Challenges of Needs and Requirements Definition and Management for Complex Systems

North Texas Chapter

10 August 2021

Lou Wheatcraft, Senior Consultant, Wheatland Consulting
Wheatland.consulting@gmail.com

Lou Wheatcraft

- **Lou Wheatcraft** is a senior consultant and managing member of Wheatland Consulting, LLC. Lou is an expert in systems engineering with a focus on needs and requirements development, management, verification, & validation. Lou provides consulting and mentoring services to clients on the importance of well-formed needs & requirements helping them implement needs & requirement development and management processes, reviewing and providing comments on their needs and requirements, and helping clients write well-formed needs & requirements.
- Specialties include: Understanding and documenting the problem; defining project & product scope; defining and maturing system concepts; assessing, mitigating, & managing risk; documenting stakeholder needs; transforming needs into well formed design input requirements; allocation, budgeting, and traceability; interface management, requirement management; & verification and validation.
- Lou's goal is to help clients practice better systems engineering from a needs & requirements perspective across all life cycle stages of system/product development. Getting the needs & requirements right upfront is key to a successful project. Poor needs & requirements can triple the chances of project failure.
- Lou has over 50 years' experience in systems engineering, including 22 years in the United States Air Force. Lou has taught over 200 requirement seminars over the last 21 years. Lou supports clients from all industries involved in developing and managing systems and products including aerospace, defense, medical devices, consumer goods, transportation, and energy.
- Lou has spoken at Project Management Institute (PMI) chapter meetings and INCOSE conferences and chapter meetings. Lou has published and presented many papers concerning needs and requirement for NASA's *PM Challenge*, INCOSE, INCOSE *INSIGHT Magazine*, and *Crosstalk Magazine*. Lou is a member of INCOSE, past Chair and current Co-Chair of the INCOSE Requirements Working Group (RWG), a member of the Project Management Institute (PMI), the Software Engineering Institute (SEI), the World Futures Society, and the National Honor Society of Pi Alpha Alpha.
- Lou has a BS degree in Electrical Engineering from Oklahoma State University; an MA degree in Computer Information Systems; an MS degree in Environmental Management; and has completed the course work for an MS degree in Studies of the Future from the University of Houston – Clear Lake.

Requirement Working Group (RWG) Charter

1 **Purpose**

Advance the practices, education and theory of needs and requirements definition and management and the relationship of needs and requirements to other systems engineering functions.

2 **Goal**

Expand and promote the body of knowledge of needs and requirements and their benefits within the systems engineering community

3 **Scope**

Activities relating to best practices for needs and requirements definition and management throughout the product lifecycle including:

Elicitation	Analysis	Allocation/Budgeting	Traceability
Elaboration	Management	Change Management	
Expression	Verification	Validation	

RWG Leadership

- **Chair**: Tami Katz; Ball Aerospace, USA
- **Co-Chair**: Lou Wheatcraft, Wheatland Consulting, USA
- **Co-Chair**: Rick Zinni, Harris Corp, USA
- **Co-Chair**: Mike Ryan; Retired

- **INCOSE Connect address:**
- **https://connect.incose.org/WorkingGroups/Requirements/Pages/Home.aspx**

- **Number of Members:** 389, largest INCOSE WG

The RWG is comprised of members from industry and academia with a common purpose of improving the practice of systems engineering through improvement of **Needs and Requirements** definition and management

Becoming Involved in RWG

- As a large working group, the RWG has been very active in virtual events as well as smaller product team efforts.
- Joining the working group enables the members to learn about the products, provide opportunity to contribute (or review) products, and participate in the RWG virtual events with other practitioners.
- Members can be very involved (product support) or minimally involved (watch meeting recordings), the intention is to enable all levels of participation and interaction.

- **Members**:
  - Contribute the benefit of their experience
  - Promote the purposes of the group
  - Write, edit and review work products
  - Liaise with other working groups and organizations
  - Assist the leadership team with specific activities

Joining the RWG

1

My Committees/Working Groups *(Join a group here)*

Committee	Position
SE Tools Database	Member

- View My Committees/Working Groups
- Browse / Join a Working Group

3

Committee Tasks

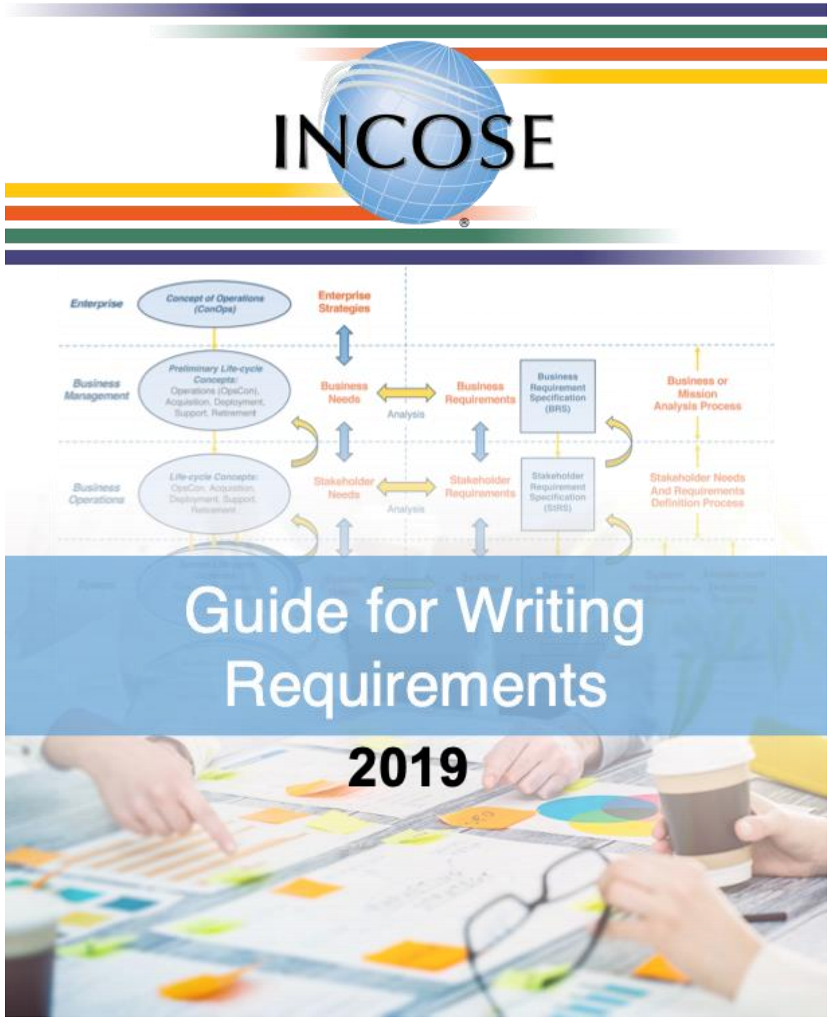
- Join this Committee
- Back to All Committees
- Go to Portal Home

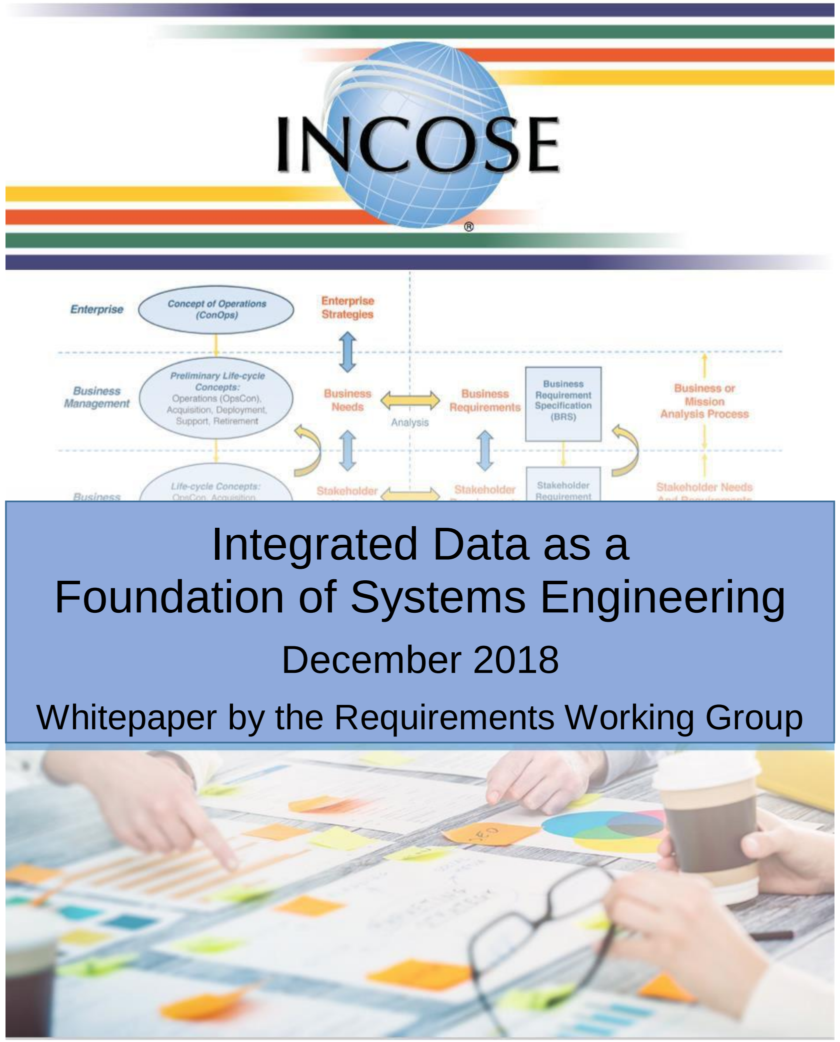
Product Line Engineering	151	(view)
Professional Development Initiative	33	(view)
Profl Development Steering Grp	29	(view)
Publications	1	(view)
Publications Office	4	(view)
Requirements	382	(view)
Resilient Systems	73	(view)
Risk Management	111	(view)

2 Click on (view)

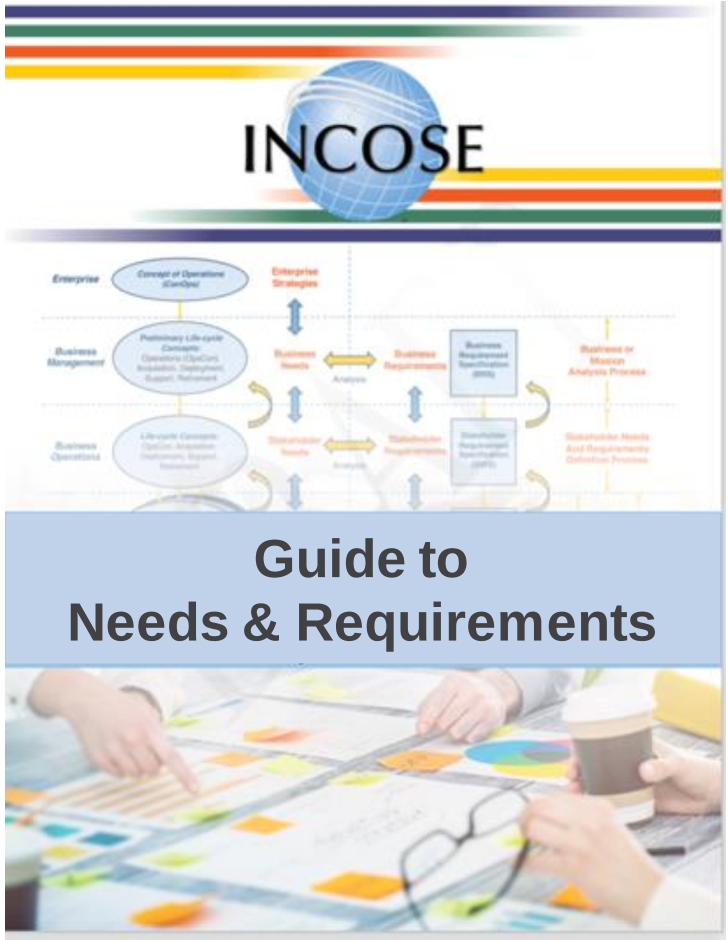
4 Go to “Edit Your Information” and under “Communications Preferences” be sure to ”opt in” for Working Group emails

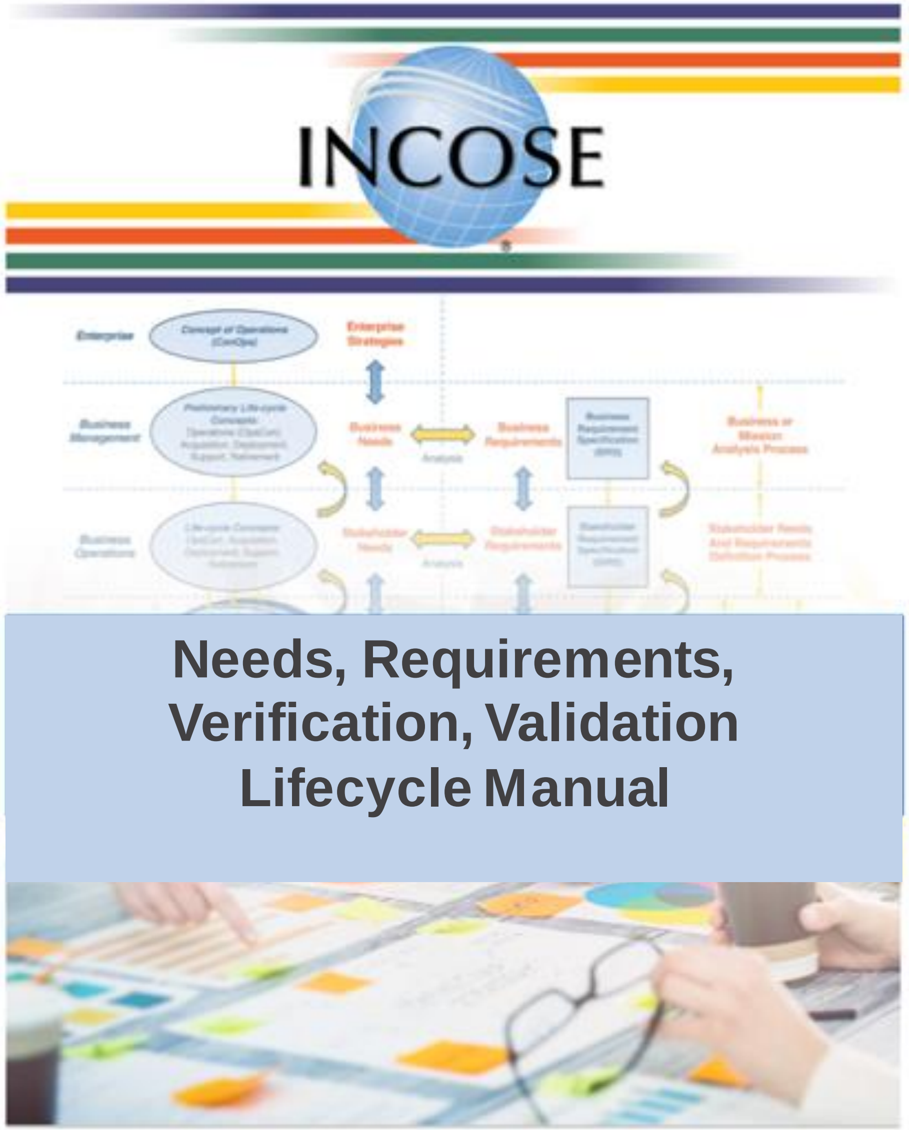
Released RWG Products –
Available for download from the INCOSE Store



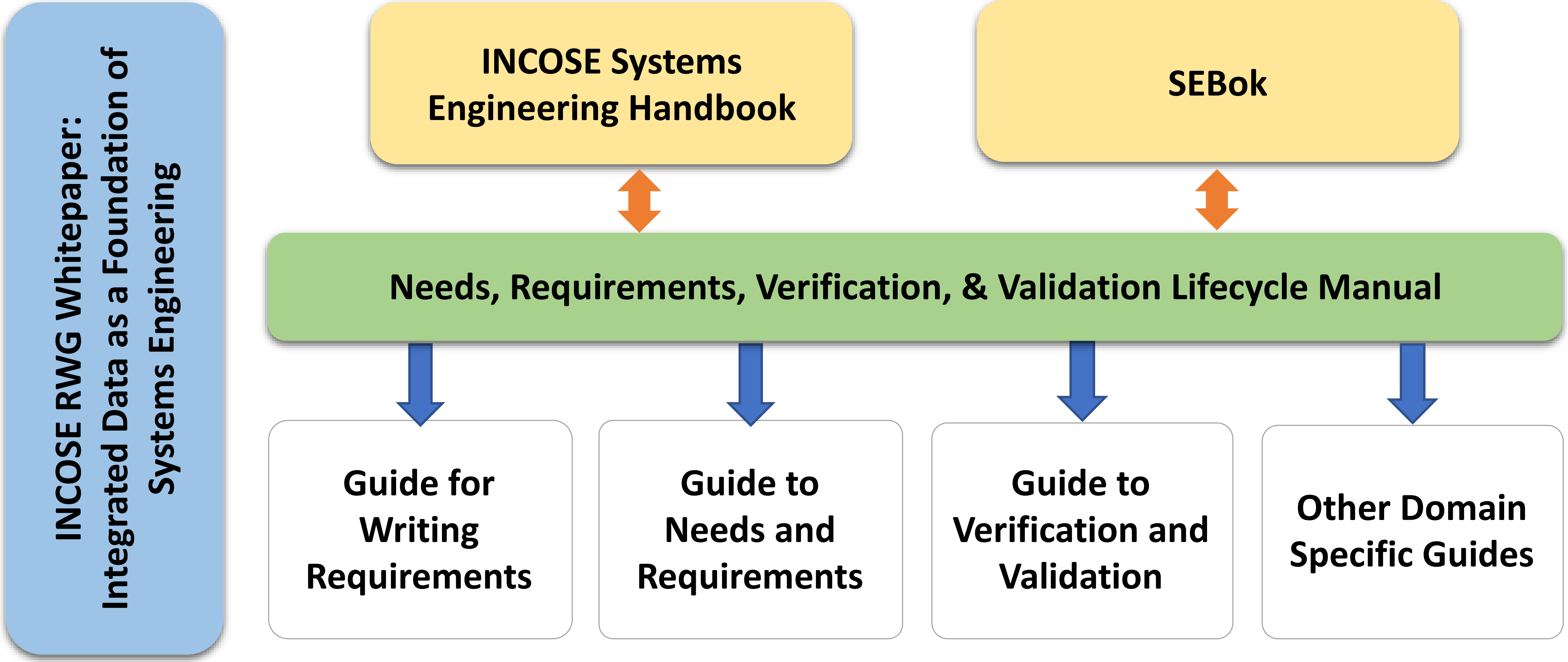


RWG Products in Work





RWG Product Relationships



“INCOSE RWG” YouTube Channel

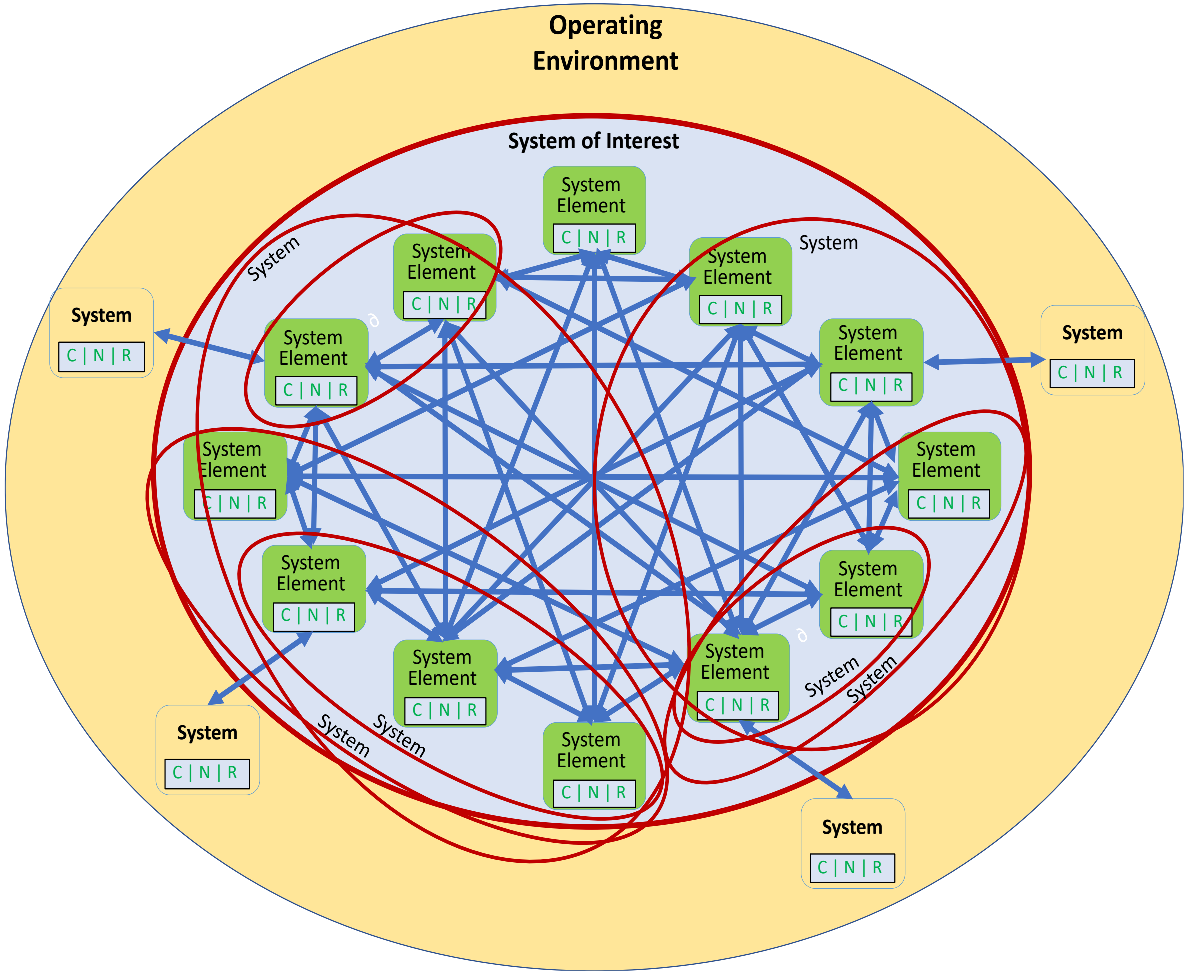
- About the RWG: https://youtu.be/L_Z6Xitprol

- White paper “Integrated Data as a Foundation of SE” https://youtu.be/Rc3O6lPO5x4

- Needs, Requirements, Verification, and Validation Lifecycle Manual
  - Overview and contents: https://youtu.be/g_fJk_UBONM
  - Basic concepts: https://youtu.be/ZRli_wSCmRg

Subscribe to be informed as we add new content!

Today's increasingly complex software-centric systems



Yesterday's electro-mechanical systems had fewer interactions both internally and externally - Interfaces could be shown on drawings.

For software-centric systems interactions, both internally and externally, have increased several orders of magnitude as have the threats and vulnerabilities.

Critical functions are carried out by the software - Electro-mechanical parts of a system are "enablers" for the software.

INCOSE Vision 2025 (INCOSE 2014)

"…… a constant throughout the evolution of systems engineering is an ever-increasing complexity of systems which can be observed in terms of the number of system functions, components, and interfaces and their non-linear interactions and emergent properties."

Increase in Software-Centric Systems: Automobiles

- The success of a car depends on its software more than the mechanical/hardware side
- Today, high-end cars may contain 150 Electronic Control Units (ECUs) or more, while pick-up trucks like Ford's F-150 have greater than 150 million lines of code.
- 40% of the cost of a new car can be attributed to semiconductor-based electronic systems, a cost doubling since 2007. This total will approach 50% by 2030.
- Each new car today has about $600 worth of semiconductors packed into it, consisting of up to 3,000 chips of all types.
- 40% or more of a vehicle's development budget, from the start of its development to the beginning of production, can be attributed to systems integration, testing, verification, and validation of these components and associated software.
- An automobile's network harness, which can attach thousands of components, may contain more than 1,500 wires totaling 5,000 meters in length and weigh in excess of 68 kg

How Software is Eating the Car; IEEE Spectrum, by Robert N. Charette Posted 2021-06-07

Increase in Software-Centric Systems: Automobiles

- Nearly all ECU design and software is outsourced to suppliers, with the OEM integrating the ECUs to create an integrated system with the desired customizable functionality.
- ~ 10% of the software is developed in-house.
- The other 90% is provided by as many 50 or more suppliers each with their own development approach, operating systems, and languages adding another level of complication, especially when performing system integration, verification, and validation.
- Individual suppliers often do not have insight into how OEMs integrate ECUs together.
- Similarly, OEMs have limited insight into the software resident within the ECUs which are often acquired as a "black box" procured for a specific purpose.

How Software is Eating the Car; IEEE Spectrum, by Robert N. Charette Posted 2021-06-07

Increase in Software-Centric Systems: Automobiles

- When asked how difficult it was to know when a code change in one ECU affects another;
  - 37% of those surveyed indicated it was difficult,
  - 31% indicated it was very difficult,
  - 7% indicated that it was pretty darn close to impossible, while
  - 16% indicated that it was not possible.
- Each increase in functionality implies additional sensors, actuators, ECUs and accompanying software, and consequently extra system integration, verification, and validation efforts to ensure they work correctly when integrated into the system.
- Nearly 30% of the defects are related to software integration where a failure results from software interfaces with other electronic components or systems in a vehicle.
- Many recalls are due to a software issue.
- Increasingly, hardware issues are fixed by a software patch.

How Software is Eating the Car; IEEE Spectrum, by Robert N. Charette Posted 2021-06-07

Challenges

- Defining and managing needs and requirements across the system lifecycle is increasingly challenging when developing today's complex, software-centric systems - especially for systems that are being contracted out to suppliers.

- These challenges are a result of **increases in**:
  - Complexity
  - The role software has in the system architecture (software-centric systems are the norm)
  - Dependencies and number of interactions between parts of the system
  - The interactions between a system and the macro system it is a part
  - The number of threats across interface boundaries and vulnerabilities to those threats
  - Dependencies between project management and systems engineering
  - Dependencies between systems engineering lifecycle process activities and artifacts
  - Oversight
  - Competition
  - The pressure (and need) to reduce development time and time to market
  - Risks: program/project, development, manufacturing, system integration, system verification, system validation, and operations
  - The number of projects that are over budget and experiencing schedule slippage

Recent article by John Mauldin "Technology Rules"

"….it's remarkable how many industries and government agencies are still operating on ancient (as in 1990s) technology, [*just*] muddling through."

"... what happens when those organizations take off their old-tech handcuffs? They will run better and develop new capabilities they never had before. Customers, workers, and investors will all benefit."

"Humans have a comparative advantage at higher levels of abstraction: creativity, intuition, and holistic judgements. Each is necessary. The best technologies do not automate complex problems, as many assume; they equip people to solve them faster and more effectively."

We are 21 years into the 21st Century, why are so many still practicing system engineering based on outdated 20th Century electro/mechanical, document-centric methodologies?

Change or risk becoming irrelevant

"Several INCOSE dignitaries have been warning since about 2013 that if INCOSE did not take the lead to quickly understand software and start leading in software-intensive systems, INCOSE would be at risk of becoming irrelevant. (Roedler 2018, Stoewer 2017).

INCOSE Past President Garry Roedler provides a quote attributed to Jack Welch of GE, "*If the rate of change on the outside exceeds the rate of change on the inside, the end is near*," as support to the idea that **INCOSE's rate of change must increase to match the rate of change in industry and the rapidly evolving technology universe**."

This applies equally to organizations practicing Systems Engineering!

S. Sheard, M. Bouyaud, M. Osaisai, J. Siviy, K. E. Nidiffer; *A Guide for Systems Engineers to Finding Your Role in 21st-Century Software-Dominant Organizations*, Technical Paper presented at IS2021

Overview

To address these challenges, we need to change how we currently practice Systems Engineering AND Project Management

- Key areas of change discussed in this presentation include:
  - Needs AND requirements
  - Integrated, multidisciplined, collaborative, project team – minimize silos!
  - Managing the Integrated System From the Beginning
  - Allocation and budgeting for software-centric systems
  - Data-centric practice of systems engineering
  - Increased focus on Validation across the lifecycle

Needs AND Requirements

Needs vs Requirements

- Needs represent the stakeholder, customer/acquirer view of the system of interest (SOI)
  - What do the stakeholders need the system to do that will result in their problem to be solved or opportunity to be realized within defined constraints?
  - Communicates the stakeholder expectations for the end-state once the SOI is delivered – in the end **what will make the customer happy**?
  - The SOI will be validated against its integrated set of needs

- Requirements represent the technical, developer view of the SOI
  - What must the SOI do in order to meet the needs?
  - The SOI will be verified against its design input requirements

The quality of the requirements is dependent on the quality of the needs from which they are transformed.

Rather than just focusing on the requirements, we must focus on both needs and requirements!!

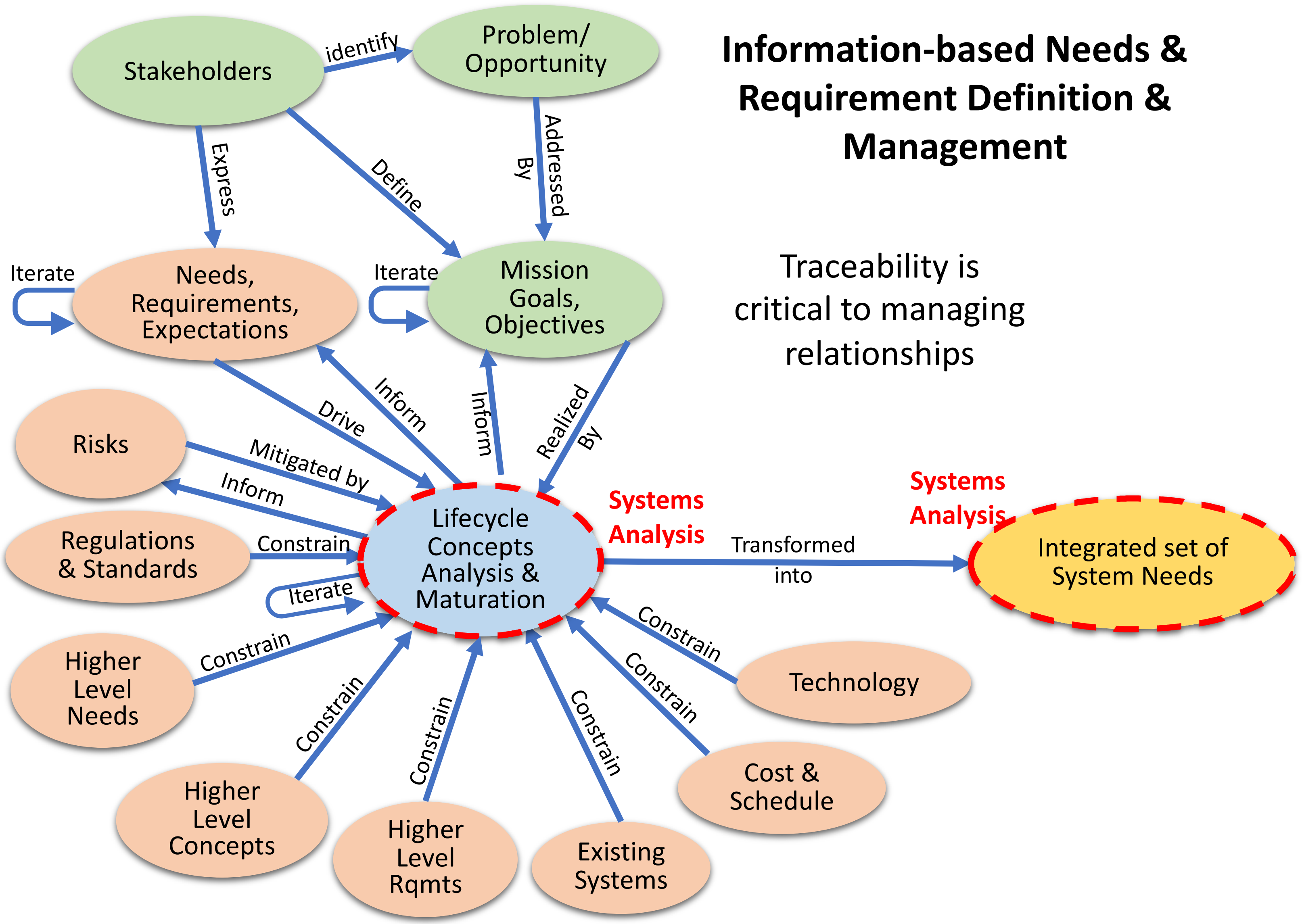
Needs and Requirements are the common treads that tie all SE lifecycle activities and artifacts together.

From a holist view of SE, needs and requirements must be defined and managed in the context of all other SE process areas rather than in a silo distinct from the other process areas.

Needs Before Requirements

Information-based Needs & Requirement Definition & Management

Traceability is critical to managing relationships

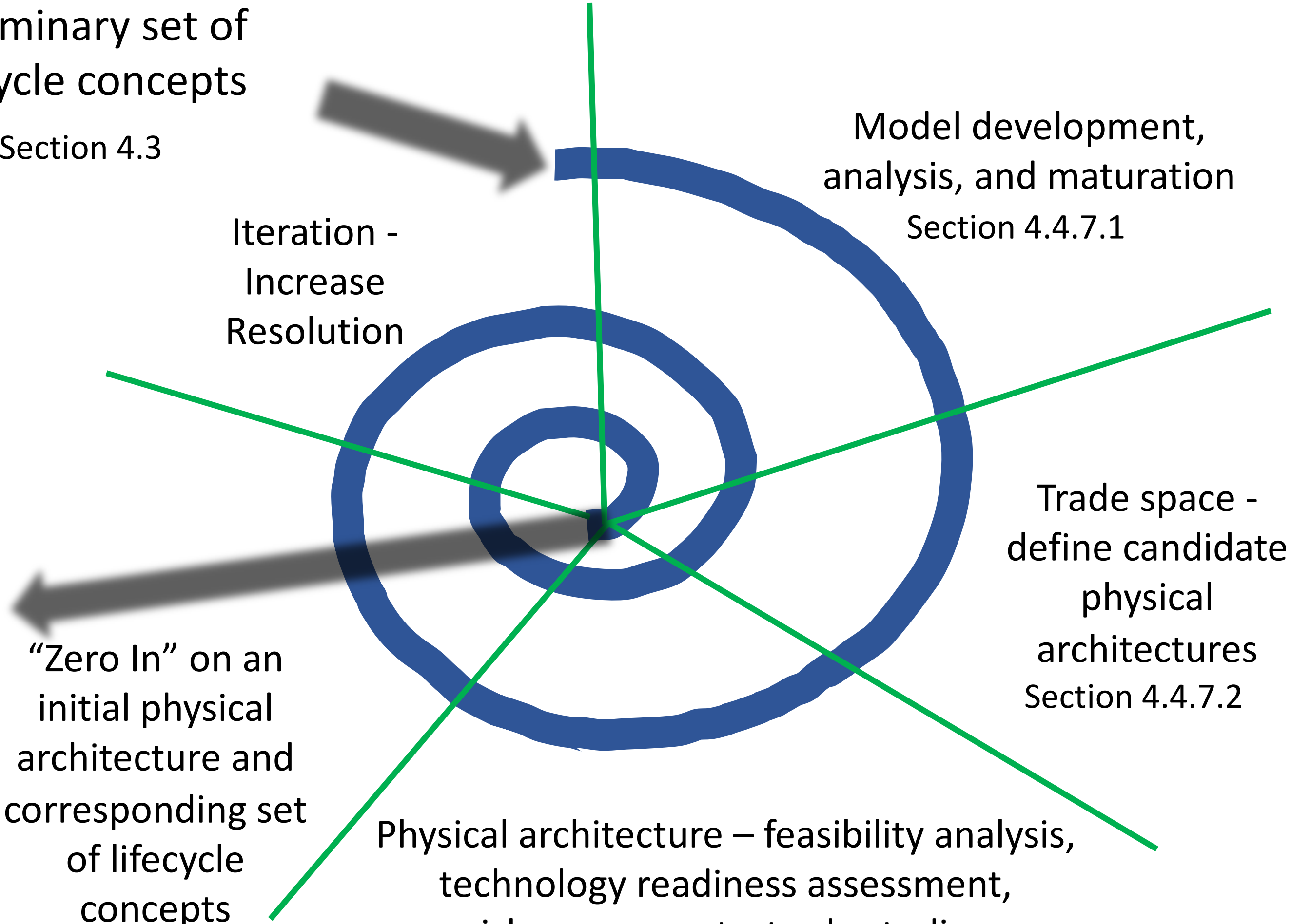


There is a lot of work to be done BEFORE defining requirements

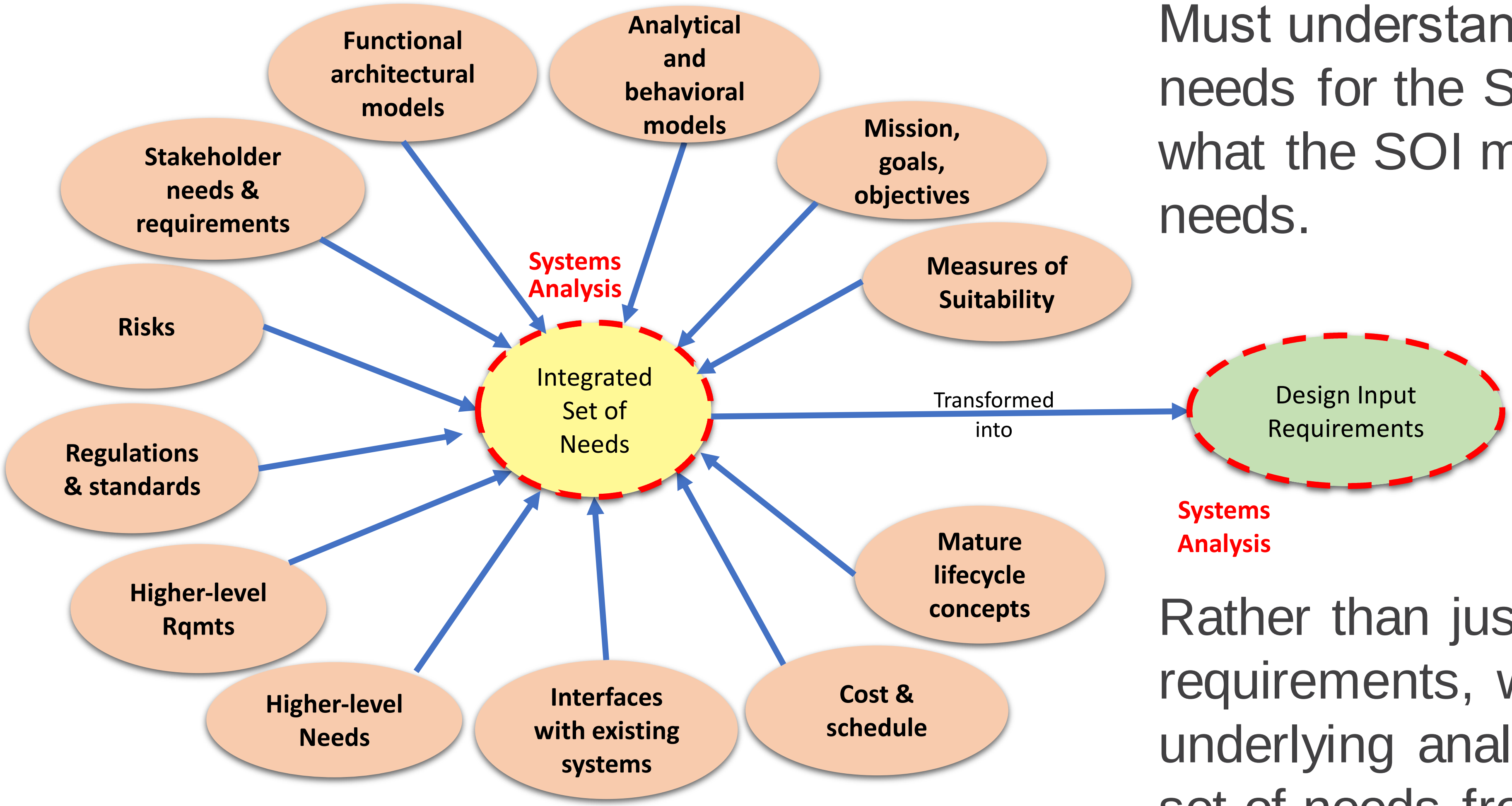
Establish completeness, consistency, correctness, and feasibility before defining needs and transforming them into the design input requirements

Lifecycle Concept Analysis and Maturation

Iterative set of activities zeroing in on a feasible set of lifecycle concepts <u>from which the needs will be derived</u>



Needs Before Requirements

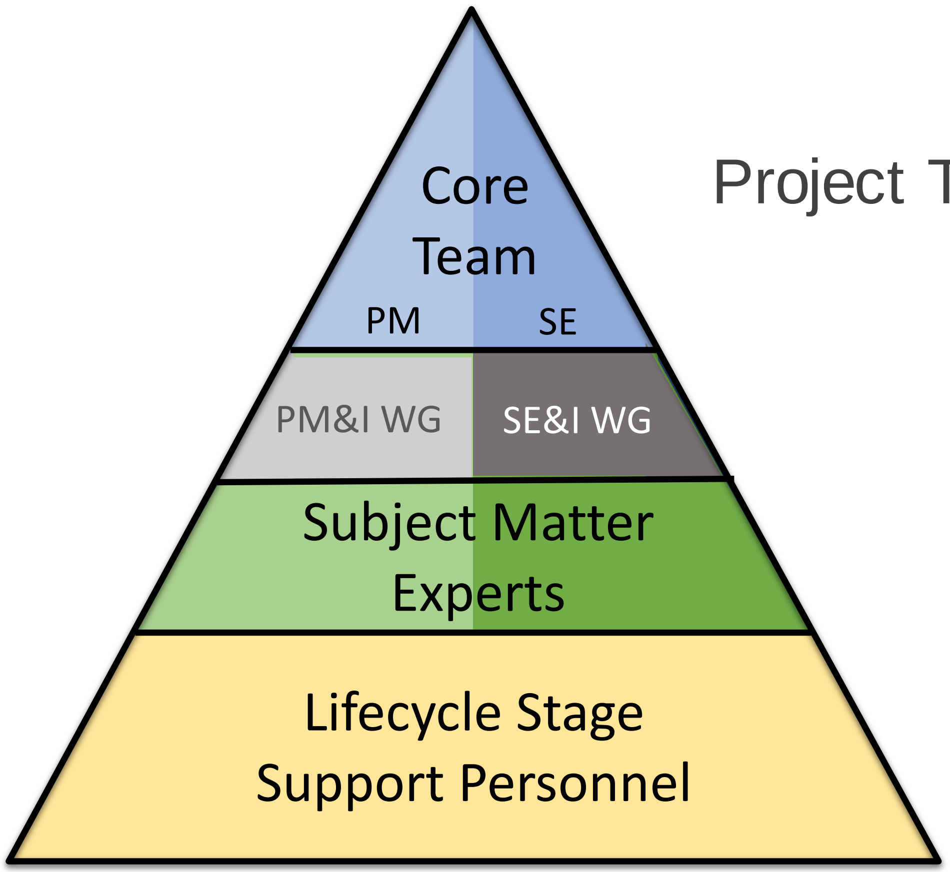


Must understand the stakeholder's needs for the SOI before defining what the SOI must do to meet those needs.

Rather than just having a set of requirements, we also have the underlying analysis and integrated set of needs from which they were transformed.

Integrated, collaborative, multidisciplined project team – minimize silos!

Integrated, collaborative, multidisciplined project team – minimize silos!



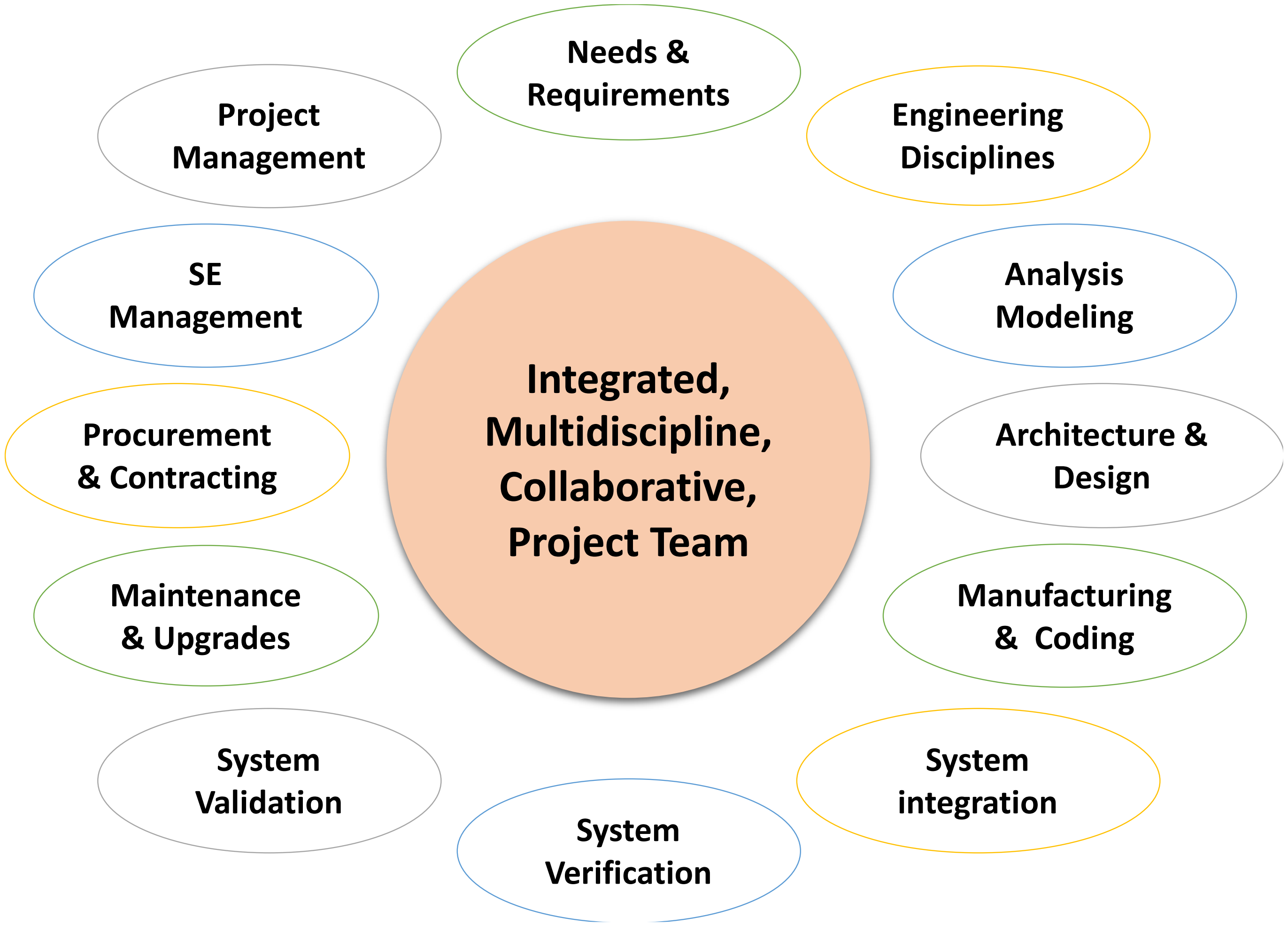
Project Team Organization

Project Management (PM) and Systems Engineering (SE) are two sides of the same coin

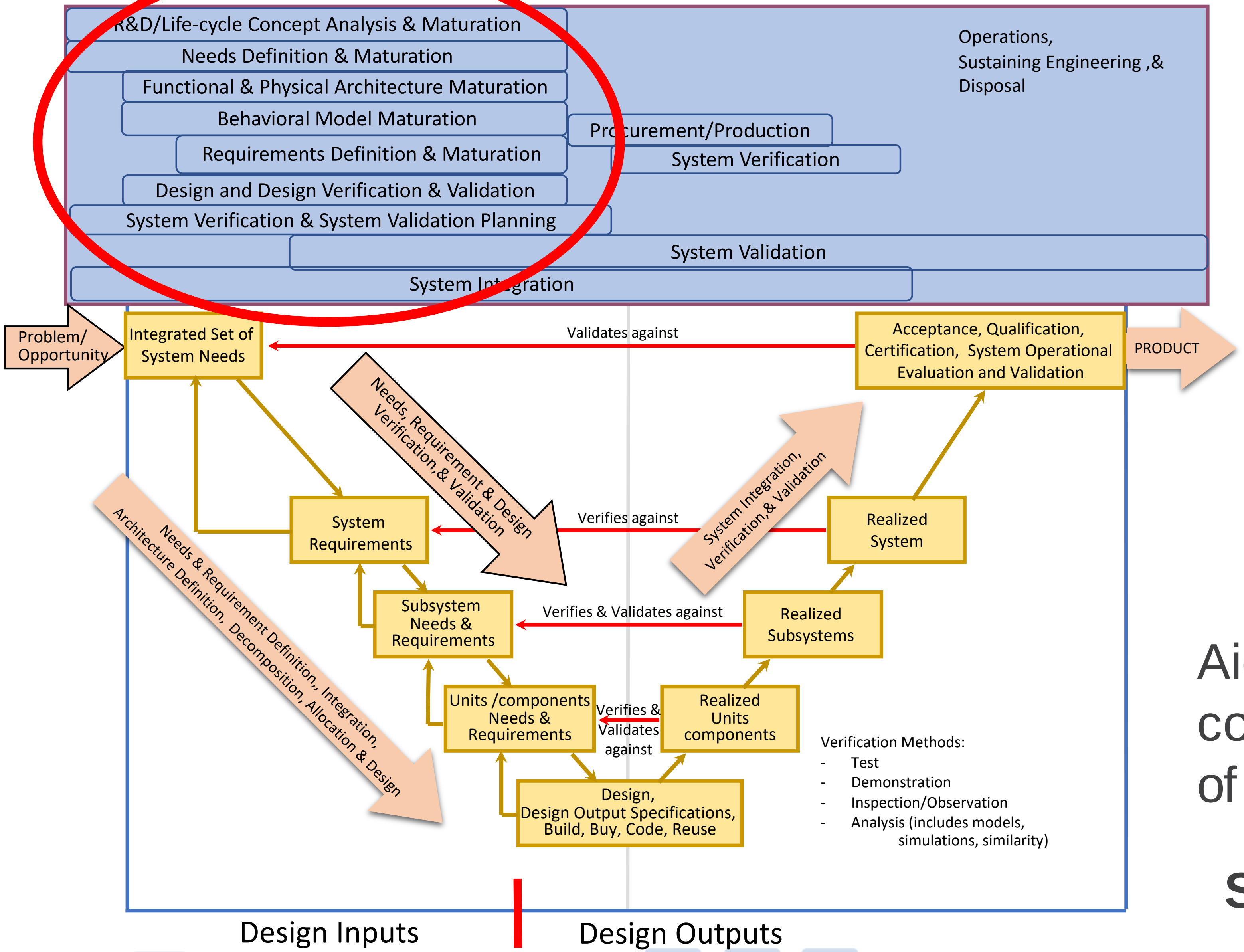
Integrated, Multidisciplined, Collaborative, Project team – minimize silos!

The team is made up of both PM and SE personnel as both are tightly dependent

This can be challenging when outsourcing development to a supplier



SE Processes are Intended to be Practiced Concurrently



Concurrent Definition, Analysis, Maturation applied iteratively and recursively as we move down the left side of the SE Vee

Faster and cheaper than classical waterfall/serial, document-centric processes with silos

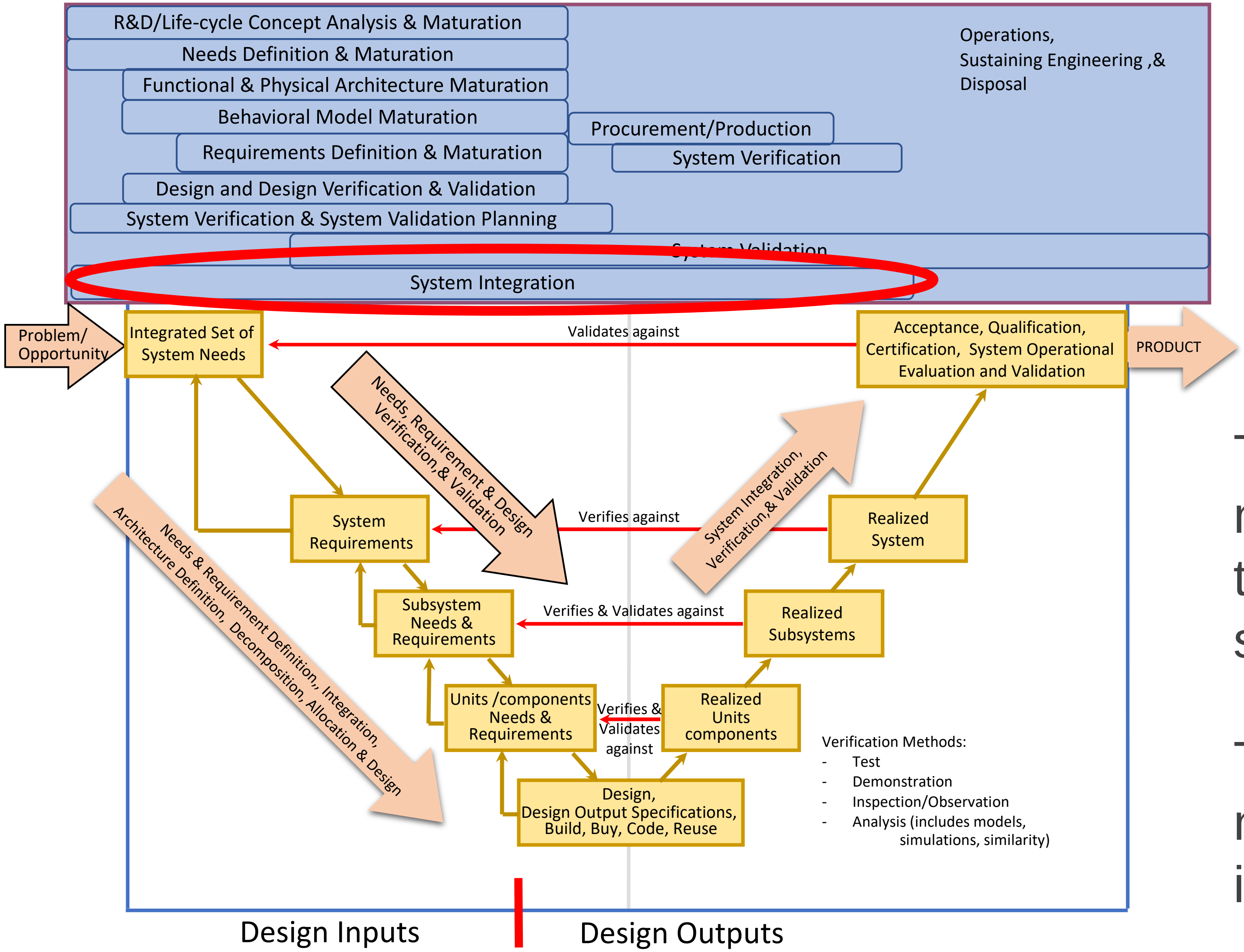
Aids in establishing correctness, completeness, and consistency, of all SE artifacts

Single Source of Truth (SSoT)

Adapted from Ryan, M.J.; Wheatcraft, L.S., “On the Use of the Terms Verification and Validation”, February 2017 and INCOSE SE HB, Version 4, Figures 4.15 & 4.19

Managing the Integrated System From the Beginning

System Integration Across the Lifecycle



System Integration should not be thought of as only occurring on the right side of the SE Vee during IV&V

The integrated SOI must be managed from the beginning of the project across all lifecycle stages

The customer/supplier relationship must allow management of the integrated SOI

Adapted from Ryan, M. J.; Wheatcraft, L.S., "On the Use of the Terms Verification and Validation", February 2017 and INCOSE SE HB, Version 4, Figures 4.15 & 4.19

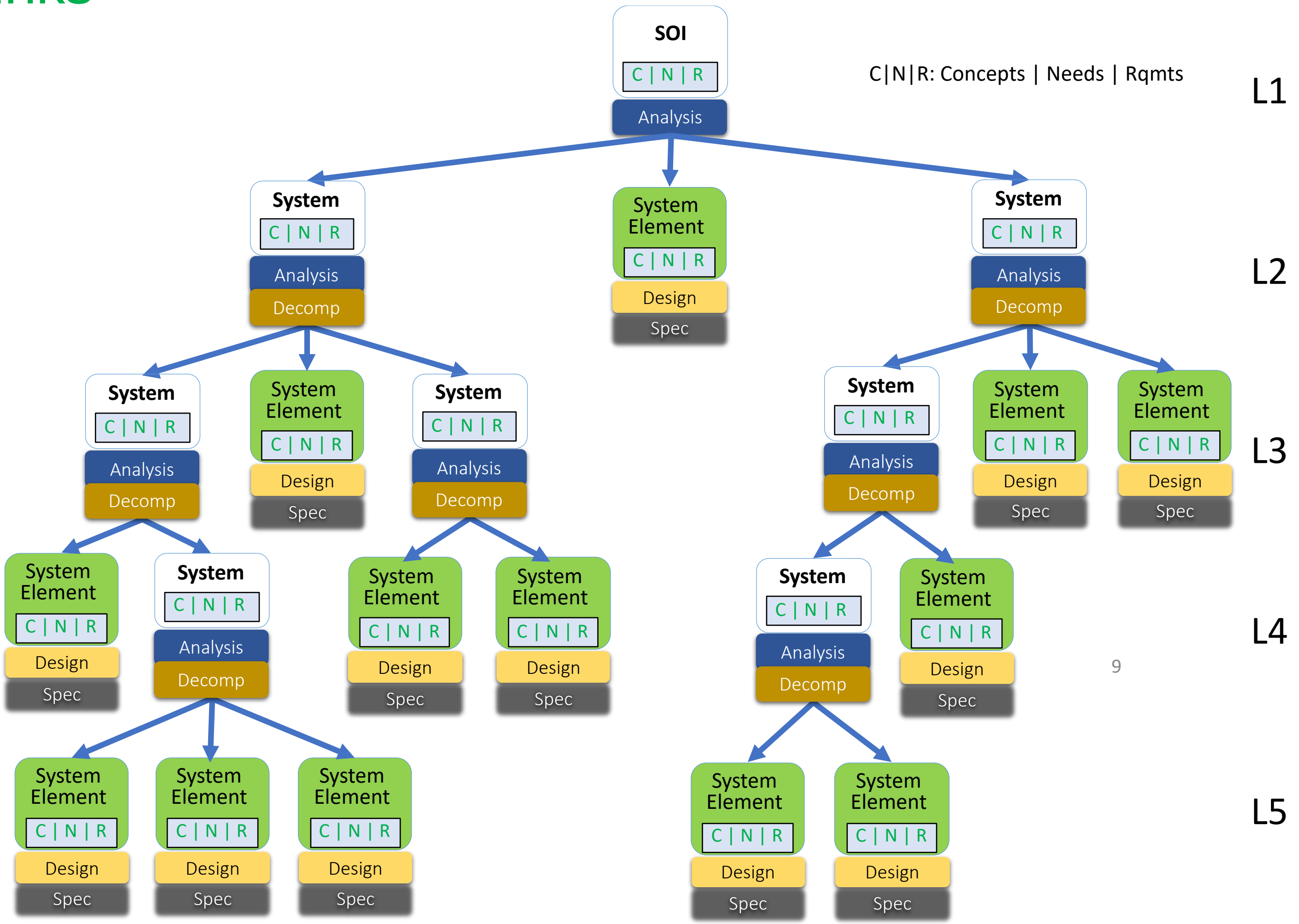
Levels of a System – Hierarchical View

Focus on decomposition makes it easier to develop the SOI in bite sized chunks across multiple organizational units (internal and external) based on specialize knowledge and expertise

Interactions (Interfaces) not shown in this view

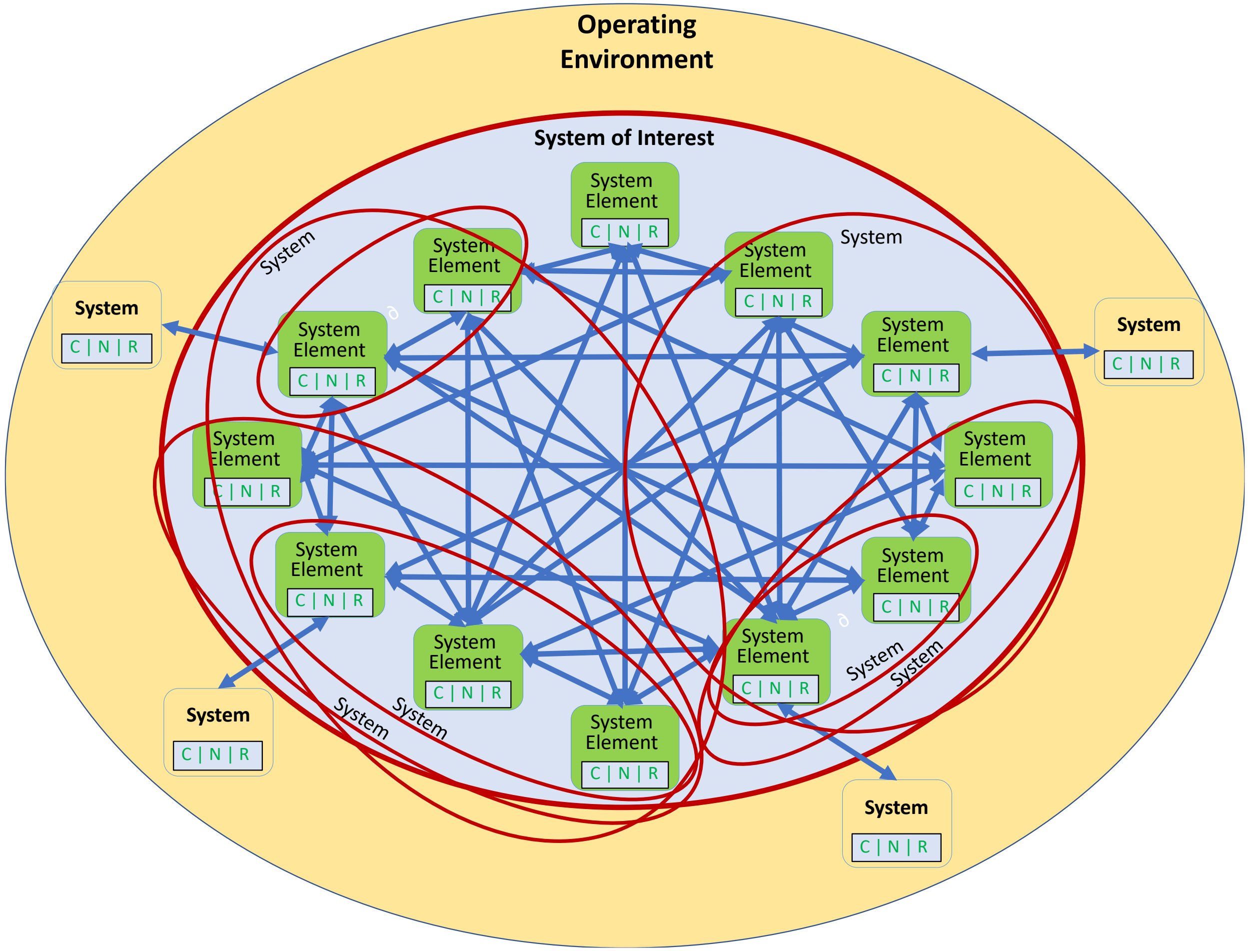
Focus tends to be more on the systems and system elements that make up the SOI than the integrated SOI

Leads to development in silos and system and system element optimization rather than optimization of the integrated SOI



Holistic View of the Integrated SOI

A system is greater than the sum of its parts.



Focus on behavior and emerging properties of the integrated SOI based on interactions between the systems & system elements that make up the SOI, as well as interactions with external systems & the operational environment.

To optimize the integrated SOI, systems and system elements within its architecture may not be able to be optimized.

Example Integrated System Model

- The Boeing 787 is an example of a large-scale system whose system life cycle process activities are distributed across many organizations and locations.
- Over 30 companies based in countries around the world built large portions of the airplane.
- To help manage this complex system, Boeing developed an integrated model that had >2,000 functions, >5,000 data flows, >1,000,000 data parameters, and >50,000,000 objects, with an average of three relationships per object, as well as ~1,000 geographically dispersed users involved in the modeling effort.

Malone, R., Herrord, J., Friedland, B., Fogarty, D., 2016. “Paper Insights from Large Scale Model Based Systems Engineering at Boeing”. 26th Annual INCOSE International Symposium (IS 2016), Edinburgh, Scotland, UK, July 18 - 21, 2016

Allocation and budgeting for software-centric systems

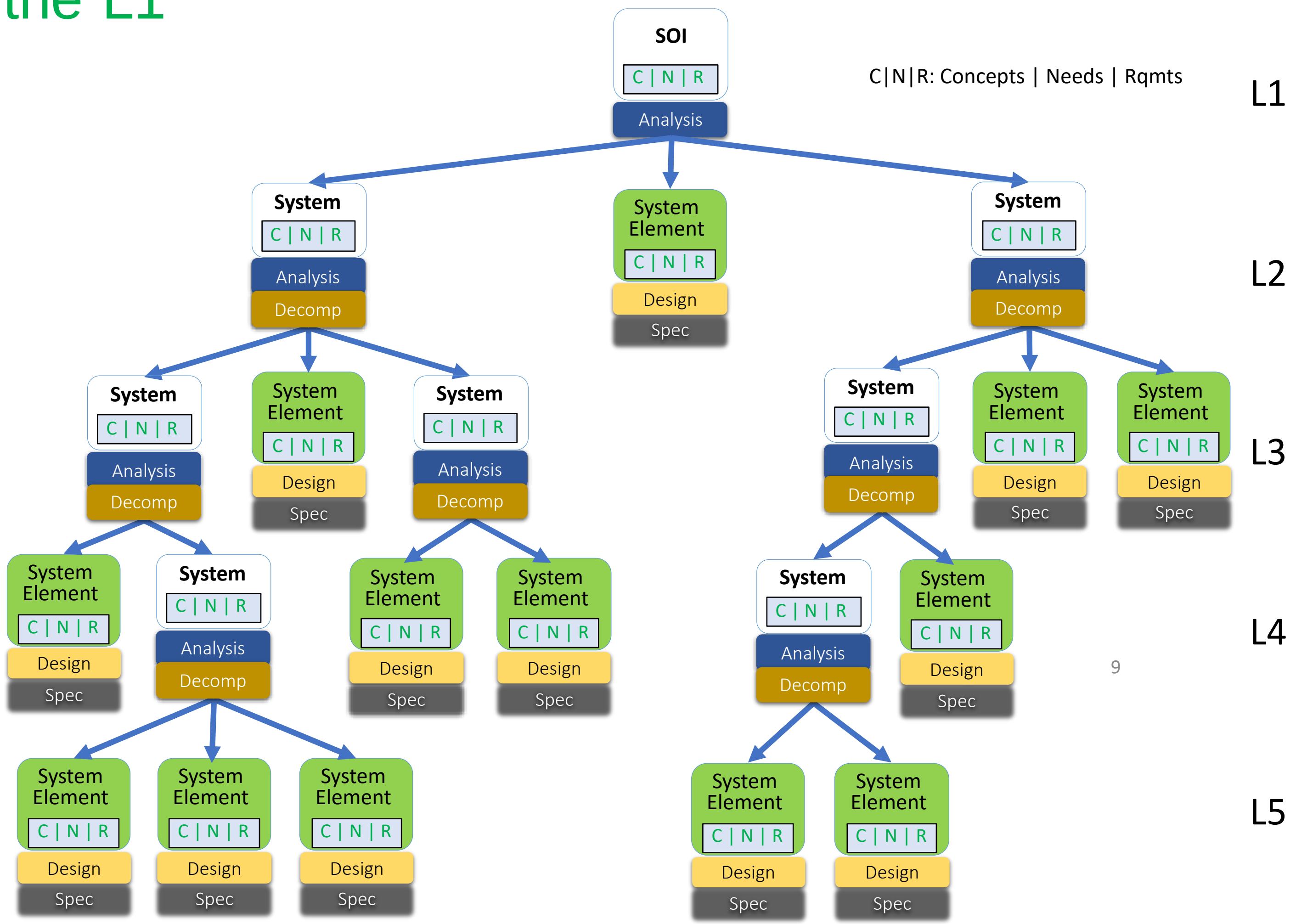
Levels of a System – Hierarchical View

Classical flow down (allocation and budgeting) of requirements is from the L1 SOI to subsystems at L2

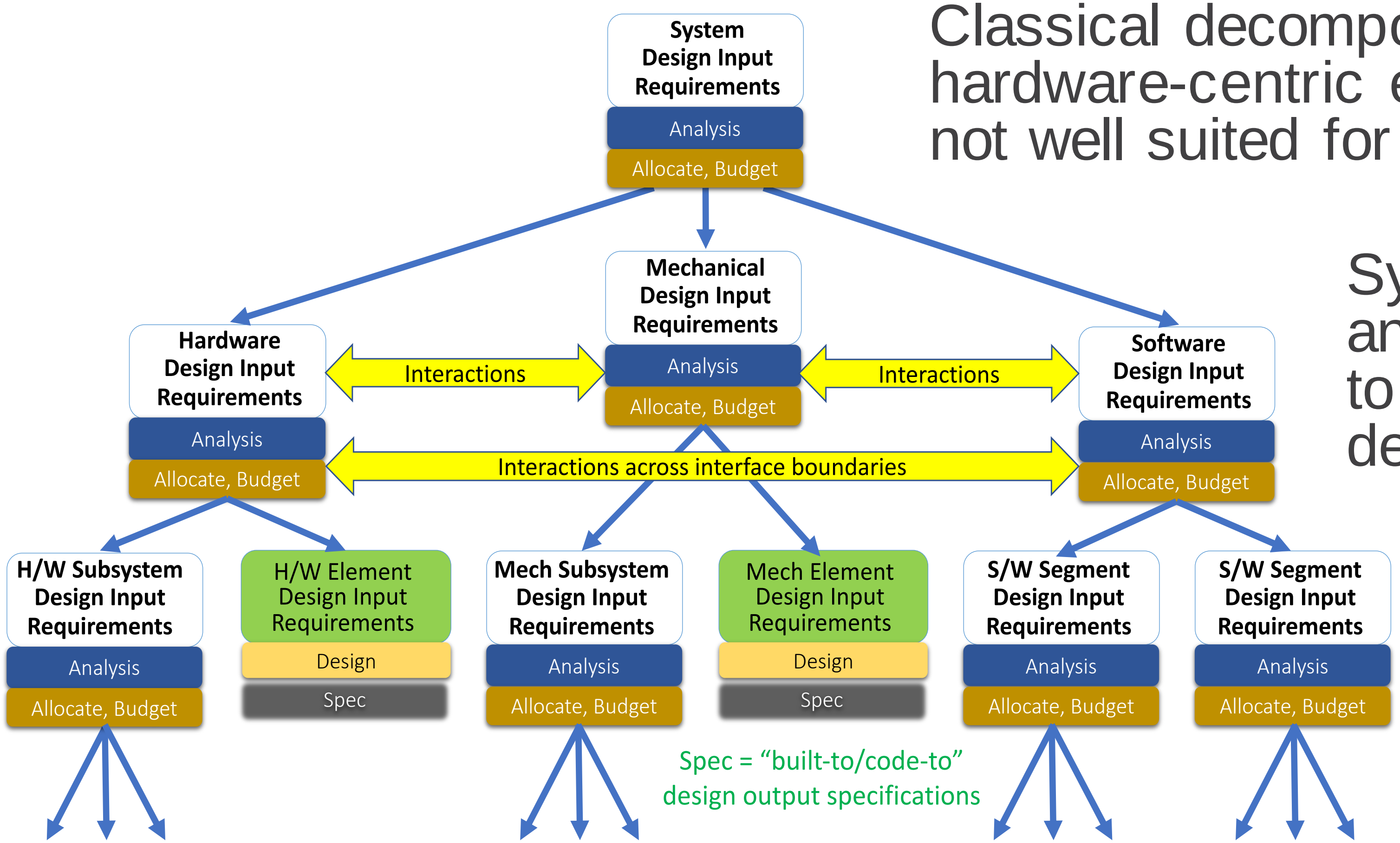
Then L2 to L3

It may be L4 or L5 until allocations are made to software

Leads to development of software in silos with little integration between different software system elements



Allocation and budgeting for software-centric systems



Classical decomposition and architecture for hardware-centric electro-mechanical systems not well suited for software-centric systems.

System level requirement allocation and budgeting to the software needs to be done at the first level of decomposition.

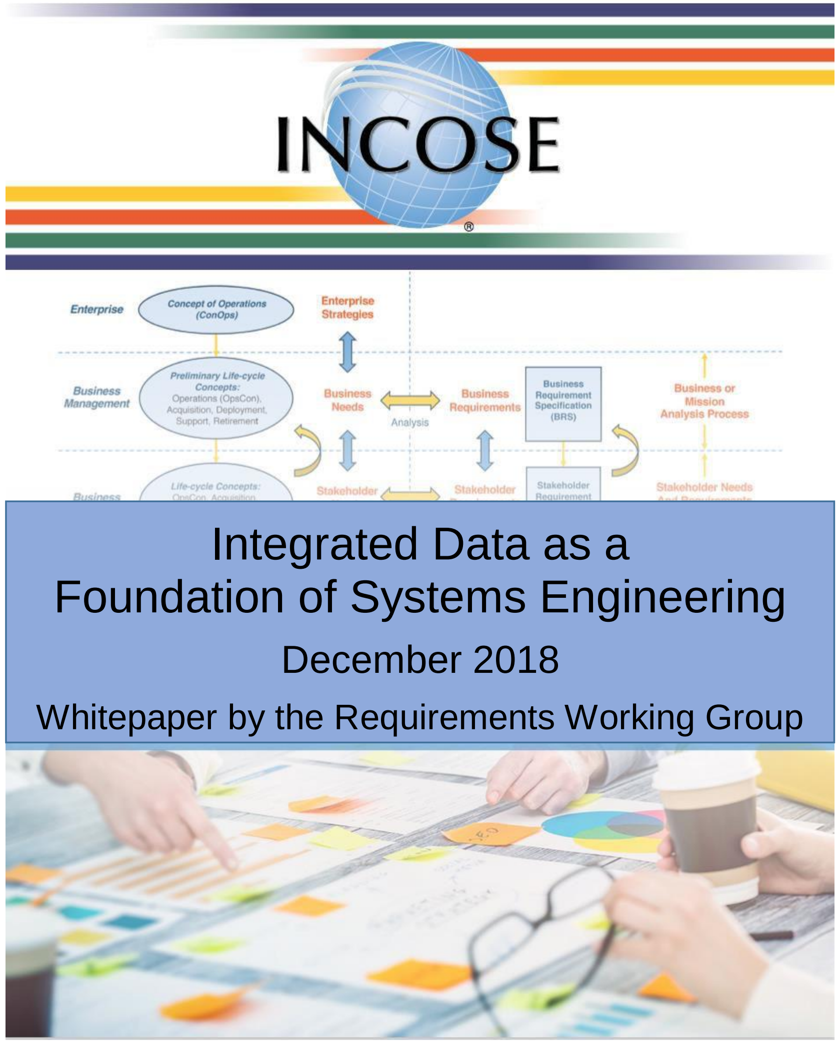
Critical functionality and performance are in the embedded software – the hardware are enabling systems.

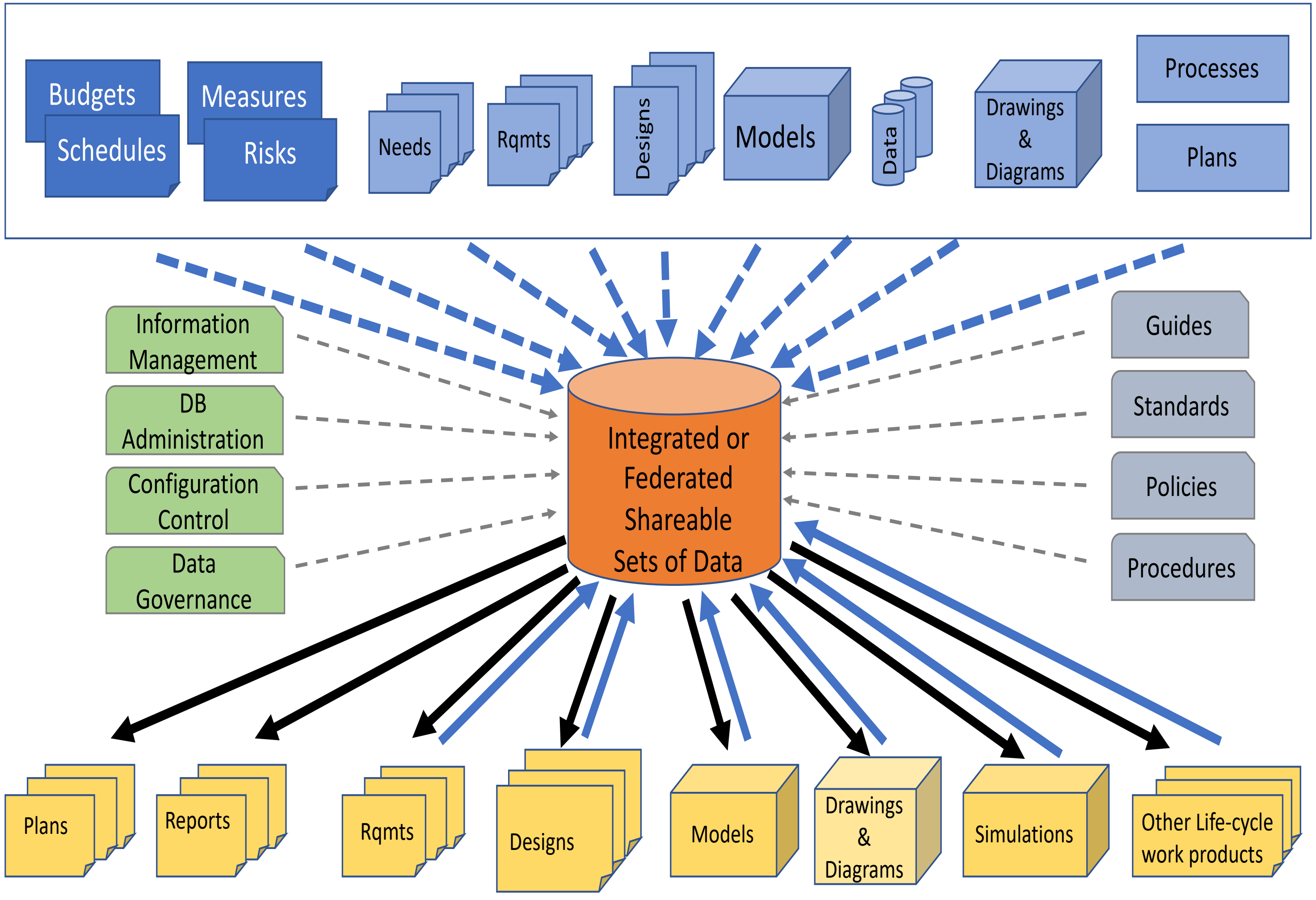
Hardware sensors, displays, actuators, processors, wiring, and communication busses constrain the software.

Enables software to be developed and managed as an integrated system within the SOI architecture.

Data-Centric practice of Systems Engineering

Data-Centric practice of Systems Engineering

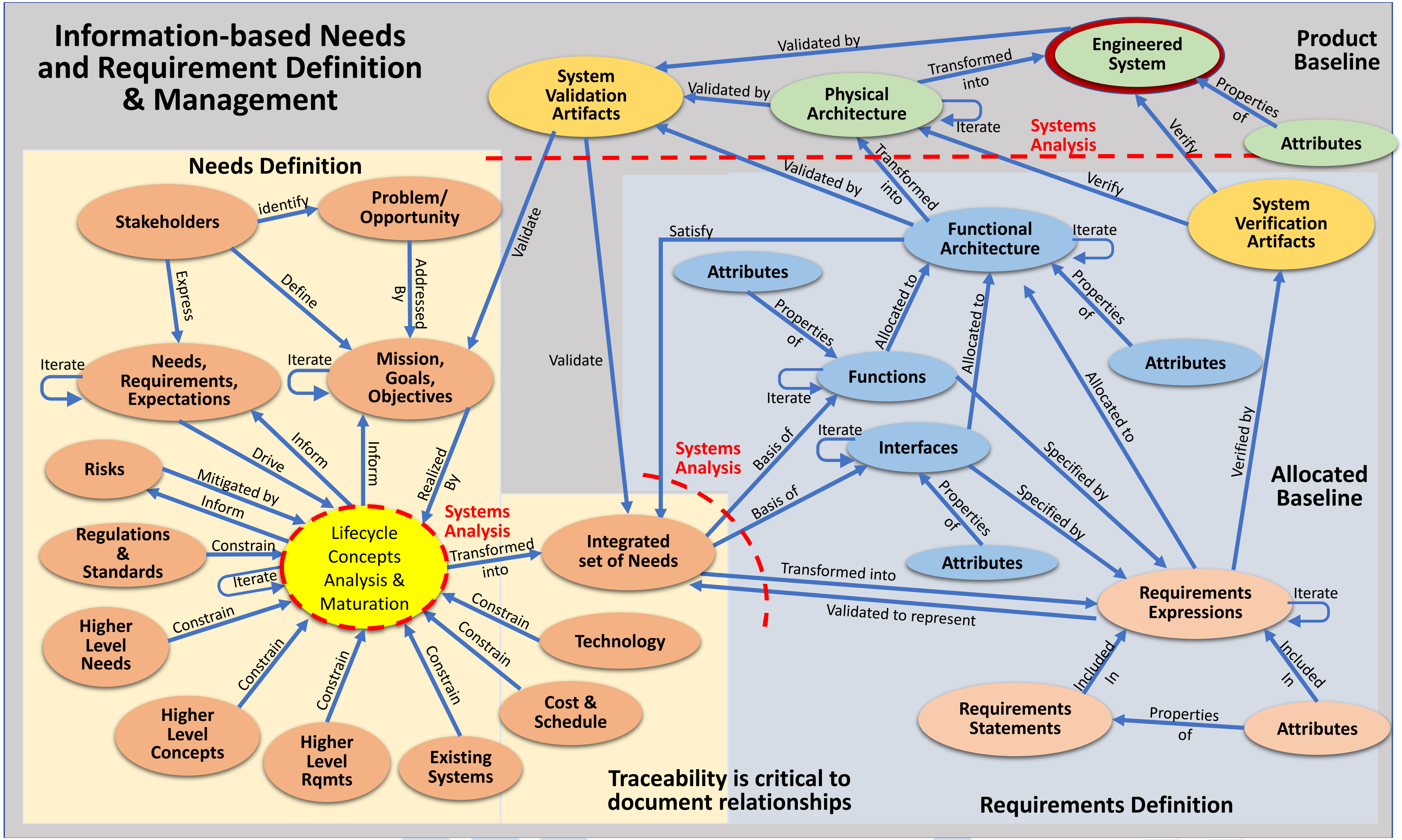




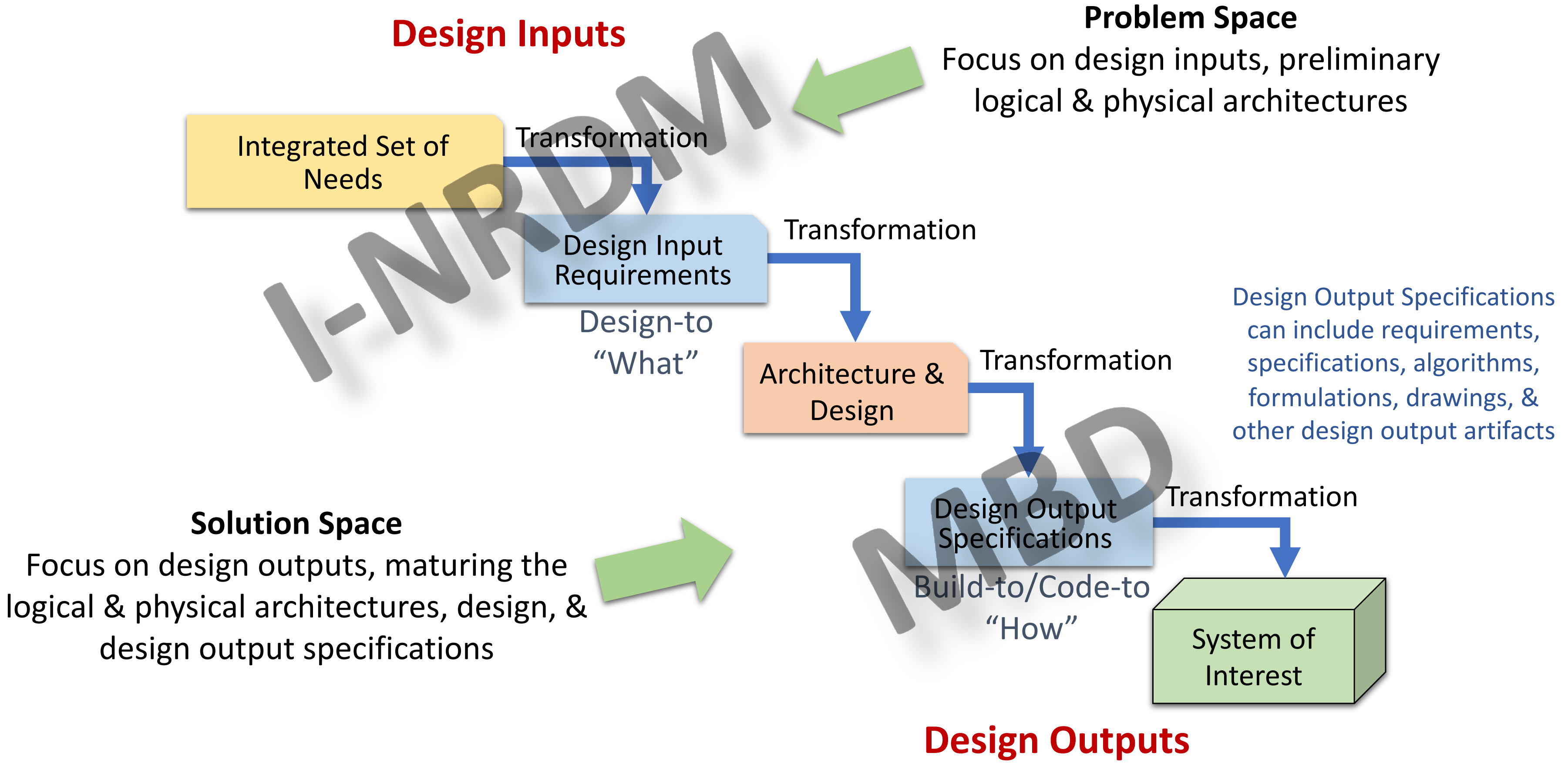
Original Developed by INCOSE RWG at IW 2017

Single Source of Truth (SSoT)

Data-Centric practice of Systems Engineering



Data-Centric practice of Systems Engineering

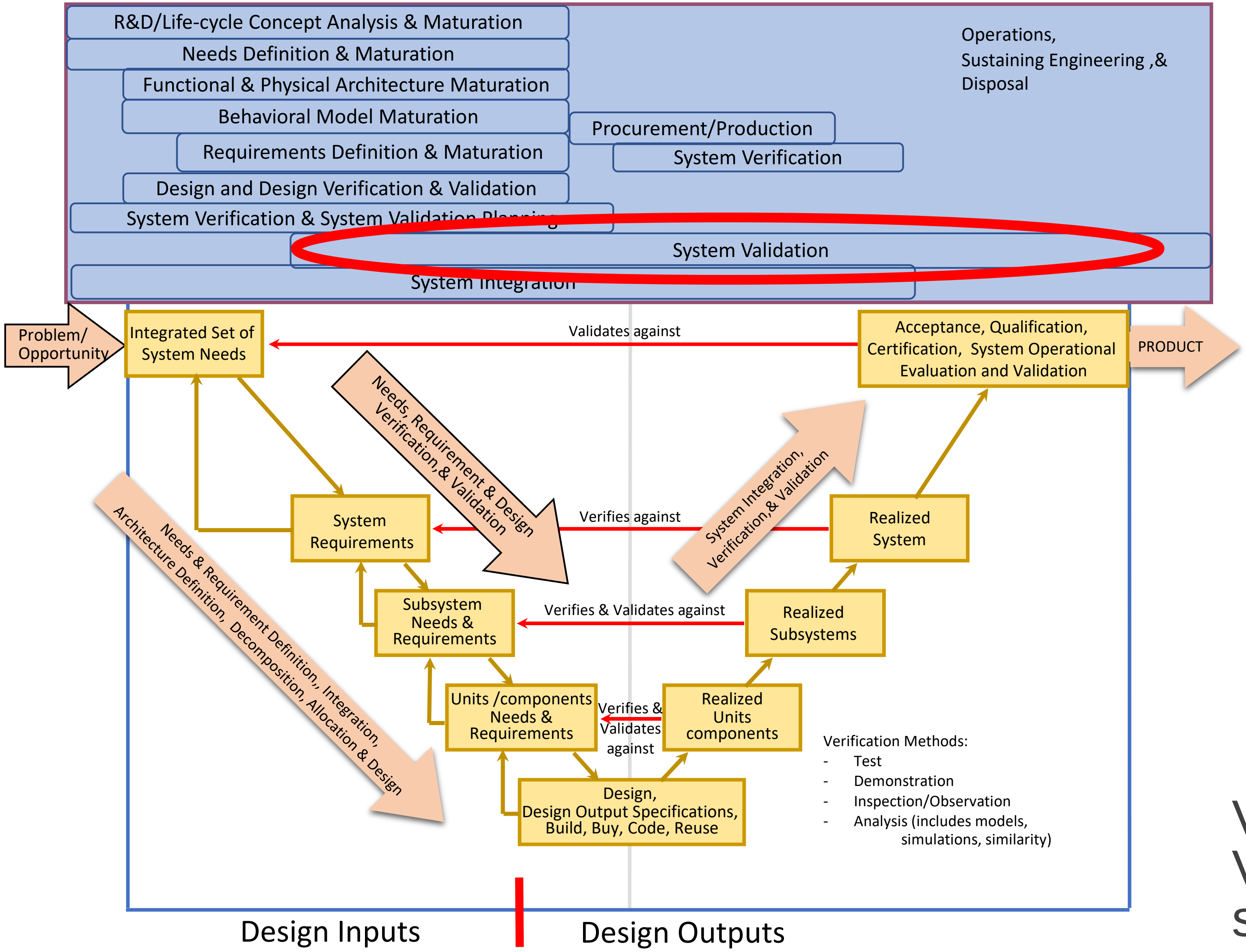


I-NRDM + MBD = MBSE

I-NRDM: Information-based Needs and Requirements Definition and Management MBD: Model-based Design
MBSE: Model-based Systems Engineering

Increased Focus on Validation across the lifecycle

Validation Across the Lifecycle



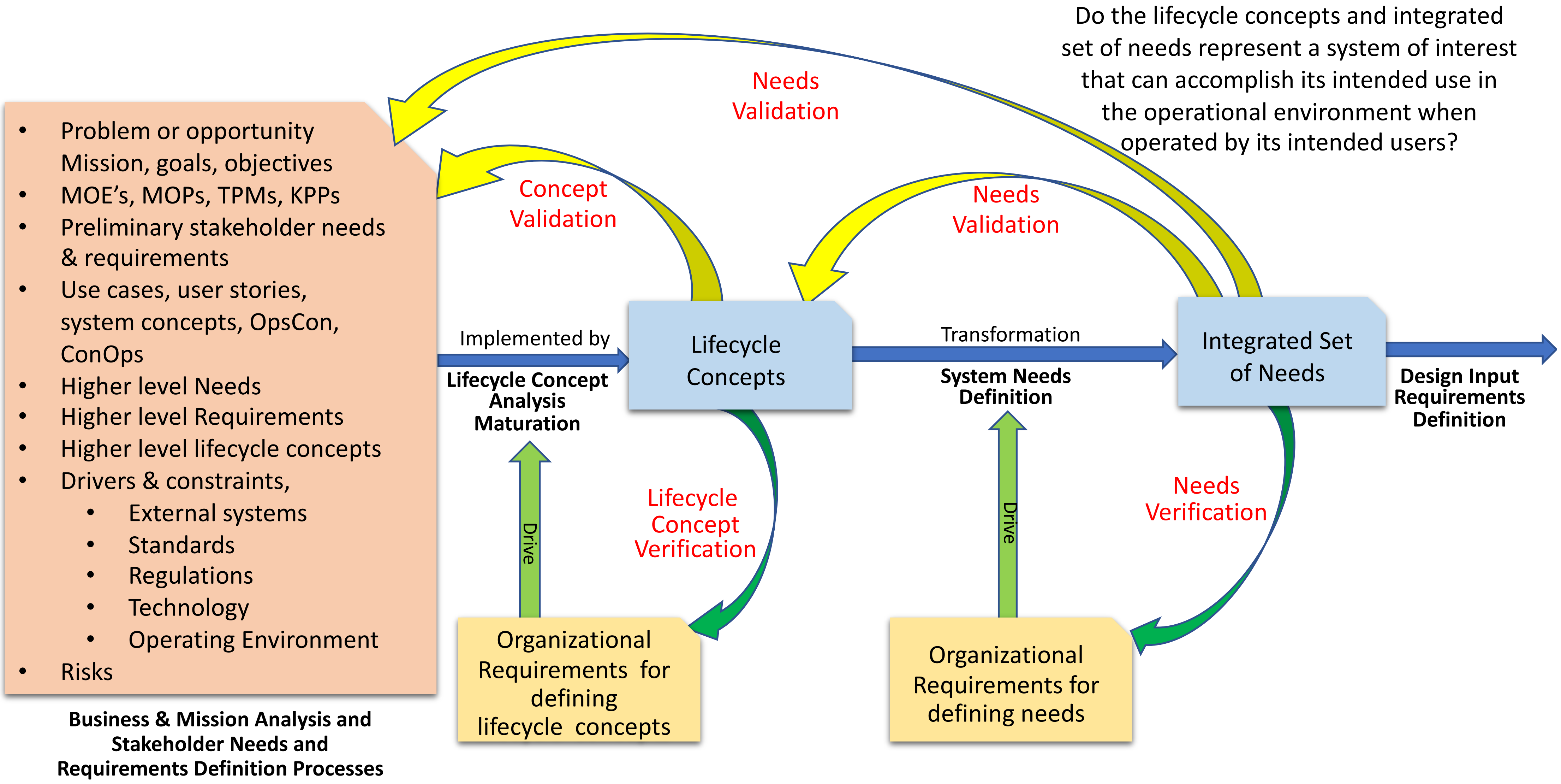
Adapted from Ryan, M. J.; Wheatcraft, L.S., "On the Use of the Terms Verification and Validation", February 2017 and INCOSE SE HB, Version 4, Figures 4.15 & 4.19

Validation should not be thought of as only occurring on the right side of the SE Vee during system integration, verification, and validation

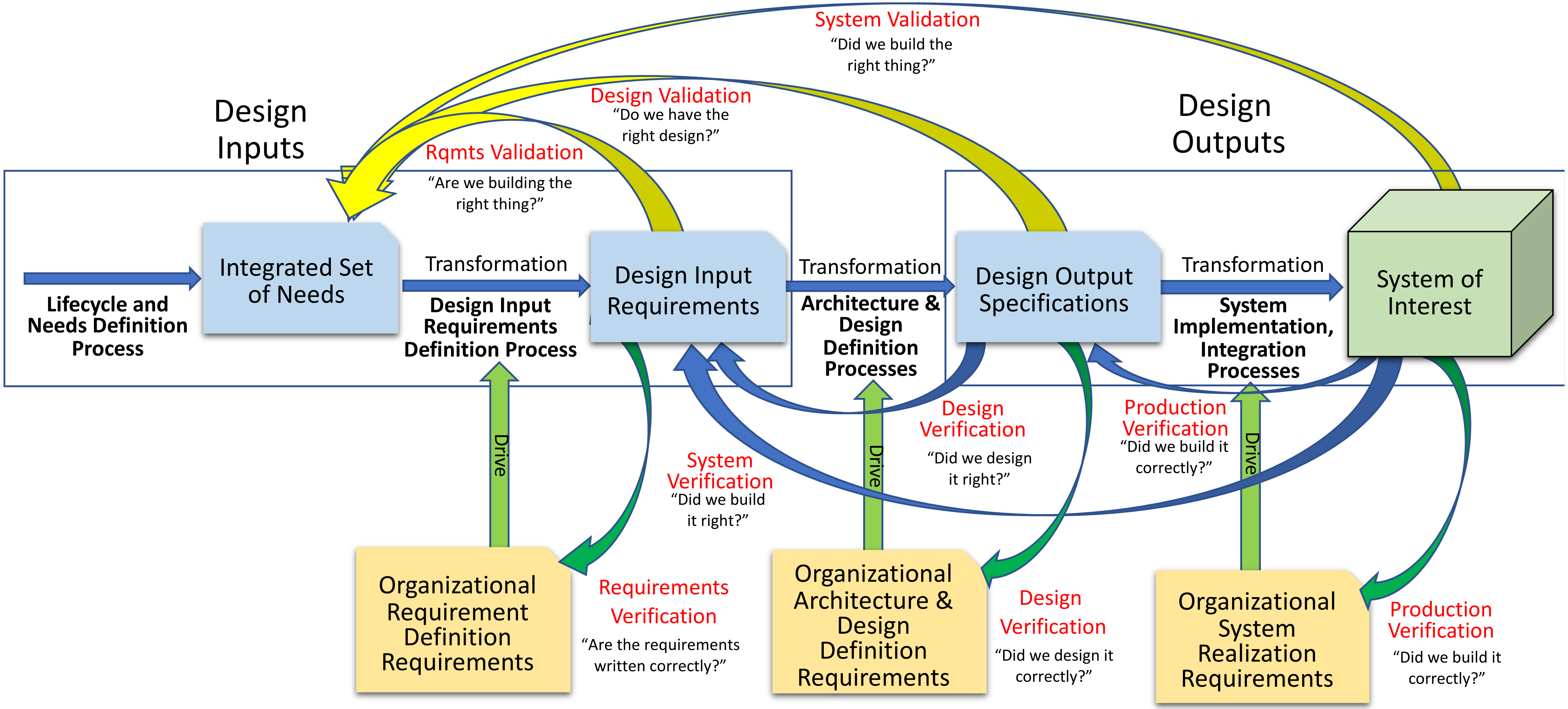
Validation is done across the system lifecycle:
- Needs
- Design Input Requirements
- Architecture
- Design
- Design Output Specifications
- Realized system elements, systems, and the integrated SOI

Validation on the left side of the SE Vee decreases issues on the right side of the SE Vee

Verification and Validation of the Needs



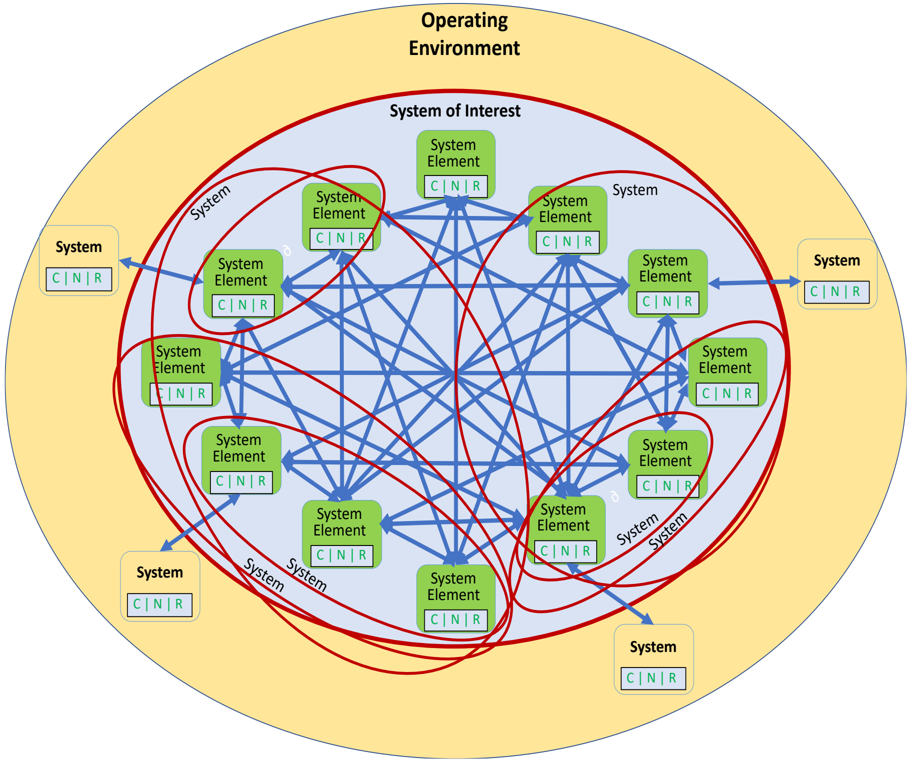
Verification & Validation in Context



Derived from Ryan, M. J.; Wheatcraft, L.S., "On the Use of the Terms Verification and Validation", February 2017

System Validation:

Validating that the realized physical, integrated SOI
meets its intended purpose AND
identify and assess its behavior and emerging properties
in its actual operational environment
when operated by its actual intended users
and does not enable unintended users to negatively impact the intended use of the system nor allow unintended users to use the system in an unintended way



System Validation

What's more important?
System Verification or System Validation?????

Project success is based on the SOI passing system validation

Passing system verification but failing system validation results in a failed project…….

What do you validate the system against?
Where is it defined?
Who defines it?
Who is responsible?
What is "Necessary for Acceptance?

This must be made clear in all customer/supplier agreements

Questions and Discussion