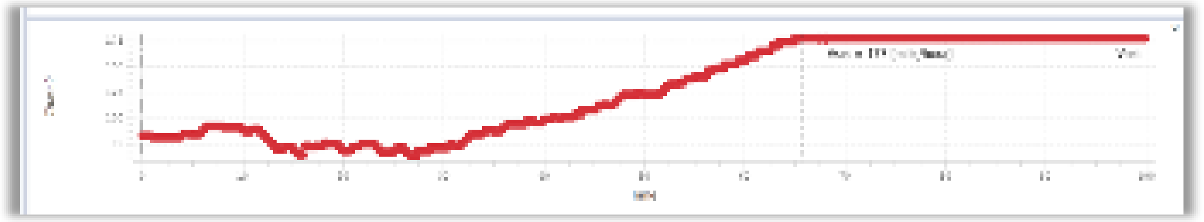
From Stimulus, connect to TRM and pull the text of the requirements into Stimulus

Model in Stimulus modify, add or remove requirements

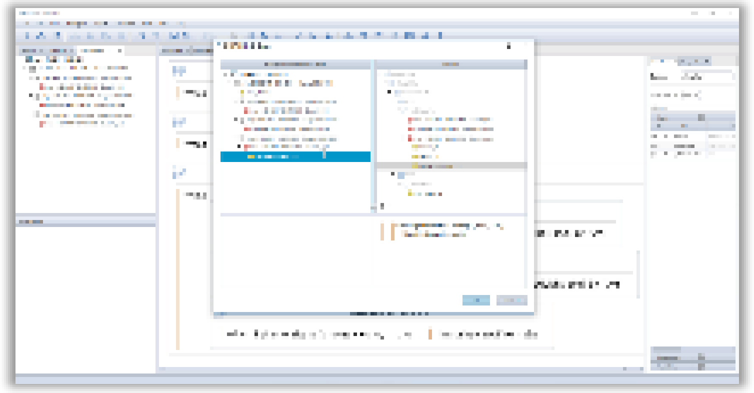




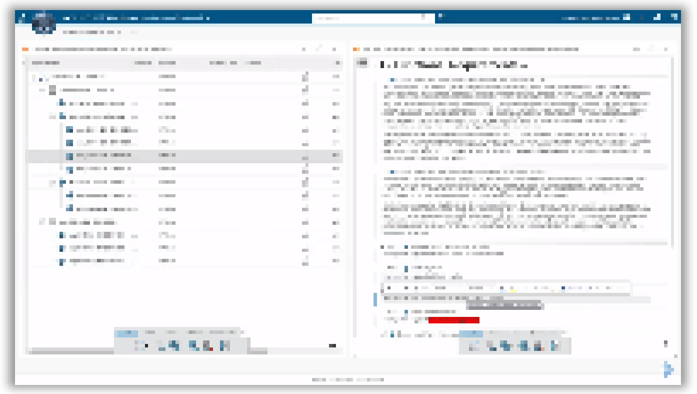
Setup test environment and build test case scenarios



Validate and debug requirements, models and code thanks to simulation



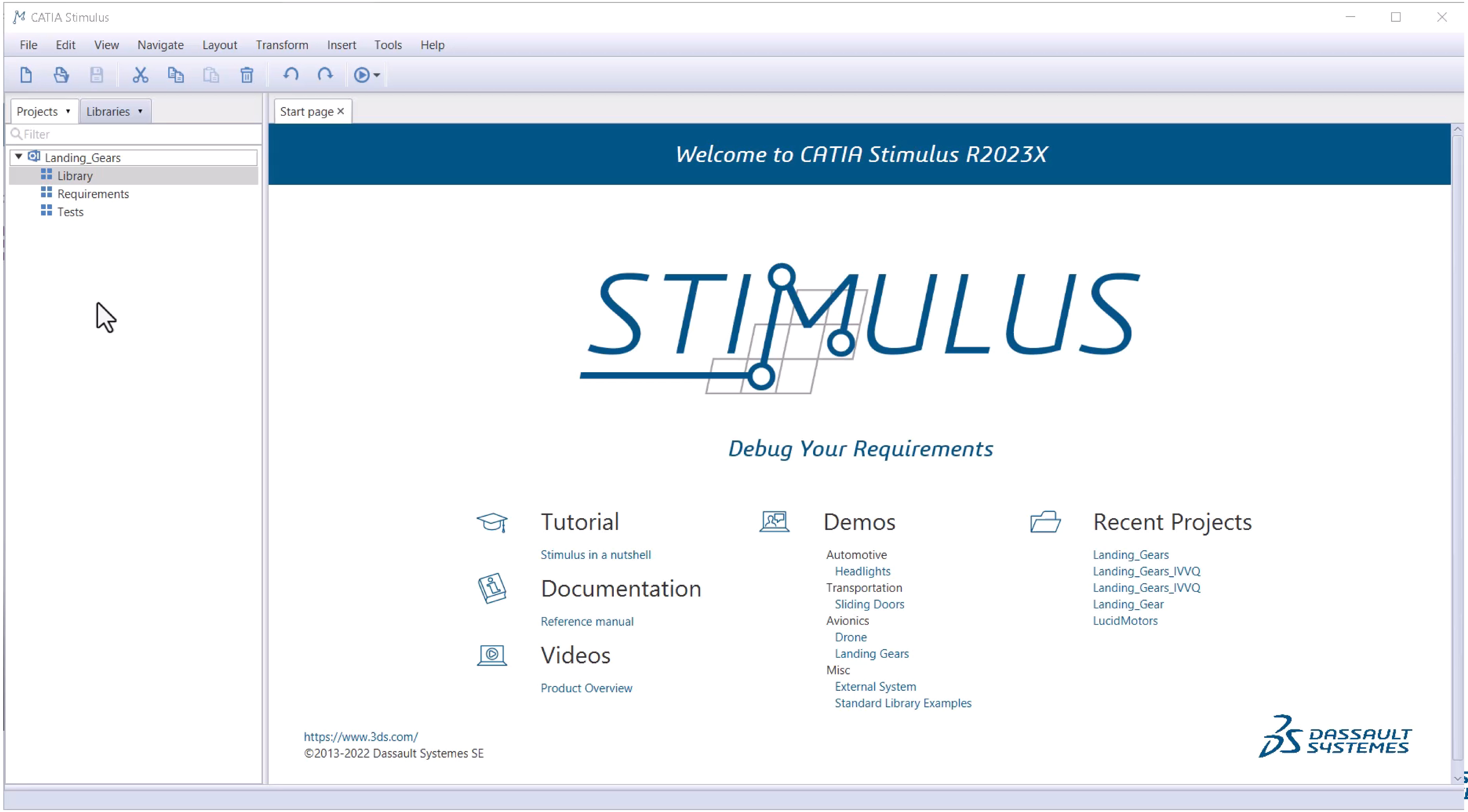
Push Stimulus formalised requirements to TRM. Show unsynchronized requirements in Stimulus.



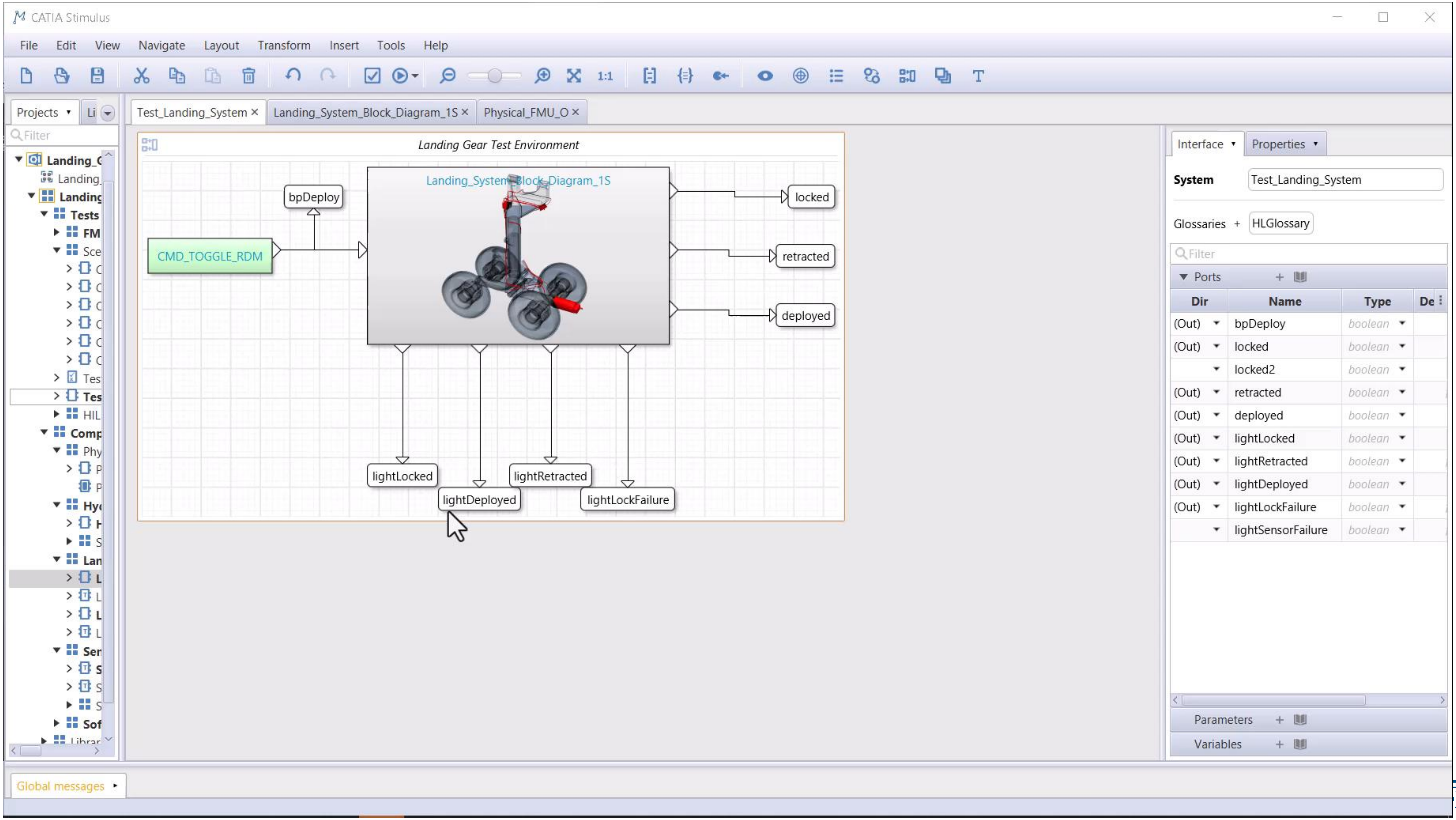
Manage LifeCycle operations inside Requirement Manager (TRM)

DASSAULT SYSTEMES

REQUIREMENTS MODELLING



REQUIREMENTS SIMULATION AND TEST ENVIRONMENT



WORKFLOW

- Elicit Stakeholder Needs
- Market-driven Needs
- Capture & reference Ideas
- External Sources

- Consider regulatory requirements impacts (Safety, Environmental …)

Operational Requirements
- Mission modelling & simulation
- Decompose Needs to Stakeholder & operational Requirements

- Activity Diagrams
- Internal block diagrams describing architectures
- State machines
- Requirements Diagrams
- Conceptual behaviour modelling & simulation (Modellica / SIMULINK,…)

System Requirements
- Refine Stakeholder Requirements into System / Sub-System Requirements.
- Physics behaviours modelling & simulation (Modellica / SIMULINK,…)

- Traced Requirements to all down stream disciplines

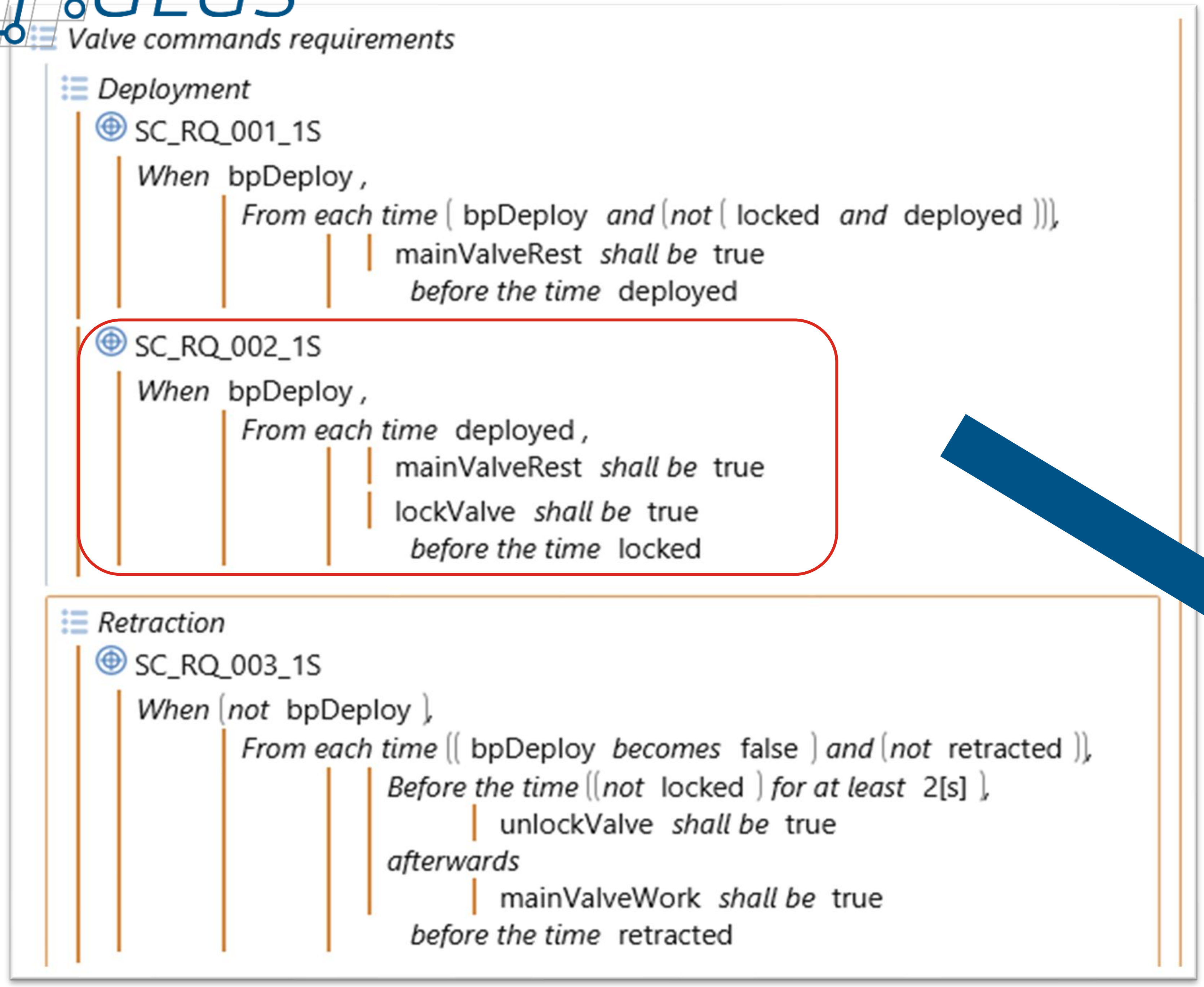
- TRM as single authoritative source of truth for all requirements
- Enable downstream disciplines to trace, reference, validate against requirements for all activities.
- Manage test strategy, cases, executions against requirements.

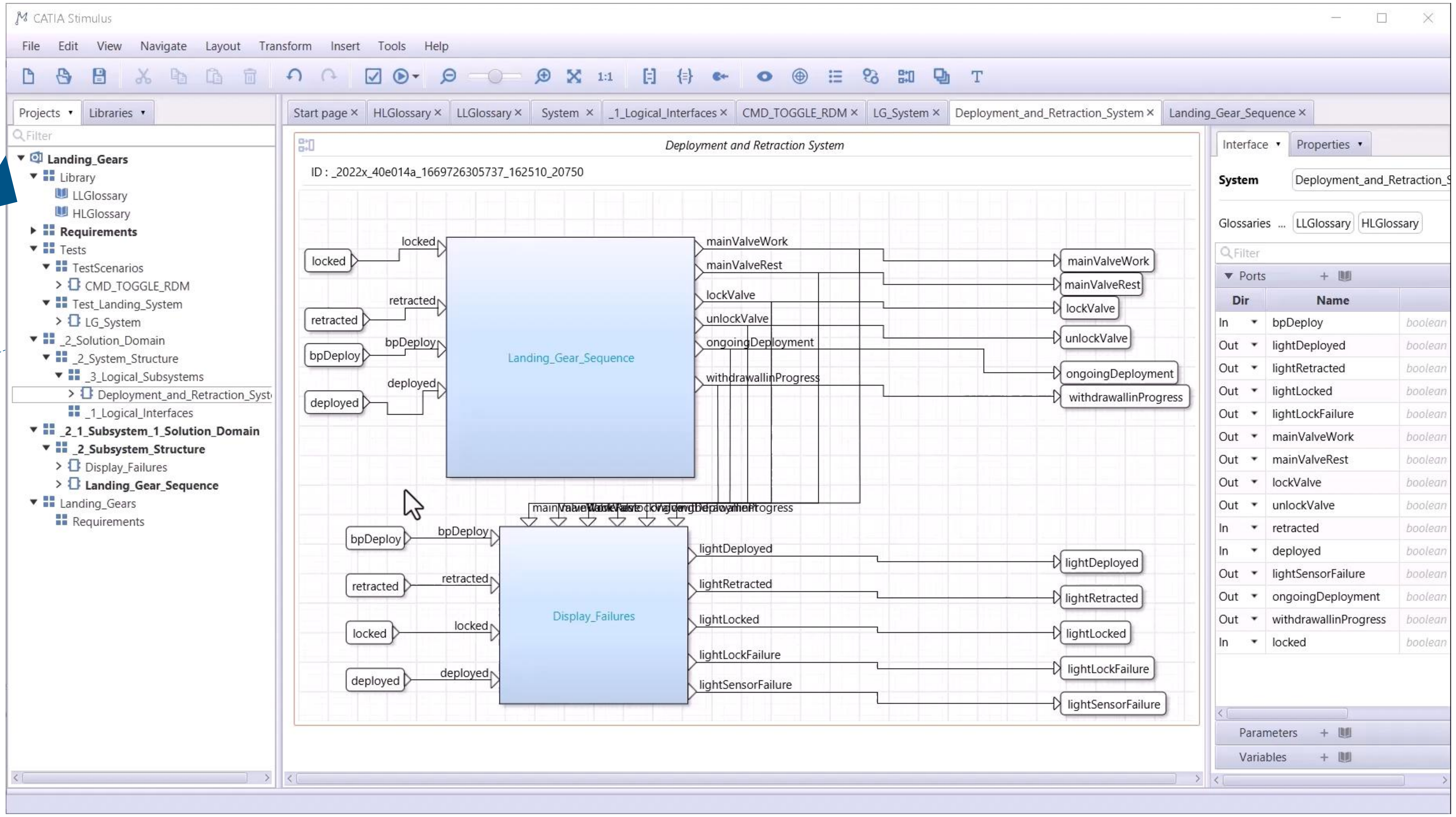
- RIGHT FIRST TIME Functional Requirements authored in STIMULUS
- Simulate for conflicting, ambiguous, missing requirements.
- Push robust requirements back to TRM

PROCESS FLOW-> FUNCTIONAL REQUIREMENTS- 3DX

VIDEO

STIMULUS





Export of validated Stimulus requirements to the 3DExperience platform through TRM

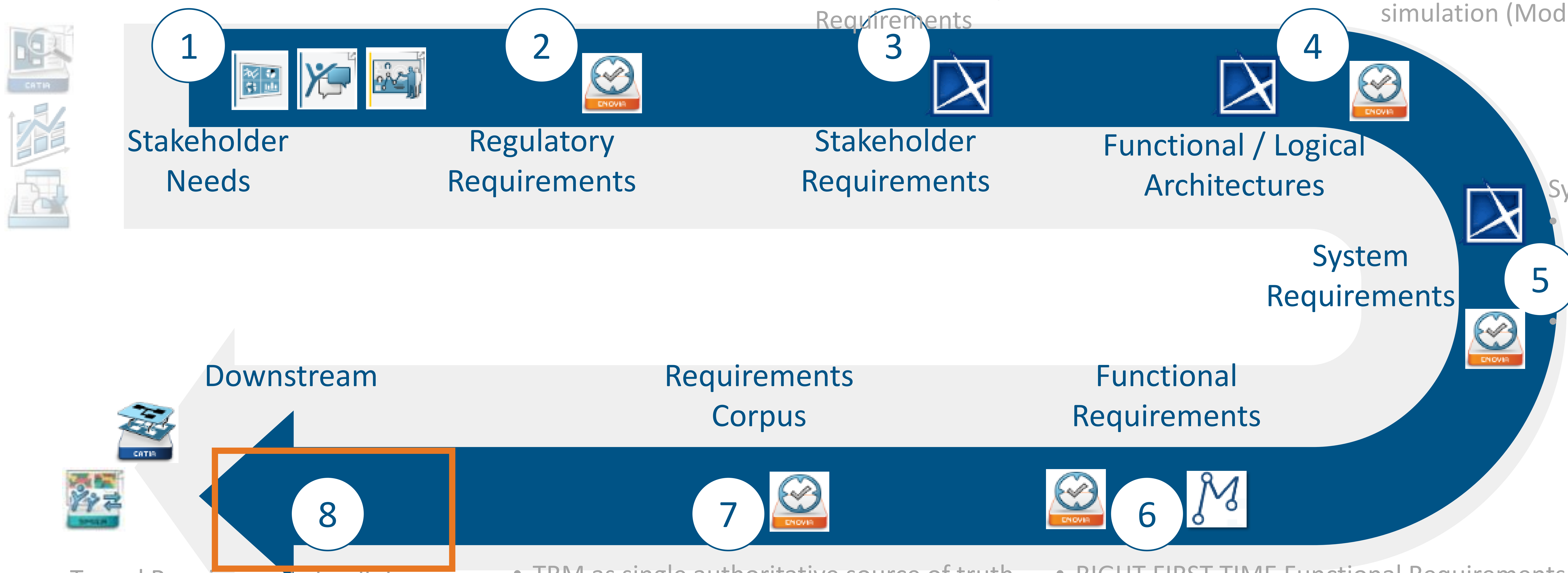
WORKFLOW

- Elicit Stakeholder Needs
- Market-driven Needs
- Capture & reference Ideas
- External Sources

- Consider regulatory requirements impacts (Safety, Environmental …)

Operational Requirements
- Mission modelling & simulation
- Decompose Needs to Stakeholder & operational Requirements

- Activity Diagrams
- Internal block diagrams describing architectures
- State machines
- Requirements Diagrams
- Conceptual behaviour modelling & simulation (Modellica / SIMULINK,…)



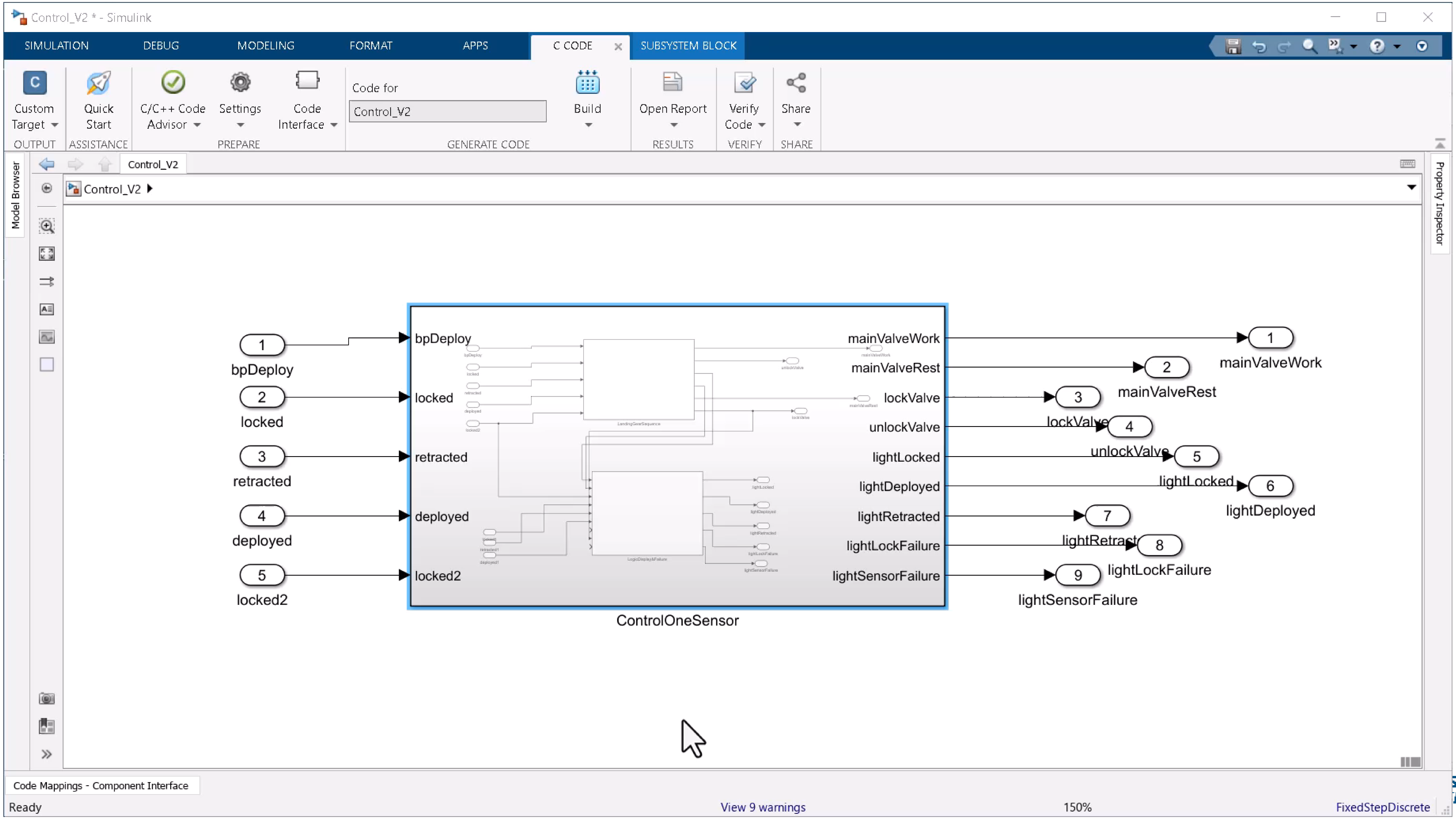
System Requirements
- Refine Stakeholder Requirements into System / Sub-System Requirements.
- Physics behaviours modelling & simulation (Modellica / SIMULINK,…)

- Traced Requirements to all down stream disciplines

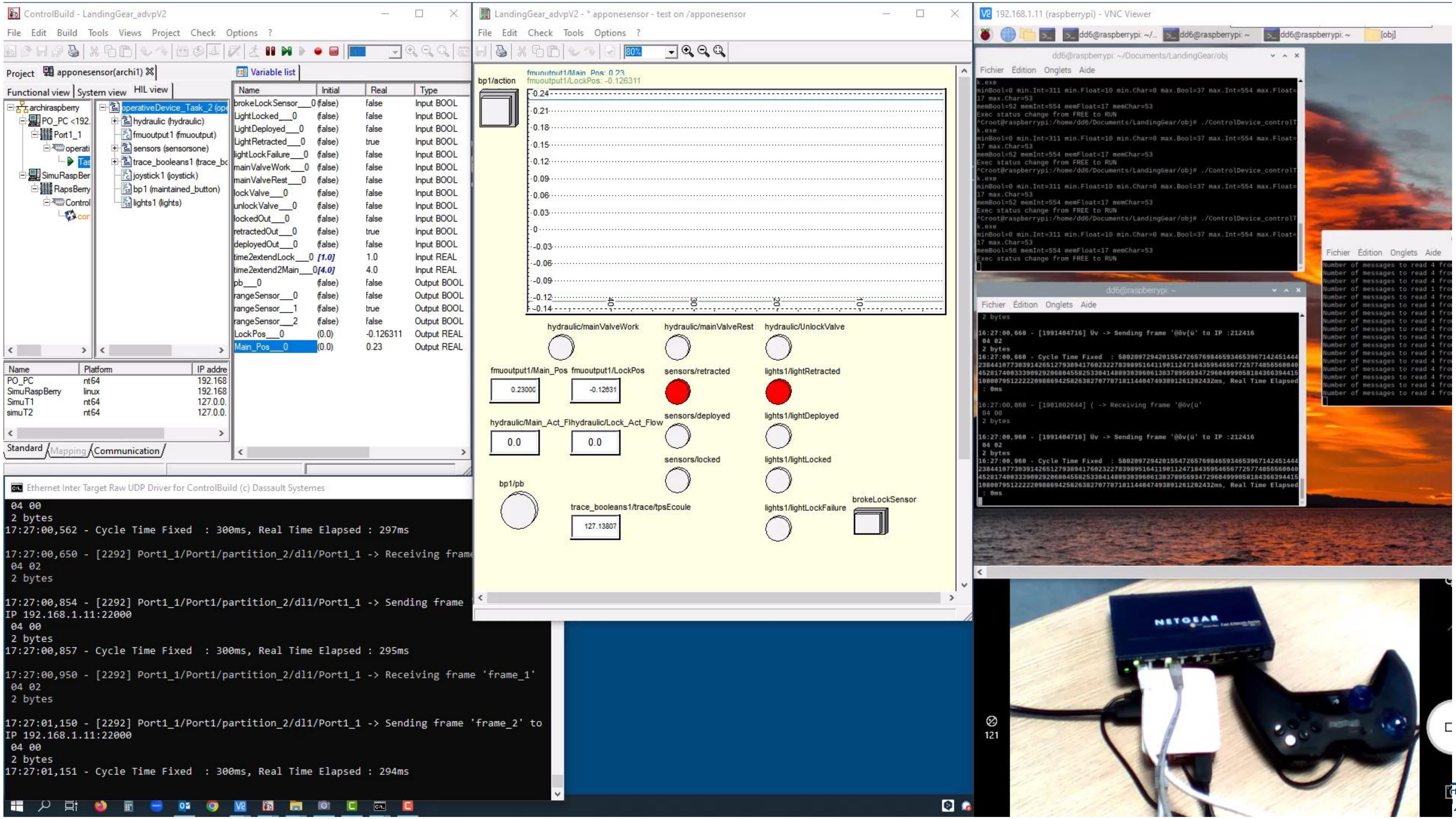
- TRM as single authoritative source of truth for all requirements
- Enable downstream disciplines to trace, reference, validate against requirements for all activities.
- Manage test strategy, cases, executions against requirements.

- RIGHT FIRST TIME Functional Requirements authored in STIMULUS
- Simulate for conflicting, ambiguous, missing requirements.
- Push robust requirements back to TRM

SOFTWARE IN THE LOOP SIMULATION



REAL TIME SIMULATION TO TEST THE CODE



DASSAULT SYSTEMES

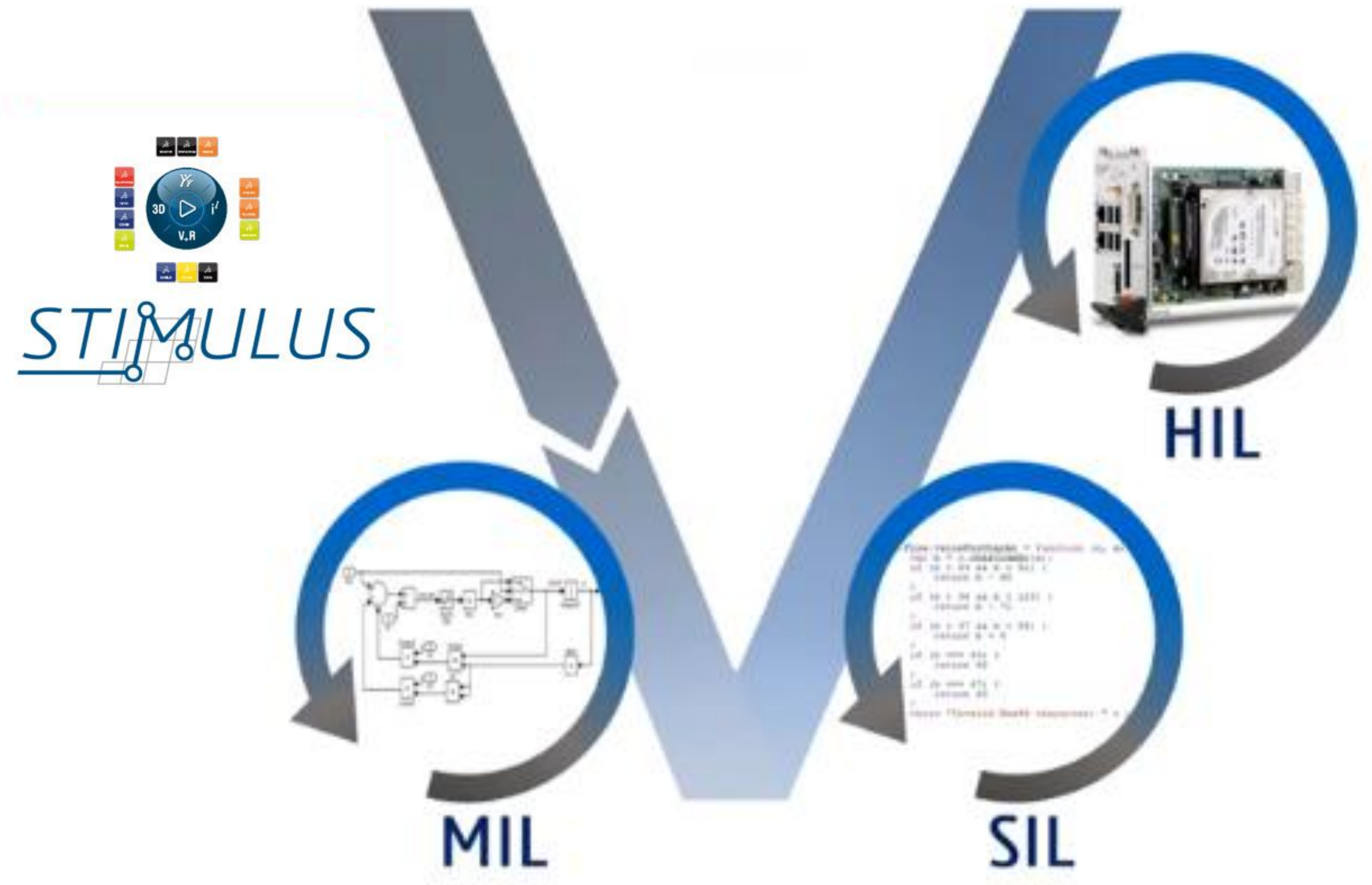
CONCLUSIONS

Quality Criteria

- Ensuring the quality and reliability of complex systems
- Automate functional tests in a completely innovative manner, leading to higher quality products and smaller maintenance costs.

Simulate and validate functional specifications

- Simulate requirements to detect incorrect, missing, conflicting requirements before design begins
- Requirements can be defined and simulated incrementally to ease the elicitation process



Validate design against specifications

- External code can be imported, allowing the compliance of the system implementation to be checked against its requirements;
- Simulate your design against requirements to check that the system actually complies with its specification

Automate hardware testing analysis

- Automate analysis of the HIL test log files allowing much stronger and faster verifications

Arshita.paliwal@3ds.com

David.diaz@3ds.com

Thankyou