How do Model-Based Best Practices enhance Requirements Engineering

Gérard Morin, ESTEREL Technologies
3ème **C**onférence en **I**ngénieri**E** du **L**ogiciel
Paris, CNAM
13 Juin 2014

scade-academics@esterel-technologies.com

Systems Engineering

Model-Based Systems Engineering

Systems & Requirements Engineering

Esterel Technologies mission

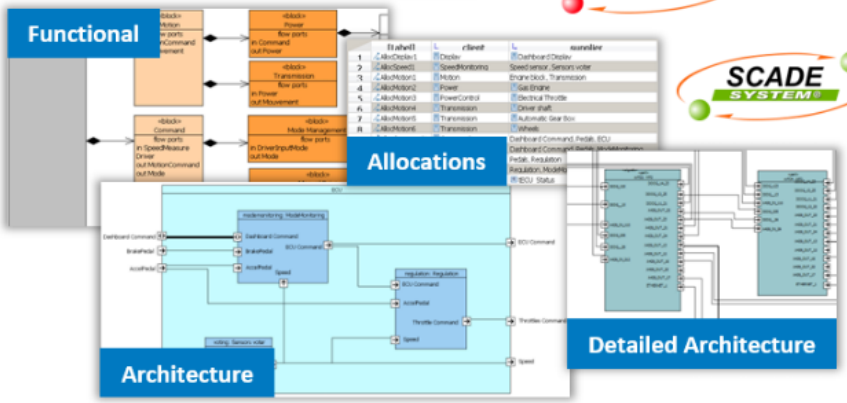
- Provide **critical system and software developers** with **model-based development solutions** that reduce **cost, risk and time-to-certification**
- Esterel Technologies is the **Embedded Systems Business Unit** of ANSYS, Inc.

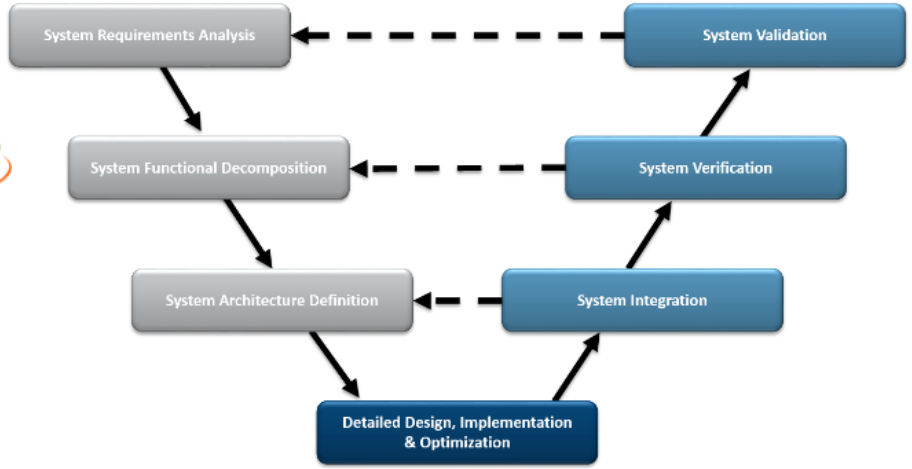
Our vision

Systems Engineering

SCADE LIFECYCLE®

SCADE SYSTEM®



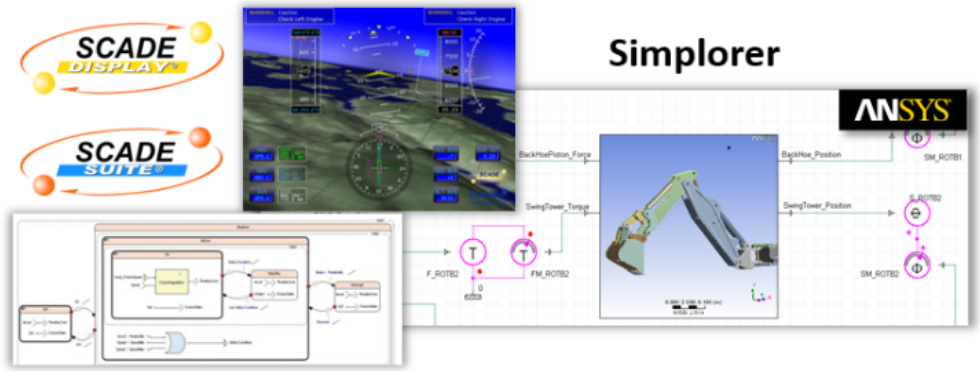


Software Engineering

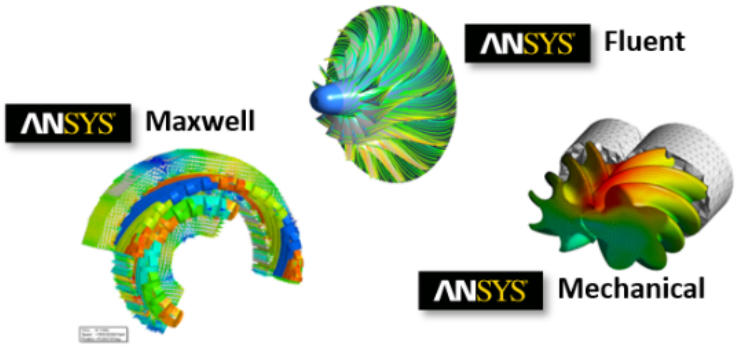
SCADE DISPLAY®

SCADE SUITE®

Simplorer



Detailed Design 3D Multiphysics



SCADE Benefits Summary

Product Development Process Improvements

SCADE enables Best Practices for:

- Model-Based Systems Engineering
- Integrated Multi-physics and Software Simulation
- Embedded Controls development
- Interactive Displays development

Development Costs Reduction

50%

Time-to-Certification Speed up

2x

Systems Engineering

Model-Based Systems Engineering

Systems & Requirements Engineering

What is a system?

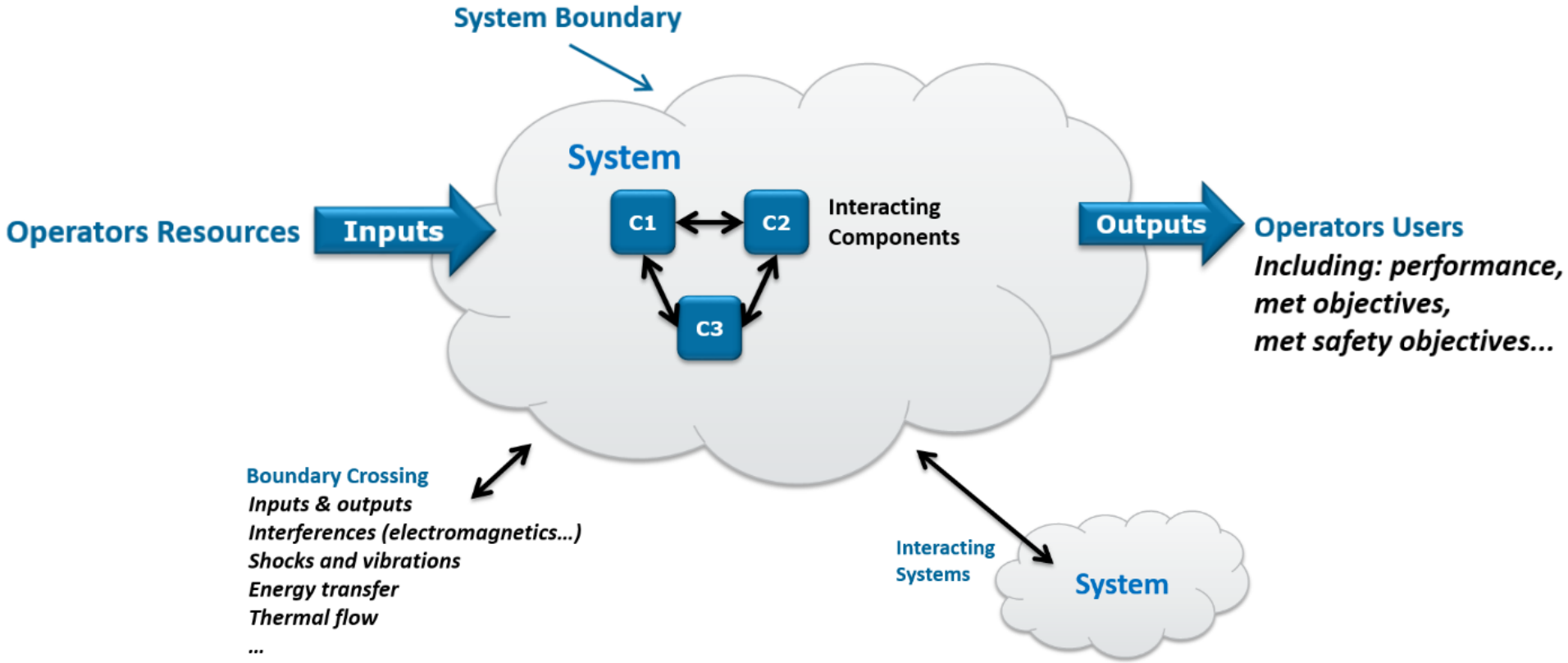
Objective

Mission to accomplish

User expectations

Environment

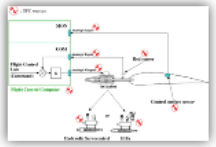
Environment conditions and exposures
Interacting actors and systems
Regulatory rules and other standards…



What is Systems Engineering?

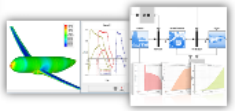
Concept

Control laws analysis

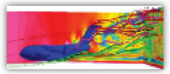


Requirements

Prototyping



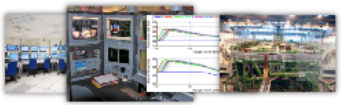
Simulation



Architecture

Systems development

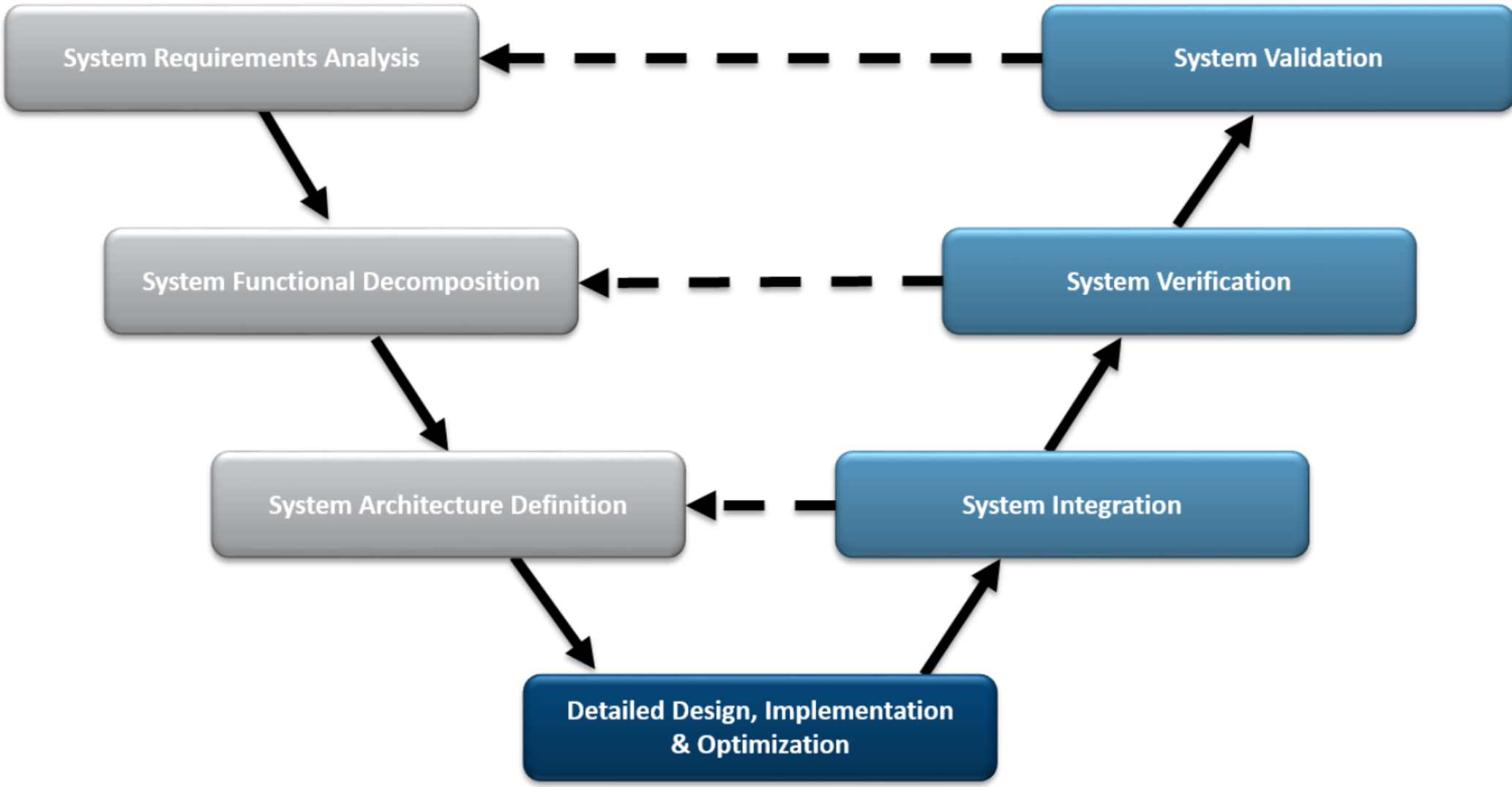
Integration and Testing



Systems Engineering: an extremely complex process

- Both a technical and a management process
- Organizing the technical efforts in the appropriate lifecycle
- Iterative and incremental
- Managing complexity
- Problem Solving oriented and Decision Making centered
- Constantly looking to increase the probability of success
- Reducing risks
- Managing safety and reliability
- Optimizing the global life cycle cost

A Generic Systems Engineering Process V-Cycle



Systems Engineering

Model-Based Systems Engineering

Systems & Requirements Engineering

Model-Based Systems Engineering

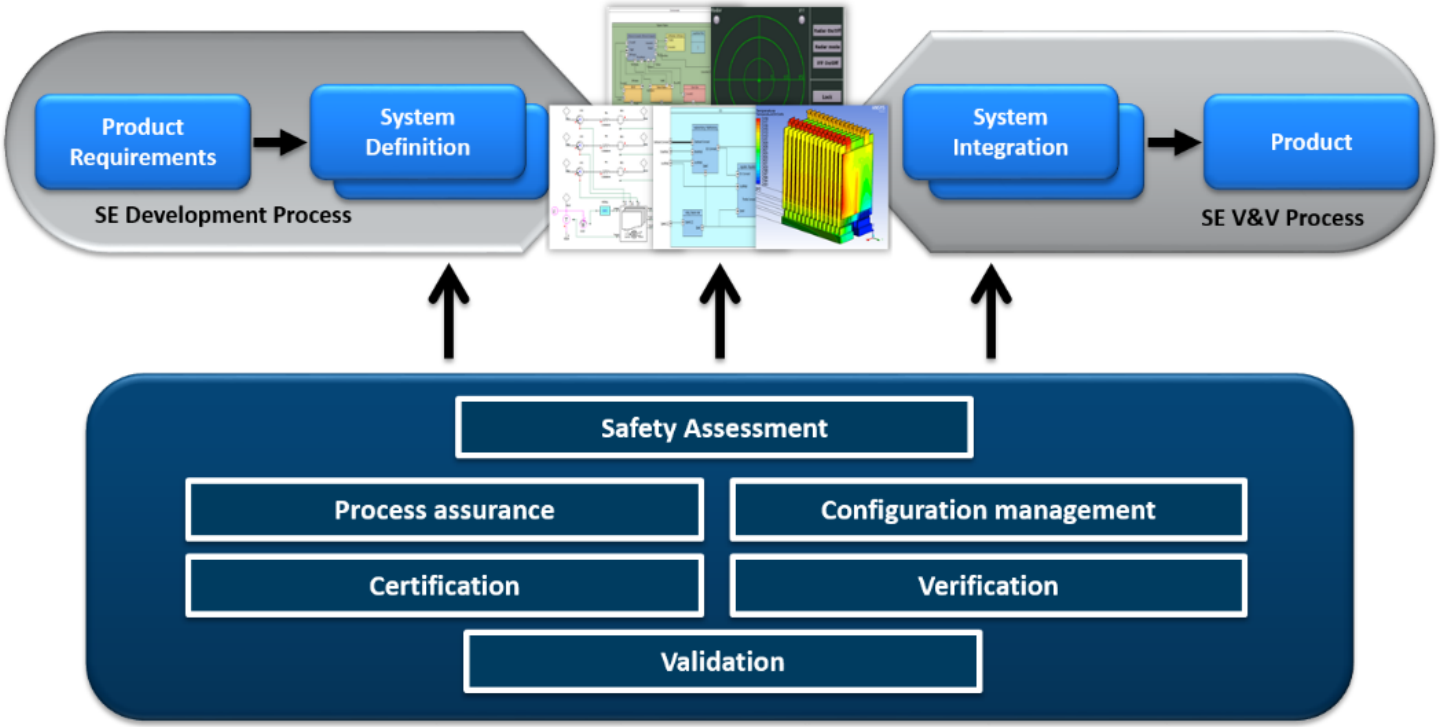
MBSE formalizes the practices of Systems Engineering through the use of models and organizes **models-centric** activities **instead of documents-centric** activities

Model-Based Systems Engineering

"Systems engineers use modeling and simulation on large complex projects to **manage the risk of failure** to meet system mission and performance requirements."

[INCOSE]

The main risk of the Systems Engineering process:
Huge gap between SE development process and SE V&V process



MBSE: Improve Quality and Productivity

- Enable early **validation of concepts**
- Enable identification of **requirements issues**
- Validate the **correctness** of the requirements
- Improve the **allocation** of requirements to items
- Establish rigorous requirements **traceability**, facilitating early verification of requirements **completeness**
- Speed up safe **impact analysis** of requirements changes

MBSE: Insure Integrity and Reliable Communication

- Create a single **reference point** that aggregates views and key data of the project
  - Integrating views from different disciplines into models naturally forces focus on the **system and components boundaries**
  - Key data are managed for an early dependable **Interface Control Documentation (ICD)**
- Enable early design **verification**
- Auto-generate **documentation**

MBSE: Optimize SE Management

- Facilitate the **reuse** of existing valid models
- Enable optimized **iterations** and valuable **increments**
- Allow early **problem solving** activities
- Improve **cost estimates**

Systems Engineering

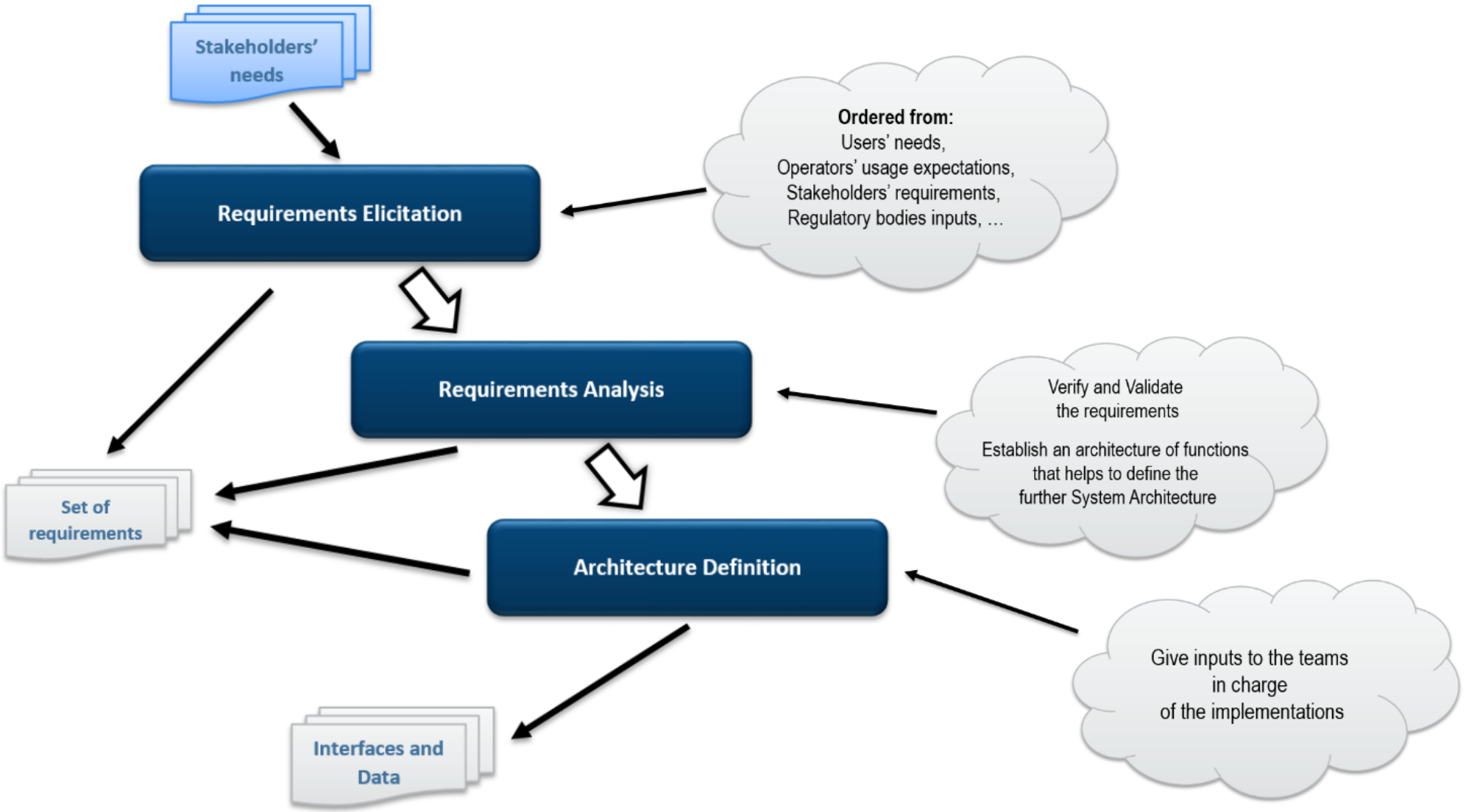
Model-Based Systems Engineering

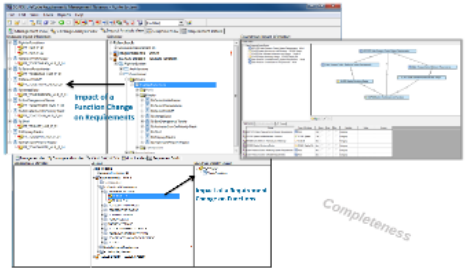
Systems & Requirements Engineering

Questions that drive the early stages of the Systems Engineering process

- What is the system supposed to do?
- How well must it do what it does?
- What is available and allowable to build the system?
- What are the criteria for judging how well resources have been utilized?
- What are the trades-offs between performance and cost?
- How can it be proven that the as-built system meets expectations?

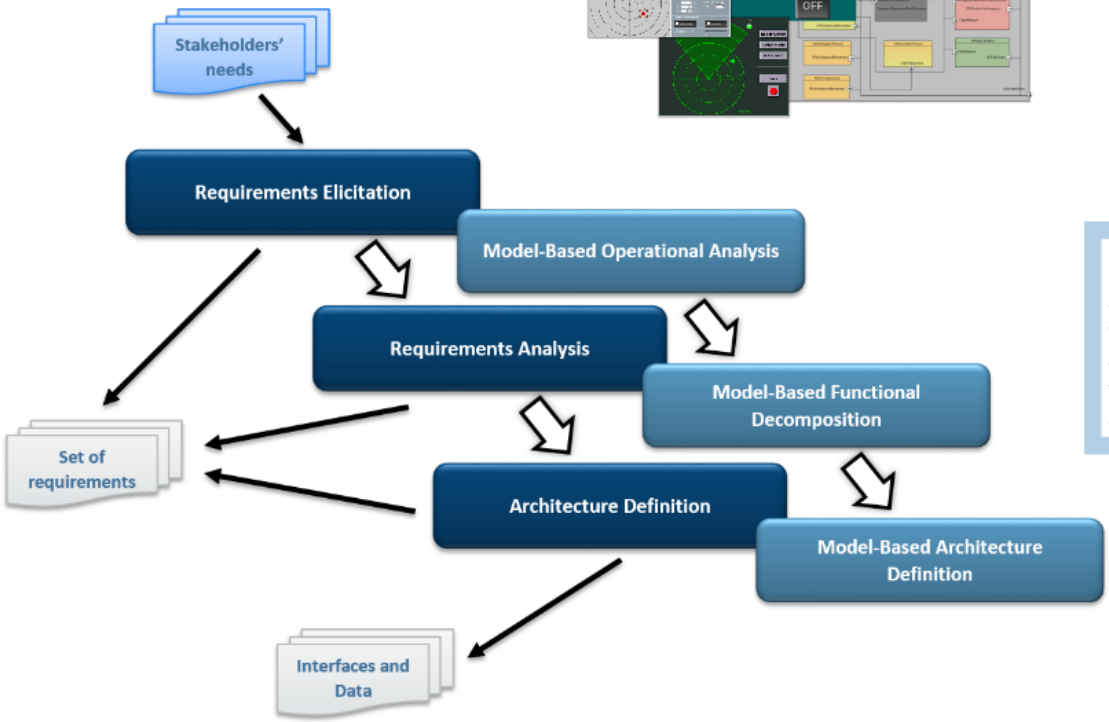
[Albert W. Wymore]


Stakeholders' needs
Requirements Elicitation
Ordered from:
Users' needs,
Operators' usage expectations,
Stakeholders' requirements,
Regulatory bodies inputs, …
Requirements Analysis
Verify and Validate
the requirements
Establish an architecture of functions
that helps to define the
further System Architecture
Set of requirements
Architecture Definition
Give inputs to the teams
in charge
of the implementations
Interfaces and Data



- Are all Parent Requirements covered (aka Stakeholders)?
- Are all Functions of the Function Decomposition traced back to Requirements?
- Are all scenarios of use (Operations) and of maintenance represented?
- Are all types of requirements represented: Safety, Regulatory, Company standards?
- Are all interfaces to other systems, actors, and processes embodied?
- For a defined use, is the prohibited use defined as well?

Completeness

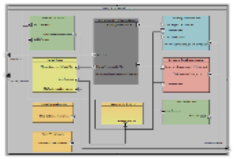


(aside) What's the difference?

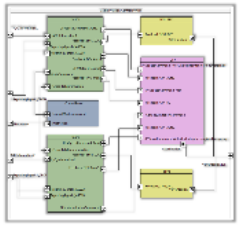
Functional	Architectural
- Takeoff and Land	
- Sense Position and Velocity	
- Navigate	
- Produce Horizontal Thrust	
- Produce Vertical Lift	

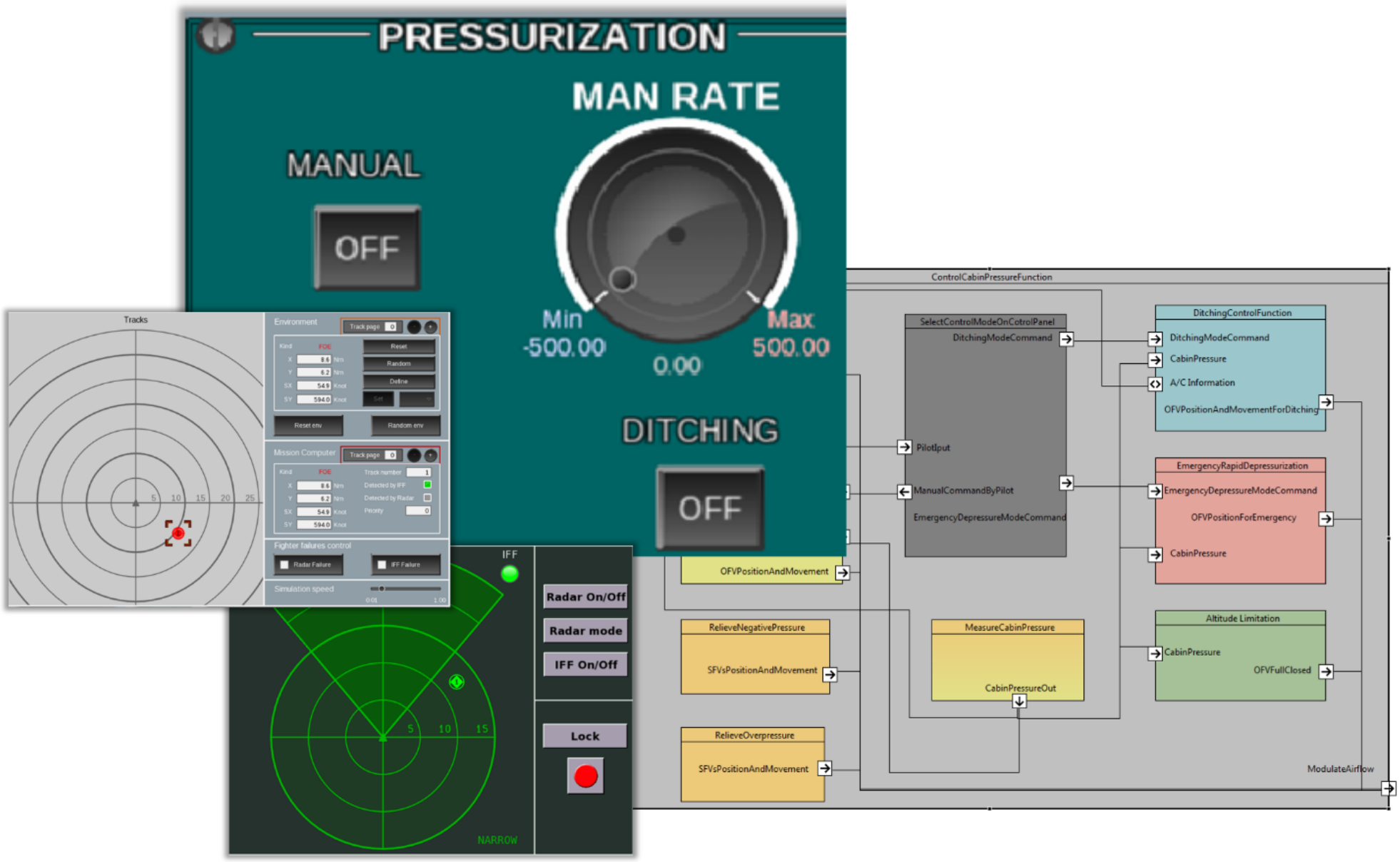
- Is the requirement unambiguous?
- Is it clearly stated as a requirement?
- Is it not redundant?
- On the other hand, does it conflict with other requirements?
- Is it feasible to serve it?
- Is it verifiable? (Testable?)
- If it is a derived requirement: is it justified?
- Is it necessary?

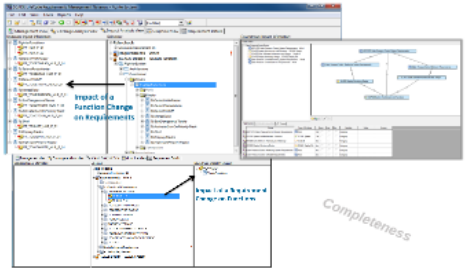
Consistency



Finalize and make decisions

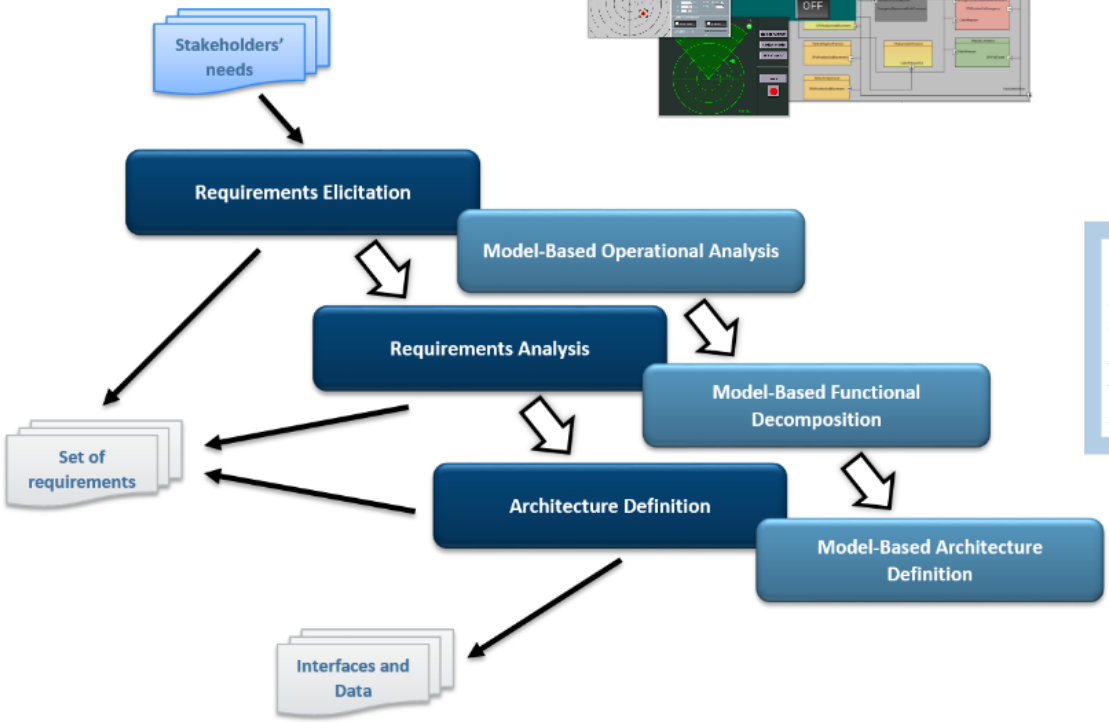



PRESSURIZATION
MAN RATE
MANUAL
OFF
Min
-500.00
0.00
Max
500.00
DITCHING
OFF
IFF
Radar On/Off
Radar mode
IFF On/Off
Lock
NARROW
Tracks
Environment
Mission Computer
Fighter failures control
Radar Failure
IFF Failure
Simulation speed
ControlCabinPressureFunction
SelectControlModeOnCotrolPanel
DitchingModeCommand
PilotIput
ManualCommandByPilot
EmergencyDepressureModeCommand
OFVPositionAndMovement
DitchingControlFunction
DitchingModeCommand
CabinPressure
A/C Information
OFVPositionAndMovementForDitching
EmergencyRapidDepressurization
EmergencyDepressureModeCommand
OFVPositionForEmergency
CabinPressure
Altitude Limitation
CabinPressure
OFVFullClosed
RelieveNegativePressure
SFVsPositionAndMovement
MeasureCabinPressure
CabinPressureOut
RelieveOverpressure
SFVsPositionAndMovement
ModulateAirflow



- Are all Parent Requirements covered (aka Stakeholders)?
- Are all Functions of the Function Decomposition traced back to Requirements?
- Are all scenarios of use (Operations) and of maintenance represented?
- Are all types of requirements represented: Safety, Regulatory, Company standards?
- Are all interfaces to other systems, actors, and processes embodied?
- For a defined use, is the prohibited use defined as well?

Completeness

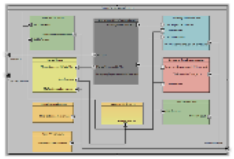


(aside) What's the difference?

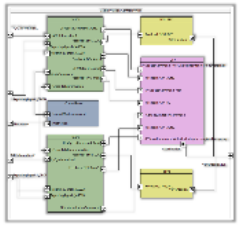
Functional	Architectural
- Takeoff and Land	
- Sense Position and Velocity	
- Navigate	
- Produce Horizontal Thrust	
- Produce Vertical Lift	

- Is the requirement unambiguous?
- Is it clearly stated as a requirement?
- Is it not redundant?
- On the other hand, does it conflict with other requirements?
- Is it feasible to serve it?
- Is it verifiable? (Testable?)
- If it is a derived requirement: is it justified?
- Is it necessary?

Consistency



Finalize and make decisions



(aside) What's the difference?

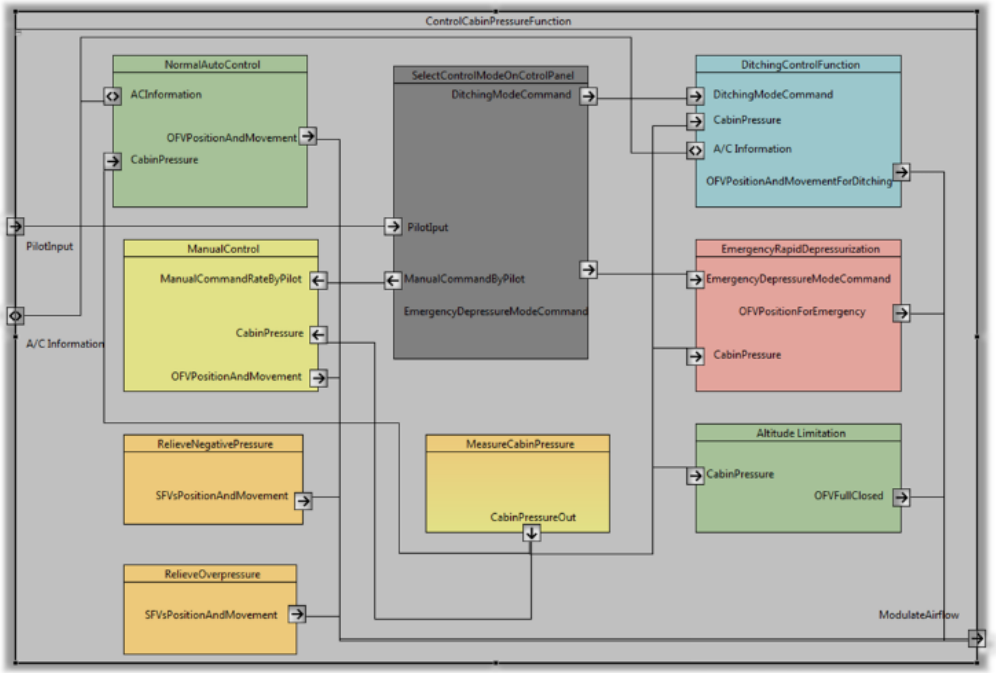
Functional

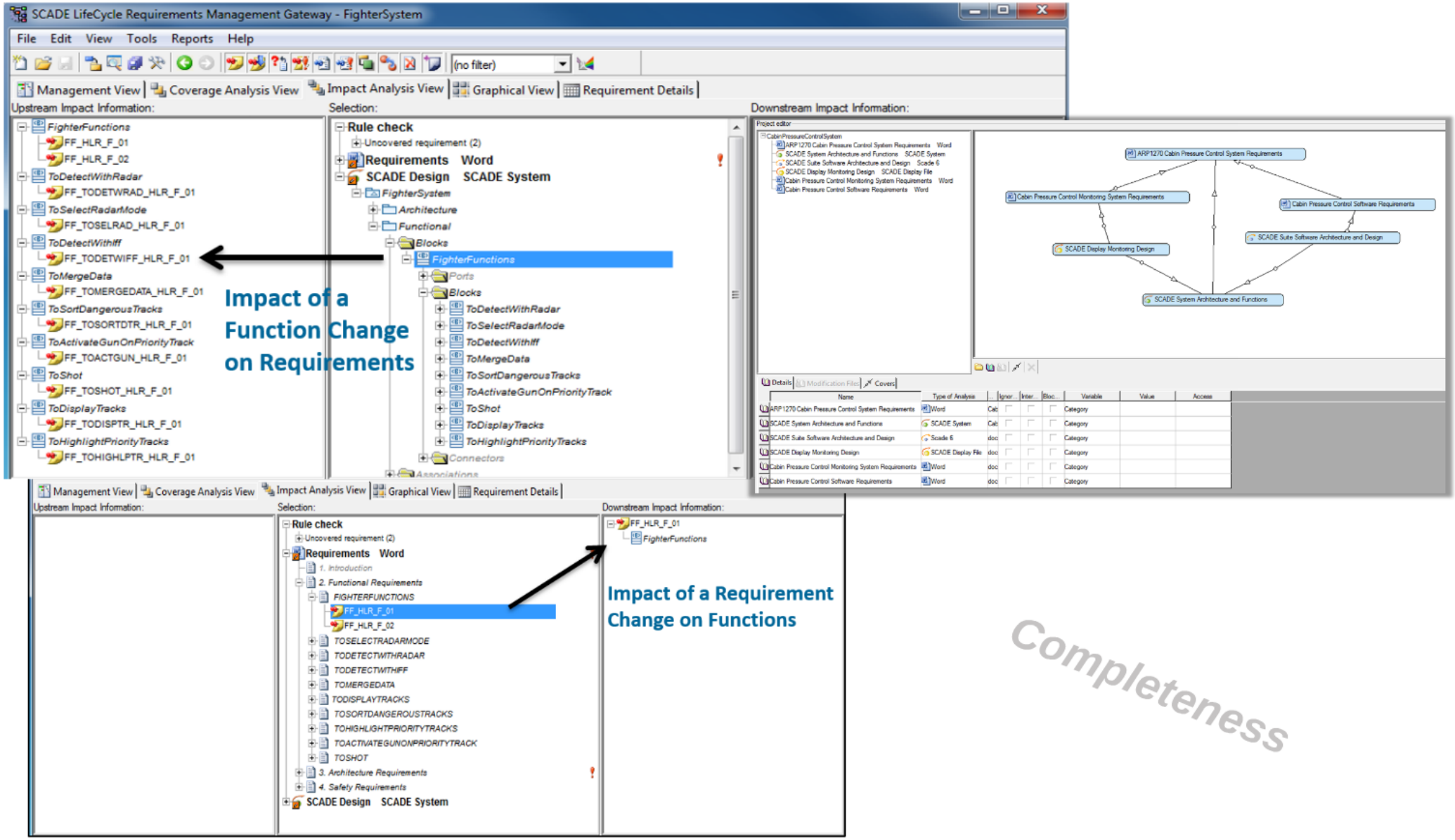
- Takeoff and Land
- Sense Position and Velocity
- Navigate
- Produce Horizontal Thrust
- Produce Vertical Lift

Architectural

Correctness

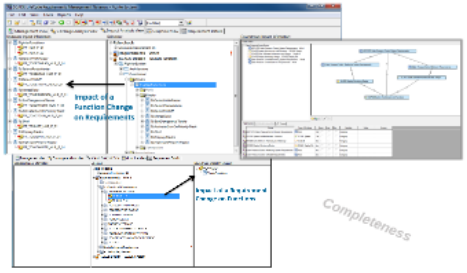
- Is the requirement unambiguous?
- Is it clearly stated as a requirement?
- Is it not redundant?
- On the other hand, does it conflict with other requirements?
- Is it feasible to serve it?
- Is it verifiable? (Testable?)
- If it is a derived requirement: is it justified?
- Is it necessary?





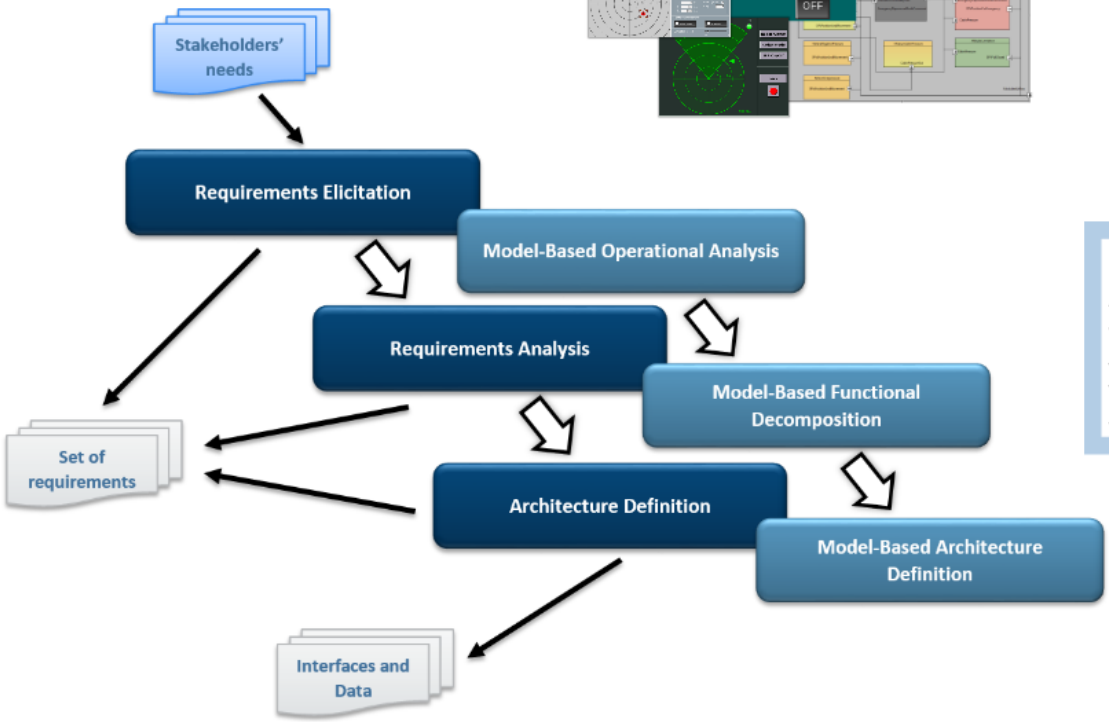
Completeness

- Are all Parent Requirements covered (aka Stakeholders')?
- Are all Functions of the Function Decomposition traced back to Requirements?
- Are all scenarios of use (Operations) and of maintenance represented?
- Are all types of requirements represented: Safety, Regulatory, Company standards?
- Are all interfaces to other systems, actors, and processes embodied?
- For a defined use, is the prohibited use defined as well?



- Are all Parent Requirements covered (aka Stakeholders)?
- Are all Functions of the Function Decomposition traced back to Requirements?
- Are all scenarios of use (Operations) and of maintenance represented?
- Are all types of requirements represented: Safety, Regulatory, Company standards?
- Are all interfaces to other systems, actors, and processes embodied?
- For a defined use, is the prohibited use defined as well?

Completeness

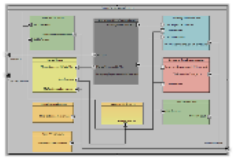


(aside) What's the difference?

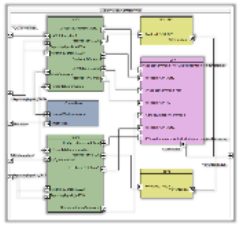
Functional	Architectural
- Takeoff and Land	
- Sense Position and Velocity	
- Navigate	
- Produce Horizontal Thrust	
- Produce Vertical Lift	

- Is the requirement unambiguous?
- Is it clearly stated as a requirement?
- Is it not redundant?
- On the other hand, does it conflict with other requirements?
- Is it feasible to serve it?
- Is it verifiable? (Testable?)
- If it is a derived requirement: is it justified?
- Is it necessary?

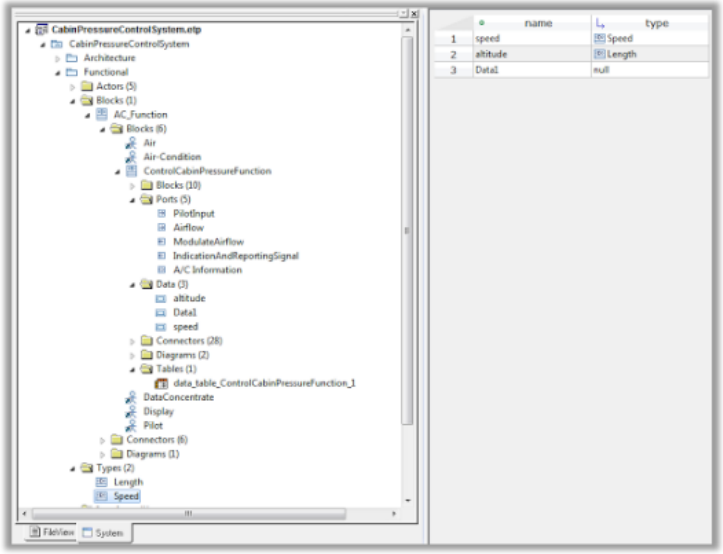
Consistency

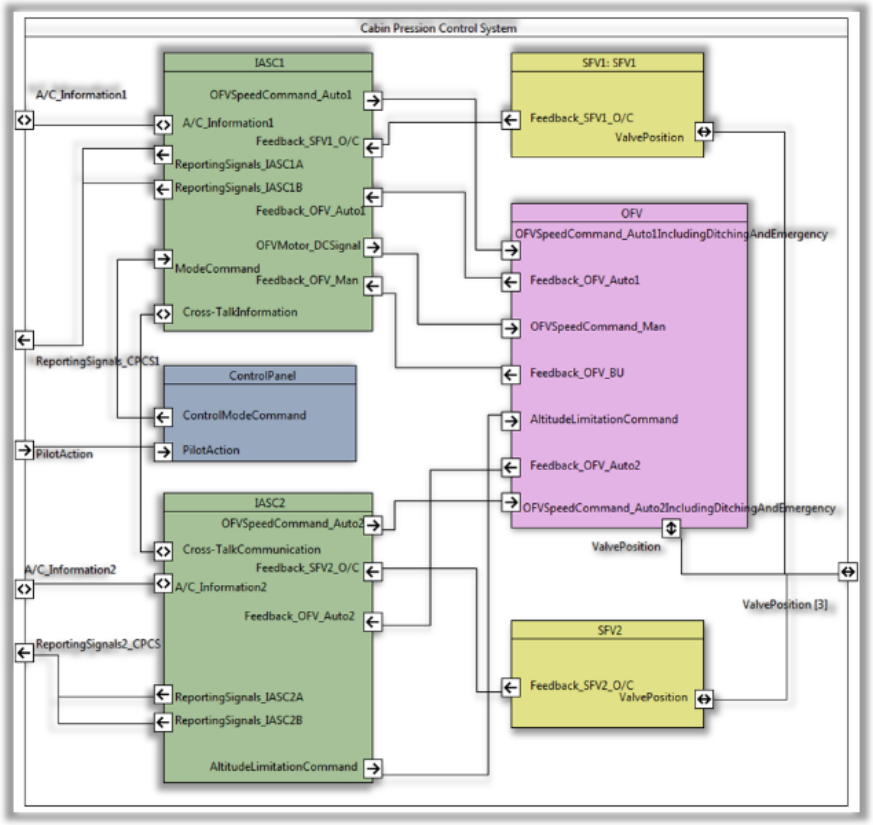


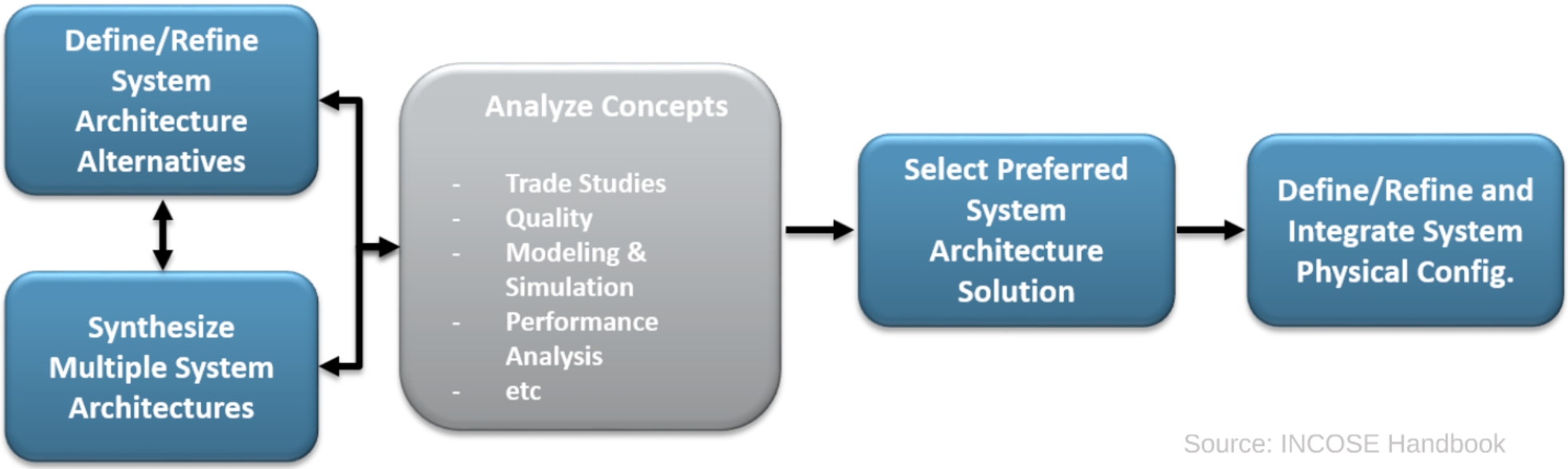
Finalize and make decisions



Finalize and make decisions

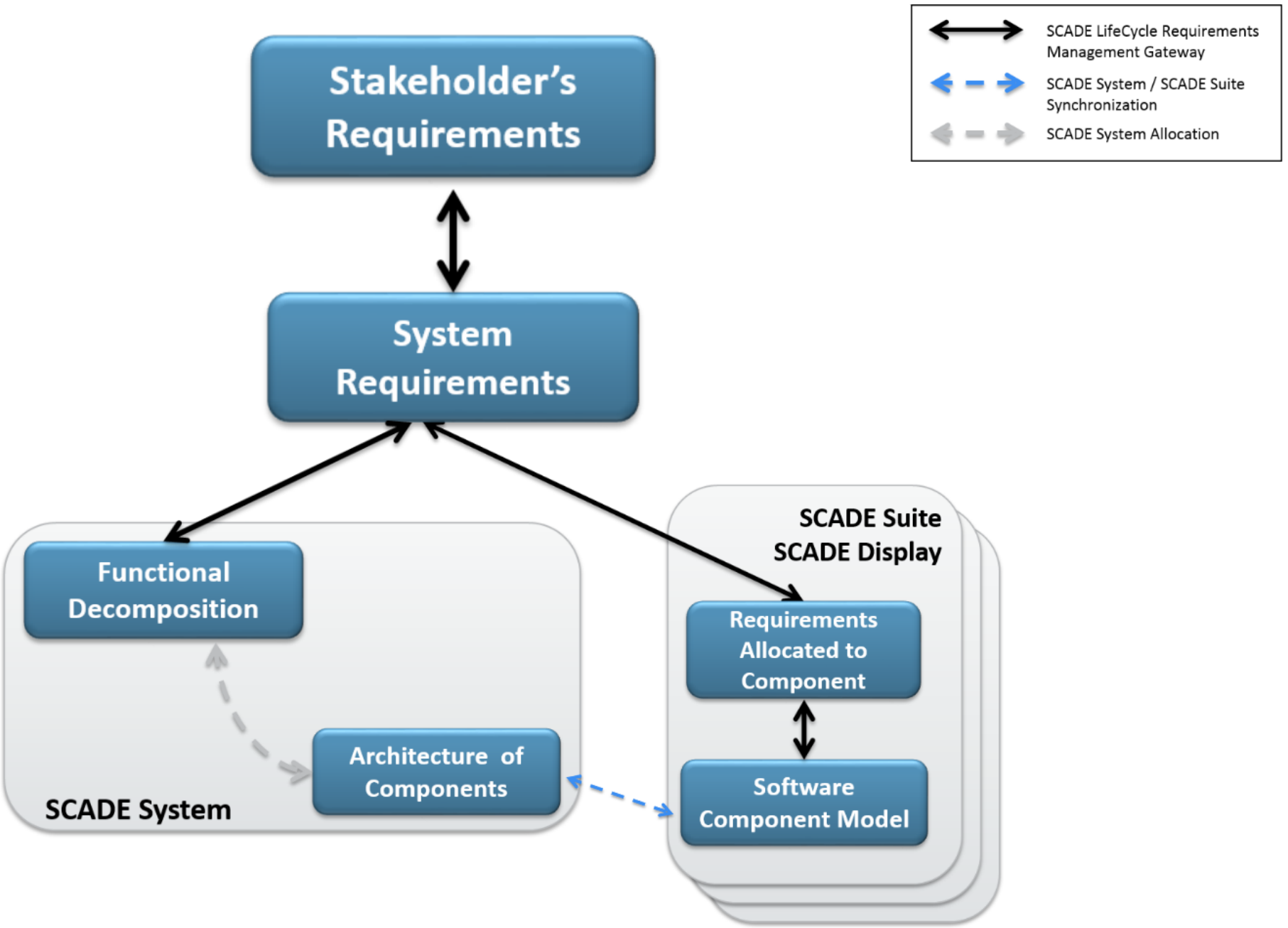






Source: INCOSE Handbook

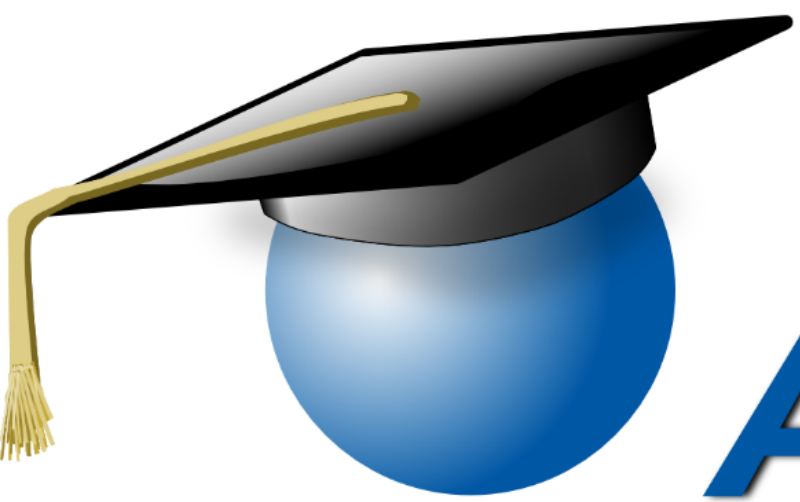
Traceability management with SCADE



Conclusion

Summary

- Creating a set of **complete and correct Requirements** is the primary responsibility of Systems Engineers, from the point of view of other teams involved in the construction of a system.
- The **Model-Based approach**, as well as Data-Based representation, help System Engineers to implement a true Requirements Engineering process.
- This approach includes the use of the **Rapid Prototyping capability** to simulate, early in the development process, the Systems operations.
- Functional Decomposition, synthesis of Architecture exploration and Interface Control Document are created and maintained through **safe iterations**, all tightly linked to the set of requirements.



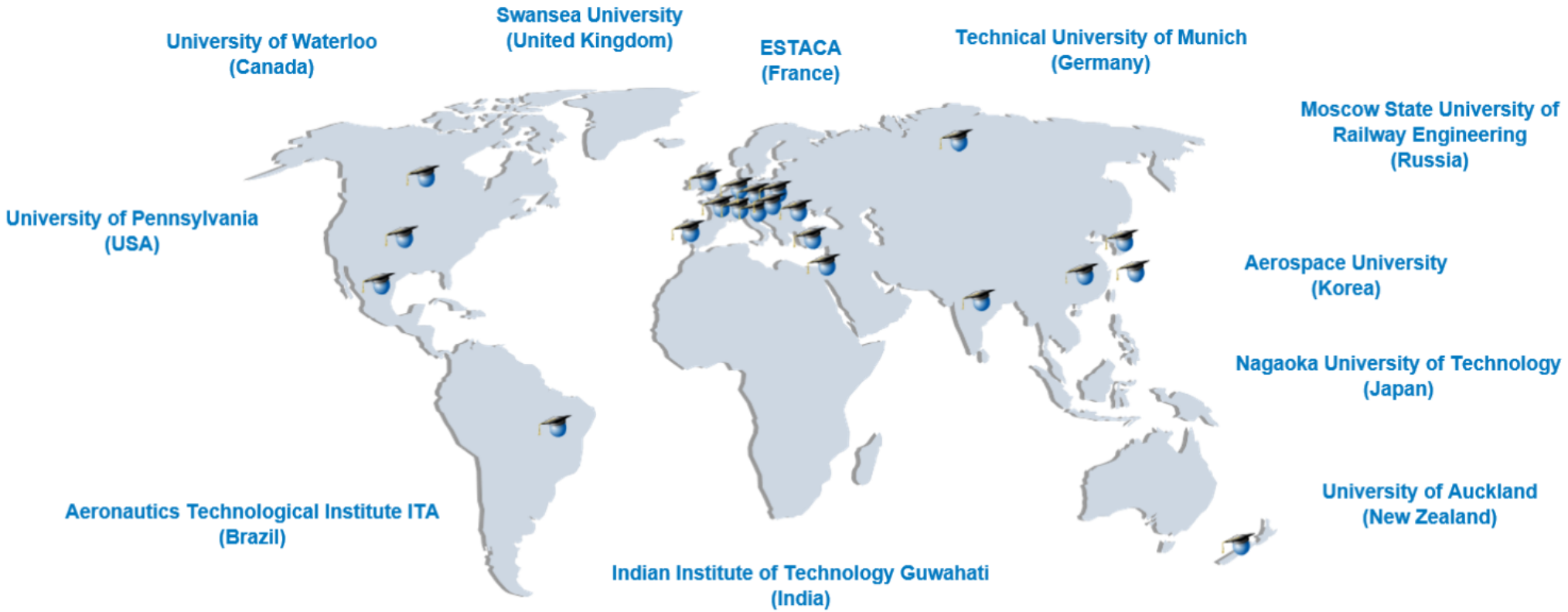
SCADE®
ACADEMIC

The SCADE Academic Program

Designed to **support teachers and researchers using SCADE** embedded system and software **for academic purposes**

Part of the ANSYS Academic Program

At the early days of the Program: 70 academies in 19 countries



SCADE® ACADEMIC

Join the SCADE Academic Community

scade-academics@esterel-technologies.com