Needs and Requirements Manual (NRM)
Section 6: Design Input Requirement Definition
Section 7: Design Input Requirement Verification and Validation

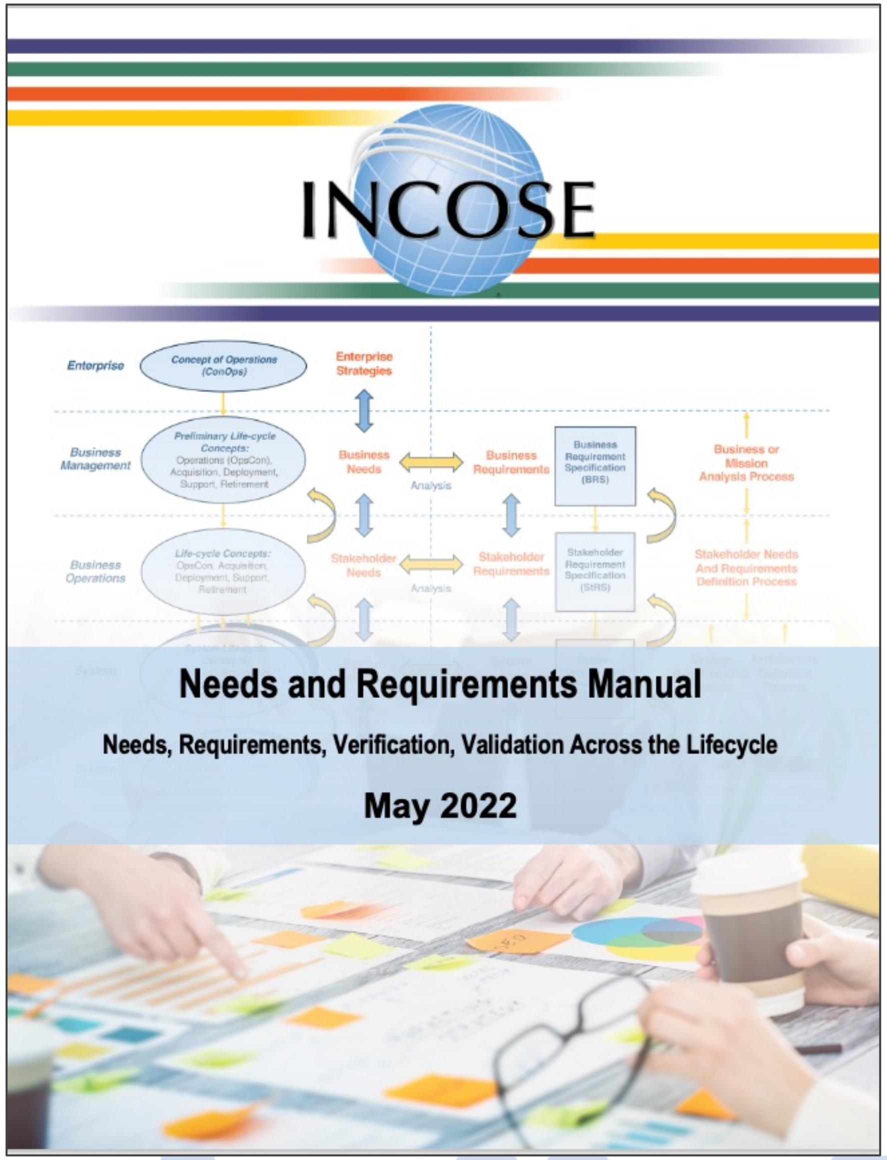
Requirements Working Group

Lou Wheatcraft, Senior Consultant, Wheatland Consulting

Wheatland.consulting@gmail.com

Needs and Requirements Manual (NRM)

- The NRM V1.1 released in May 2022 to shorten title, add subtitle, and align with other RWG products
- Content from this aligns with, and expands, the INCOSE SE Handbook version 5 material.
- Will be updating to V2.0 for publication in 2024.

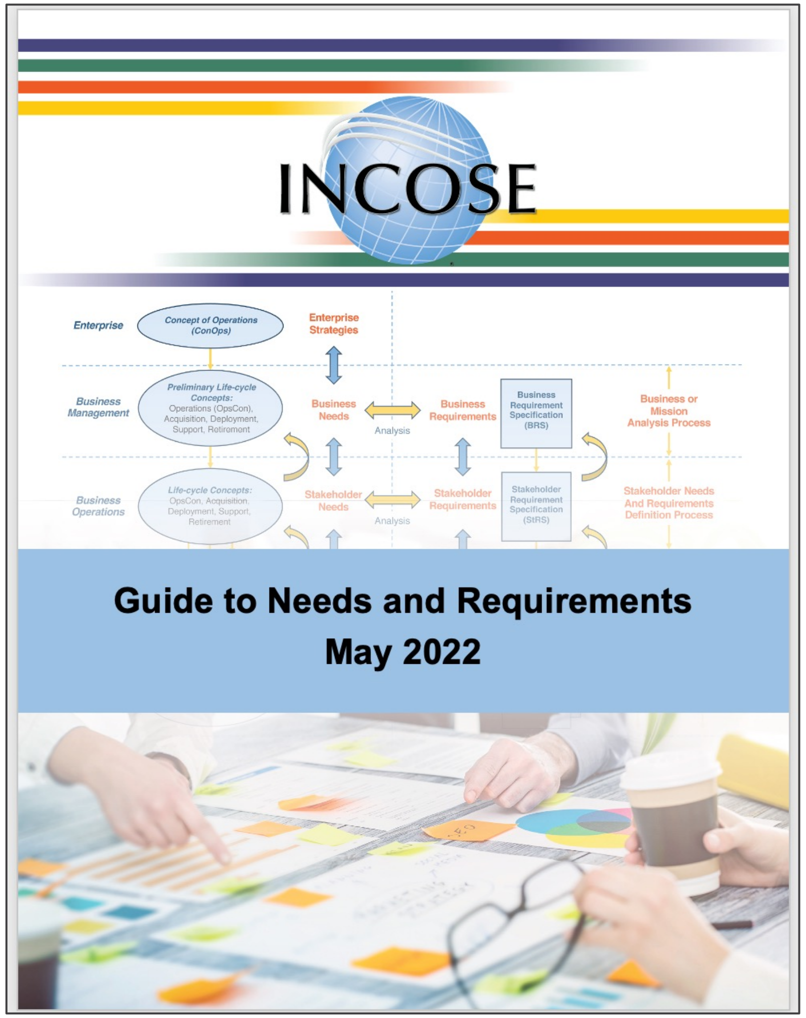

INCOSE

Needs and Requirements Manual

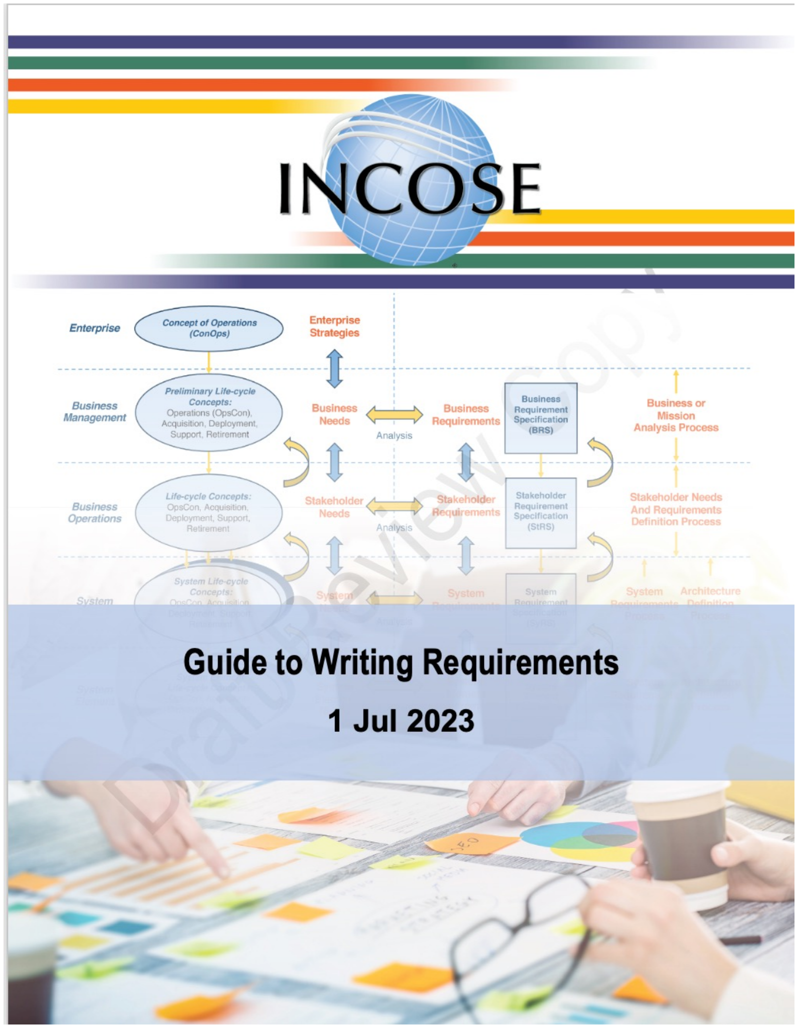
Needs, Requirements, Verification, Validation Across the Lifecycle

May 2022

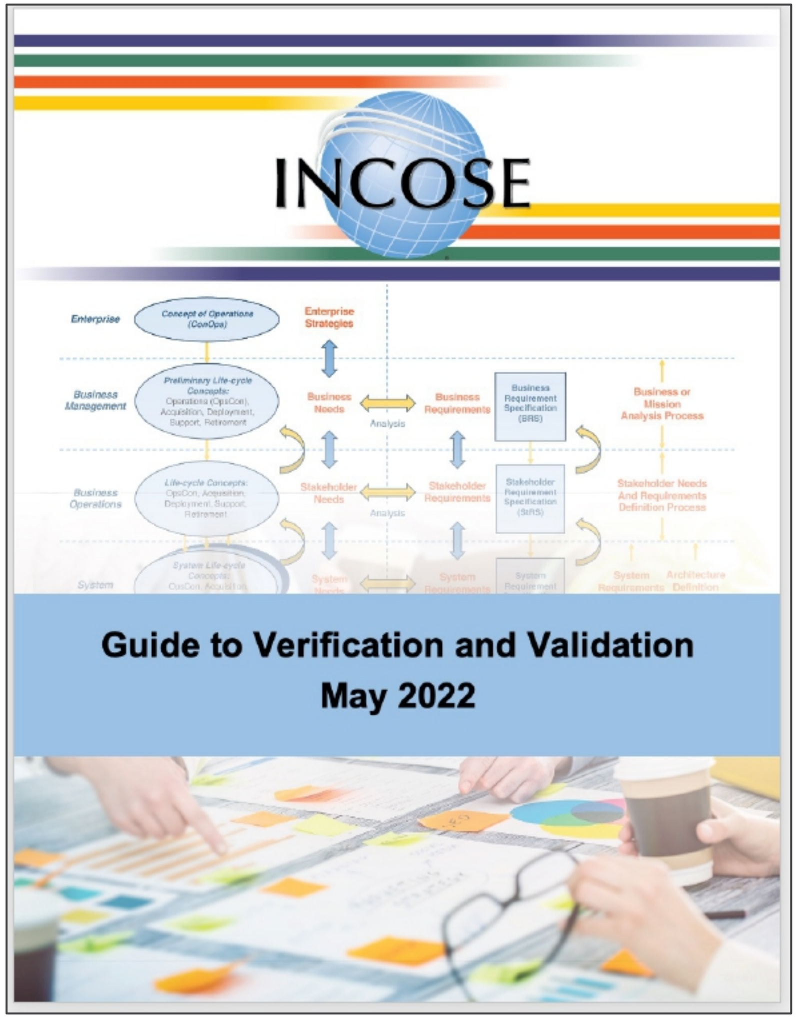
The RWG Guides Provide Practical Application of the NRM


INCOSE

Guide to Needs and Requirements
May 2022

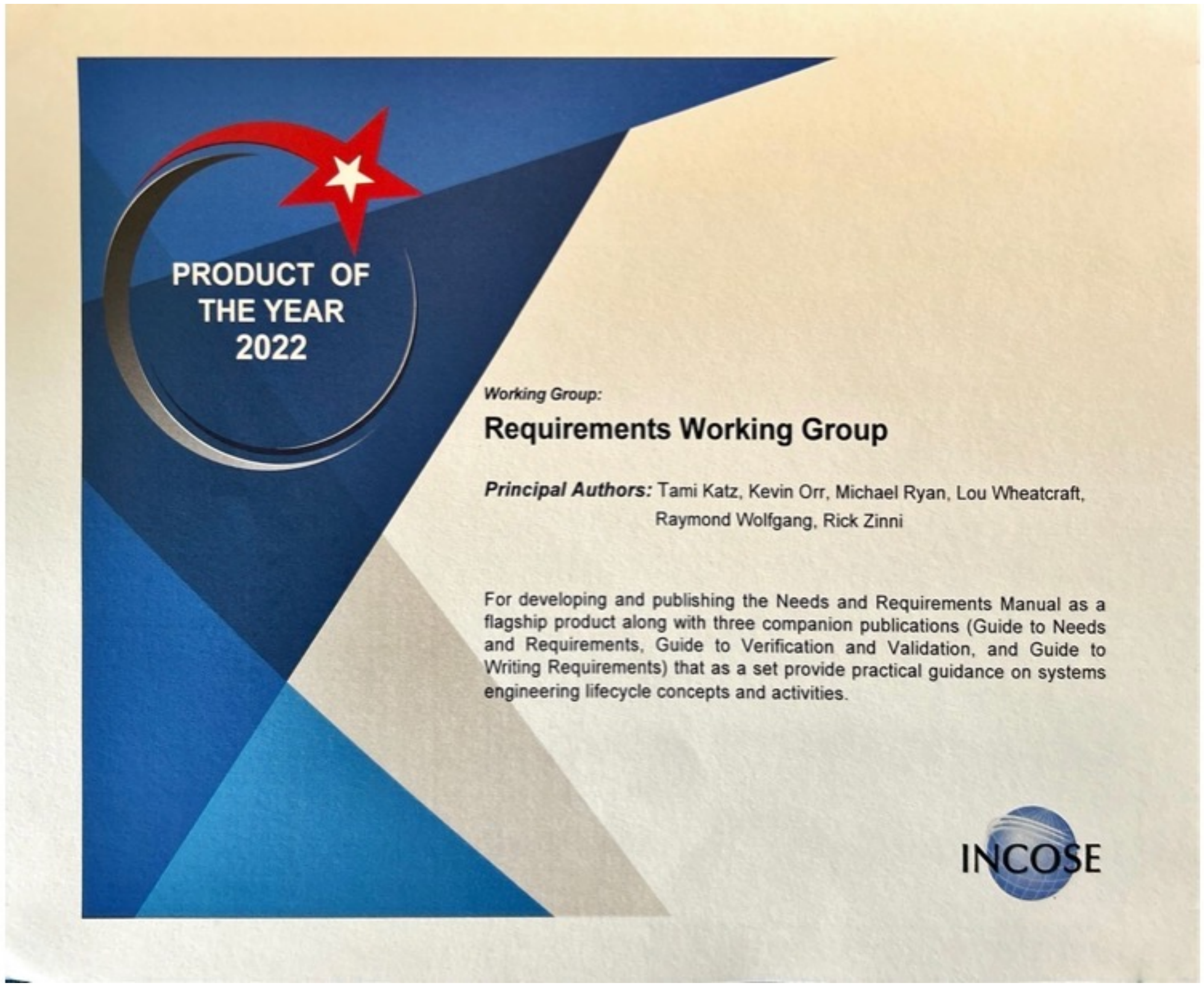

INCOSE

Guide to Writing Requirements
1 Jul 2023


INCOSE

Guide to Verification and Validation
May 2022

RWG Award at IE2023



PRODUCT OF THE YEAR 2022

Working Group:

Requirements Working Group

Principal Authors: Tami Katz, Kevin Orr, Michael Ryan, Lou Wheatcraft, Raymond Wolfgang, Rick Zinni

For developing and publishing the Needs and Requirements Manual as a flagship product along with three companion publications (Guide to Needs and Requirements, Guide to Verification and Validation, and Guide to Writing Requirements) that as a set provide practical guidance on systems engineering lifecycle concepts and activities.

INCOSE

NRM

- Supplements and elaborates on the INCOSE SE HB v5
  - Provides more detailed guidance on the what, how, and why concerning needs and requirement definition and management, verification, and validation (NRVV) definition and management across the system lifecycle. .
  - Addresses ambiguity and inconsistencies in ontology concerning needs and requirements definition and management, verification, and validation.
- To successfully complete system verification and system validation, the needs and requirements of the system as well as the system verification and system validation artifacts must be managed throughout the entire system lifecycle.
- **The NRM provides practical guidance on the activities required to achieve those outcomes**.

"Needs, Requirements, Verification, and Validation are common threads that tie all lifecycle activities and processes together." Lou Wheatcraft

NRM TOC

Focus of this presentation

Focus of this Presentation

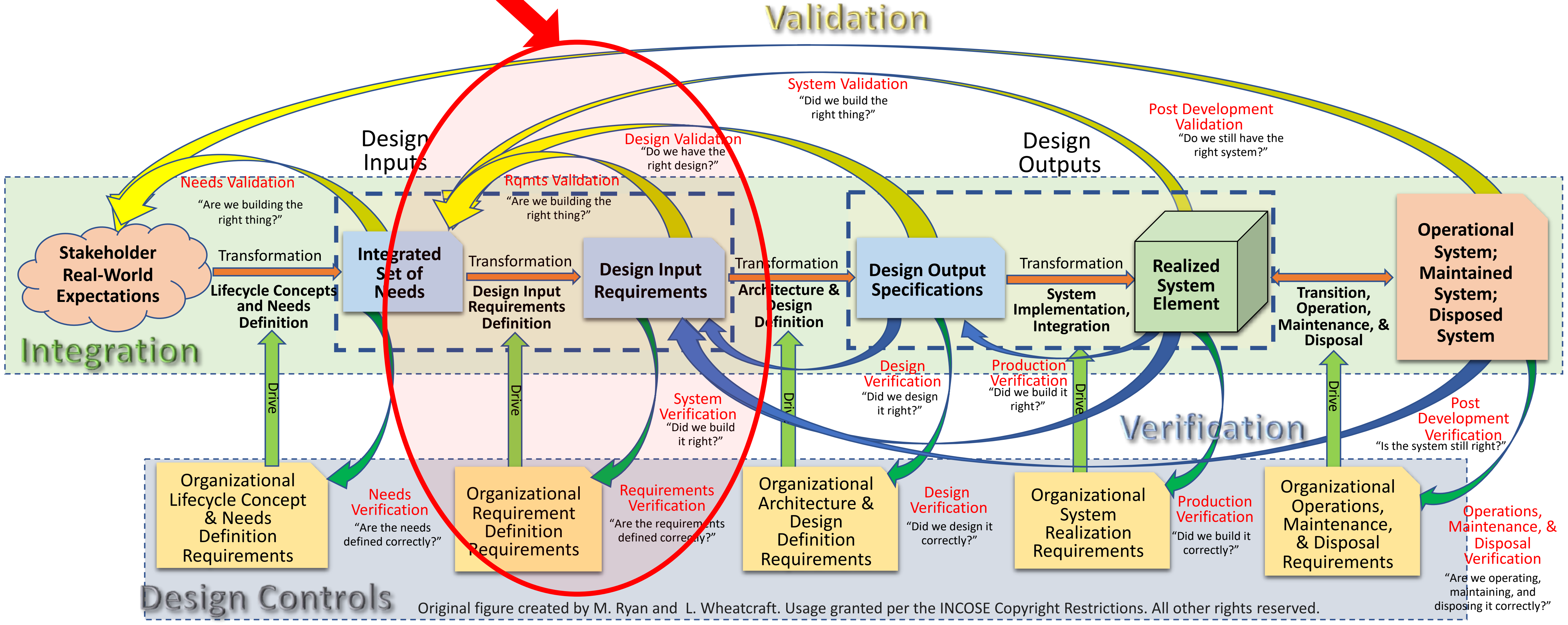
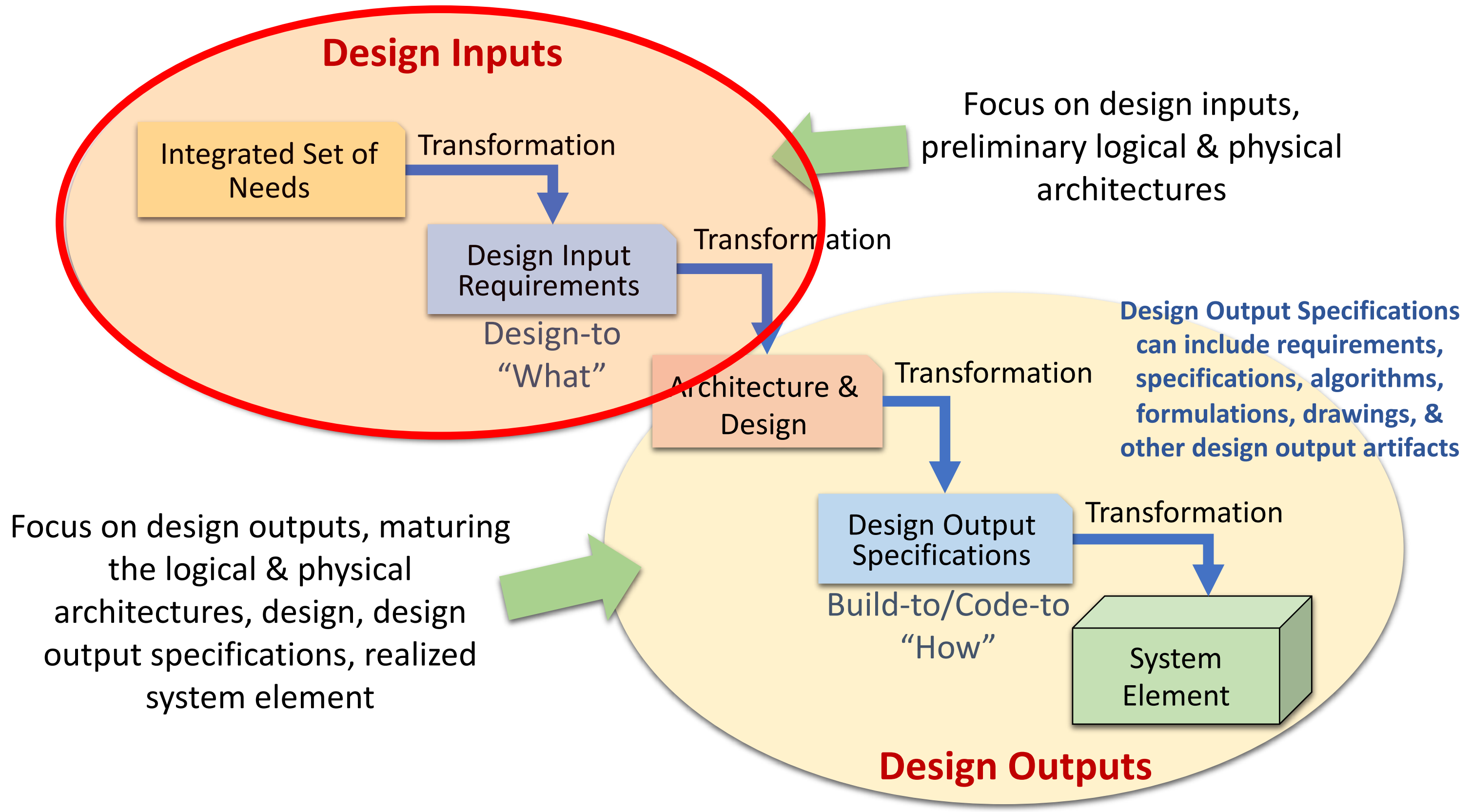

Validation
System Validation
"Did we build the right thing?"
Post Development Validation
"Do we still have the right system?"
Design Validation
"Do we have the right design?"
Rqmts Validation
"Are we building the right thing?"
Needs Validation
"Are we building the right thing?"
Design Inputs
Design Outputs
Stakeholder Real-World Expectations
Transformation
Lifecycle Concepts and Needs Definition
Integrated Set of Needs
Transformation
Design Input Requirements Definition
Design Input Requirements
Transformation
Architecture & Design Definition
Design Output Specifications
Transformation
System Implementation, Integration
Realized System Element
Transition, Operation, Maintenance, & Disposal
Operational System; Maintained System; Disposed System
Integration
Drive
Drive
Drive
Drive
Drive
System Verification
"Did we build it right?"
Design Verification
"Did we design it right?"
Production Verification
"Did we build it right?"
Verification
Post Development Verification
"Is the system still right?"
Organizational Lifecycle Concept & Needs Definition Requirements
Needs Verification
"Are the needs defined correctly?"
Organizational Requirement Definition Requirements
Requirements Verification
"Are the requirements defined correctly?"
Organizational Architecture & Design Definition Requirements
Design Verification
"Did we design it correctly?"
Organizational System Realization Requirements
Production Verification
"Did we build it correctly?"
Organizational Operations, Maintenance, & Disposal Requirements
Operations, Maintenance, & Disposal Verification
"Are we operating, maintaining, and disposing it correctly?"
Design Controls
Original figure created by M. Ryan and L. Wheatcraft. Usage granted per the INCOSE Copyright Restrictions. All other rights reserved.

Table of Contents

Section 6: Design Input Requirements Definition

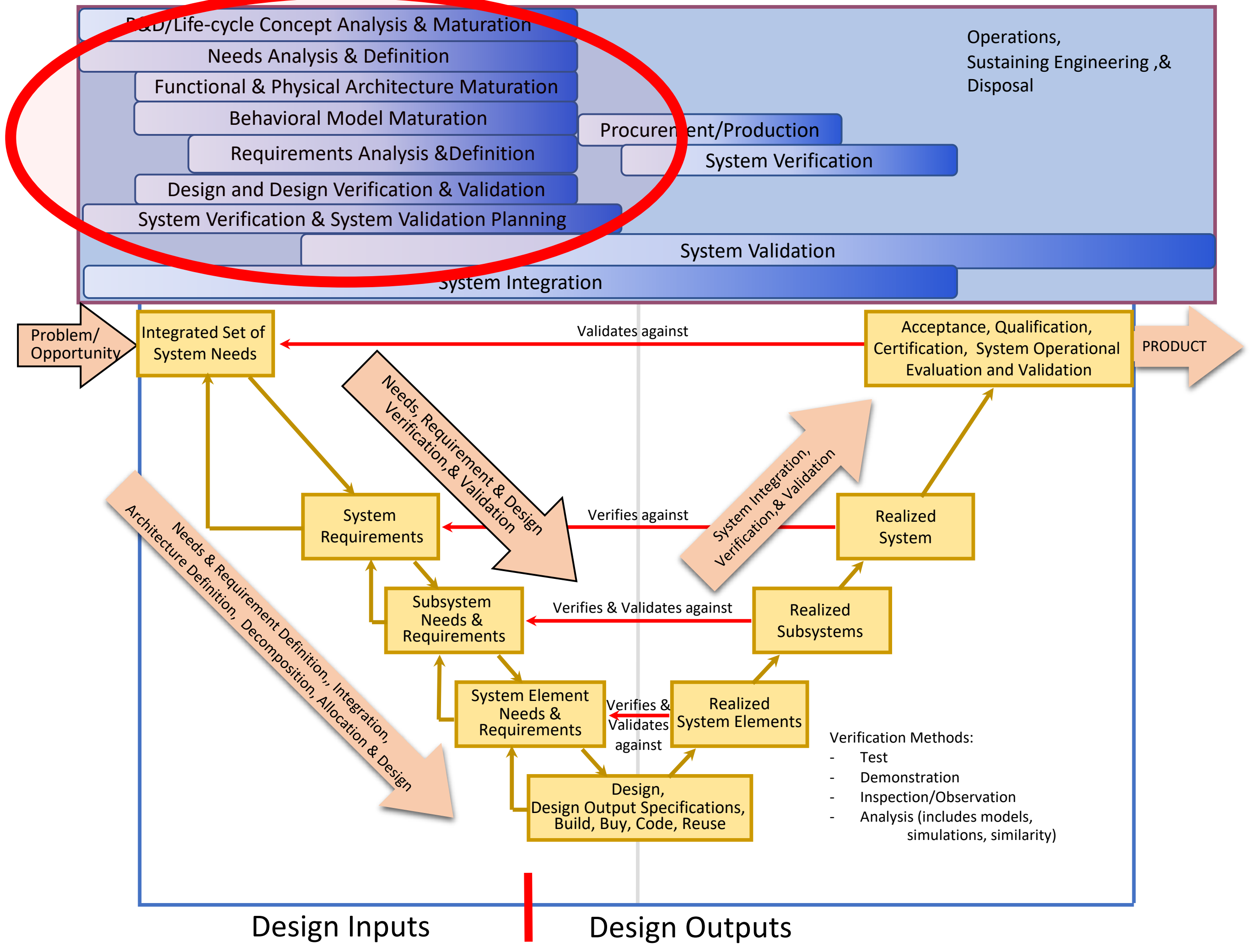
The focus of the Design Input Requirements Definition activities is on transforming the baselined Integrated Set of Needs for a SOI into a unique, quantitative, and measurable set of Design Input Requirements expressed as "shall" statements.

These Design Input Requirements are inputs for defining the system architecture, flowing the requirements down (allocating) from one level of the architecture to the next, and implementing a design solution.


Design Inputs
Integrated Set of Needs
Transformation
Design Input Requirements
Transformation
Design-to "What"
Focus on design inputs, preliminary logical & physical architectures
Architecture & Design
Transformation
Design Output Specifications can include requirements, specifications, algorithms, formulations, drawings, & other design output artifacts
Focus on design outputs, maturing the logical & physical architectures, design, design output specifications, realized system element
Design Output Specifications
Transformation
Build-to/Code-to "How"
System Element
Design Outputs

Original figure created by M. Ryan and L. Wheatcraft. Usage granted per the INCOSE Copyright Restrictions. All other rights reserved.

Section 6: Design Input Requirements Definition


R&D/Life-cycle Concept Analysis & Maturation
Needs Analysis & Definition
Functional & Physical Architecture Maturation
Behavioral Model Maturation
Requirements Analysis &Definition
Design and Design Verification & Validation
System Verification & System Validation Planning
System Validation
System Integration
Operations, Sustaining Engineering ,& Disposal
Procurement/Production
System Verification
Problem/ Opportunity
Integrated Set of System Needs
Validates against
Acceptance, Qualification, Certification, System Operational Evaluation and Validation
PRODUCT
Needs, Requirement & Design Verification,& Validation
System Integration, Verification & Validation
System Requirements
Verifies against
Realized System
Needs & Requirement Definition, Integration, Architecture Definition, Decomposition, Allocation & Design
Subsystem Needs & Requirements
Verifies & Validates against
Realized Subsystems
System Element Needs & Requirements
Verifies & Validates against
Realized System Elements
Design, Design Output Specifications, Build, Buy, Code, Reuse
Verification Methods:
- Test
- Demonstration
- Inspection/Observation
- Analysis (includes models, simulations, similarity)
Design Inputs
Design Outputs

Adapted from M. Ryan, L. 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
Usage granted per the INCOSE Copyright Restrictions. All other rights reserved.

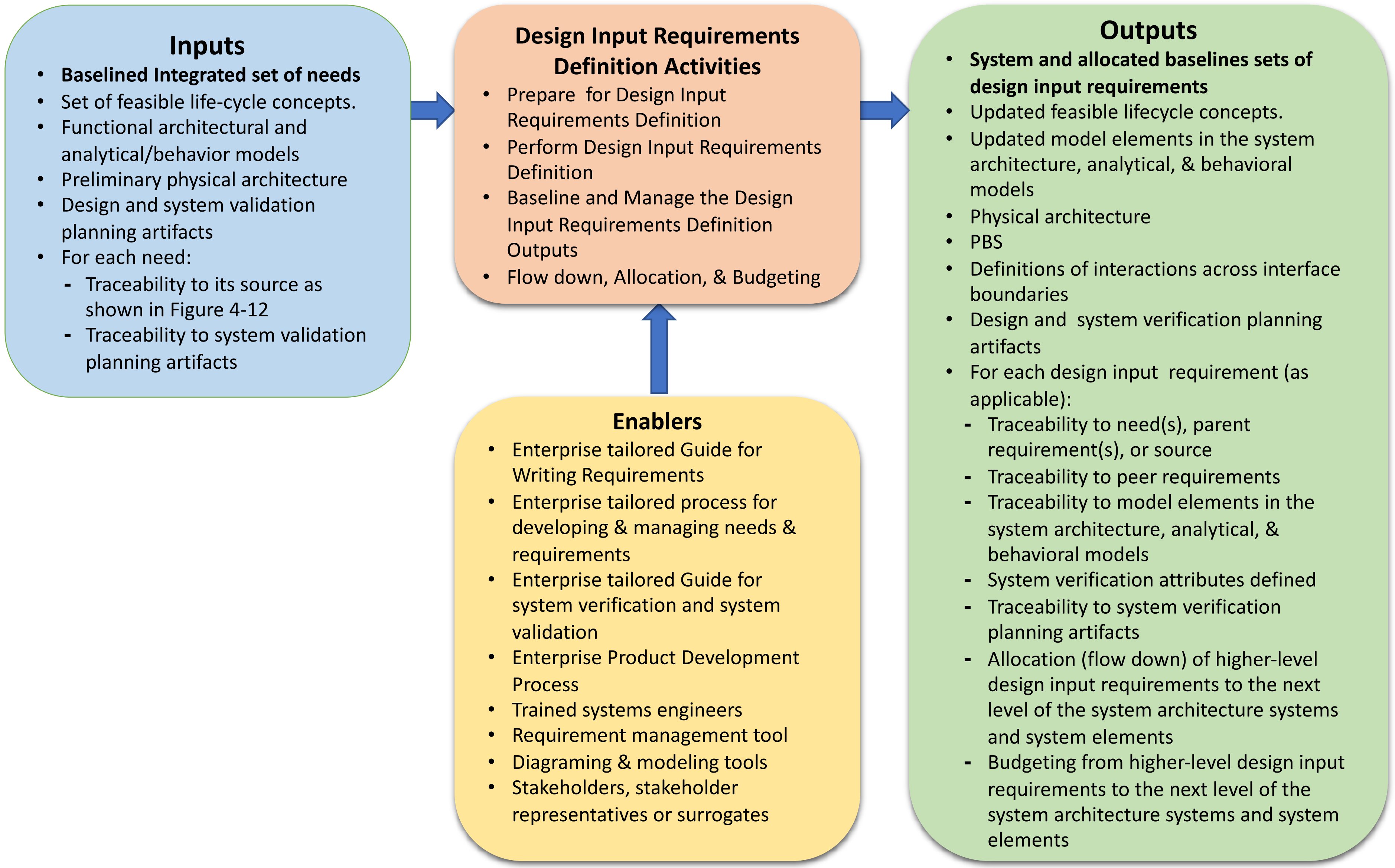
Given that the SE technical lifecycle processes are applied iteratively and recursively as the project team moves down the physical architecture, what is described in this section can be applied to the development of a SOI (system, subsystem, and system element) set of design input requirements - no matter the architectural level the SOI exists.

Section 6: Design Input Requirements Definition

- It is important to understand the I-NRDM approach defined in Section 3 and the Lifecycle Concepts and Needs Definition activities discussed in Section 4.
  - The result of completing these activities is not only an Integrated Set of Needs but also the underling analysis and associated artifacts from which they were formed.
  - As part of these activities, the project team will have been concurrently defining a preliminary set of design input requirements.
- Following this approach, the Design Input Requirements definition activities discussed in this section will build upon this work, resulting in a mature, well-formed set of Design Input Requirements.
- This concurrent approach is preferred in that issues that may come up while defining the preliminary set of Design Input Requirements can be addressed in a less formal, agile manner earlier in the lifecycle during lifecycle concept analysis and maturation activities.
- The resulting lifecycle concepts, models, and integrated set of needs will address these issues prior to them being baselined; avoiding technical debt associated with a more serial document-centric “waterfall” approach.

I-NRDM: Information-Based Needs and Requirement Development and Management

Section 6.1: Prepare for Design Input Requirements Definition



Inputs

- **Baselined Integrated set of needs**
- Set of feasible life-cycle concepts.
- Functional architectural and analytical/behavior models
- Preliminary physical architecture
- Design and system validation planning artifacts
- For each need:
  - Traceability to its source as shown in Figure 4-12
  - Traceability to system validation planning artifacts

Design Input Requirements Definition Activities

- Prepare for Design Input Requirements Definition
- Perform Design Input Requirements Definition
- Baseline and Manage the Design Input Requirements Definition Outputs
- Flow down, Allocation, & Budgeting

Enablers

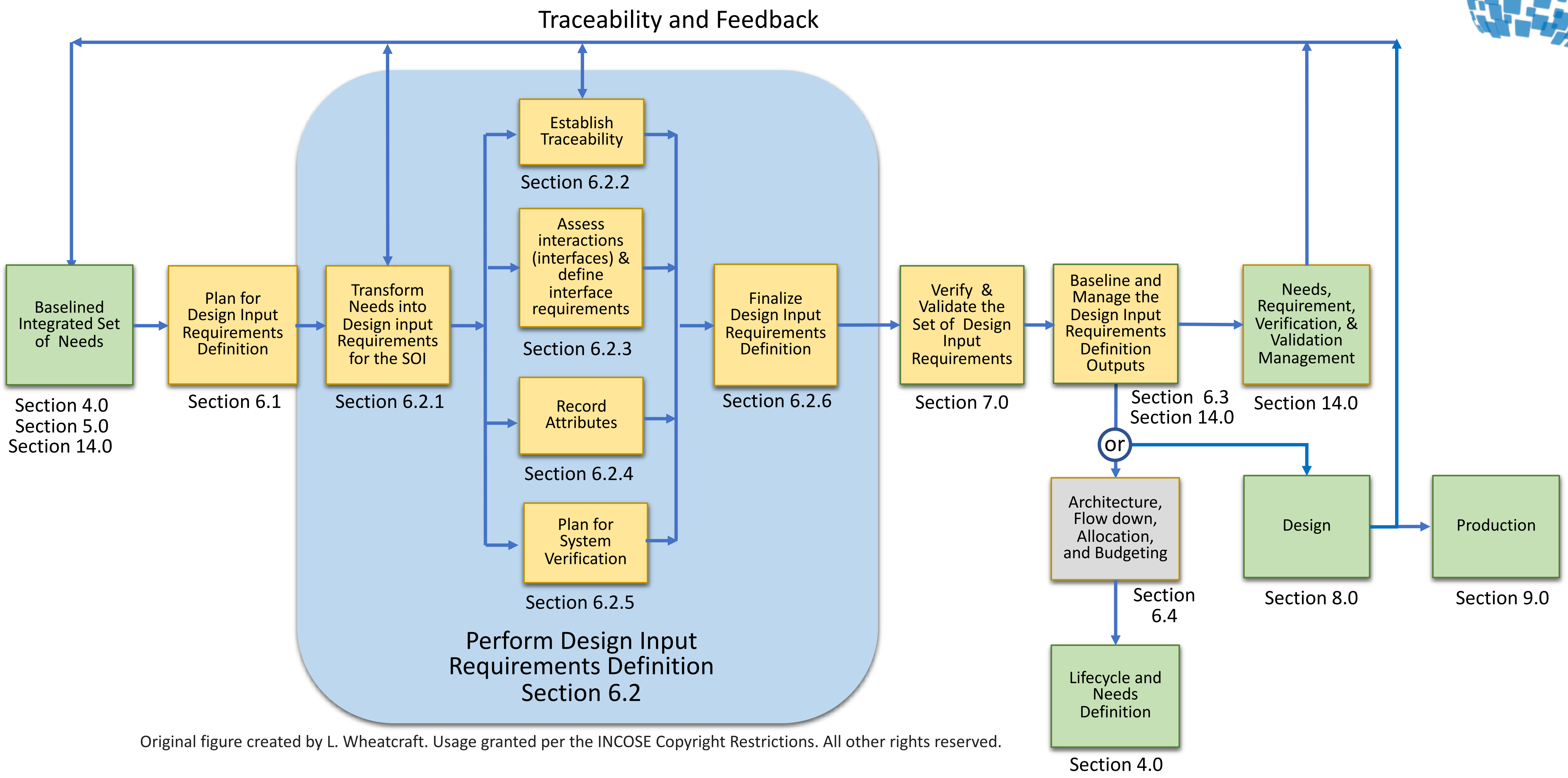
- Enterprise tailored Guide for Writing Requirements
- Enterprise tailored process for developing & managing needs & requirements
- Enterprise tailored Guide for system verification and system validation
- Enterprise Product Development Process
- Trained systems engineers
- Requirement management tool
- Diagraming & modeling tools
- Stakeholders, stakeholder representatives or surrogates

Outputs

- **System and allocated baselines sets of design input requirements**
- Updated feasible lifecycle concepts.
- Updated model elements in the system architecture, analytical, & behavioral models
- Physical architecture
- PBS
- Definitions of interactions across interface boundaries
- Design and system verification planning artifacts
- For each design input requirement (as applicable):
  - Traceability to need(s), parent requirement(s), or source
  - Traceability to peer requirements
  - Traceability to model elements in the system architecture, analytical, & behavioral models
  - System verification attributes defined
  - Traceability to system verification planning artifacts
  - Allocation (flow down) of higher-level design input requirements to the next level of the system architecture systems and system elements
  - Budgeting from higher-level design input requirements to the next level of the system architecture systems and system elements

Original figure created by L. Wheatcraft. Usage granted per the INCOSE Copyright Restrictions. All other rights reserved.

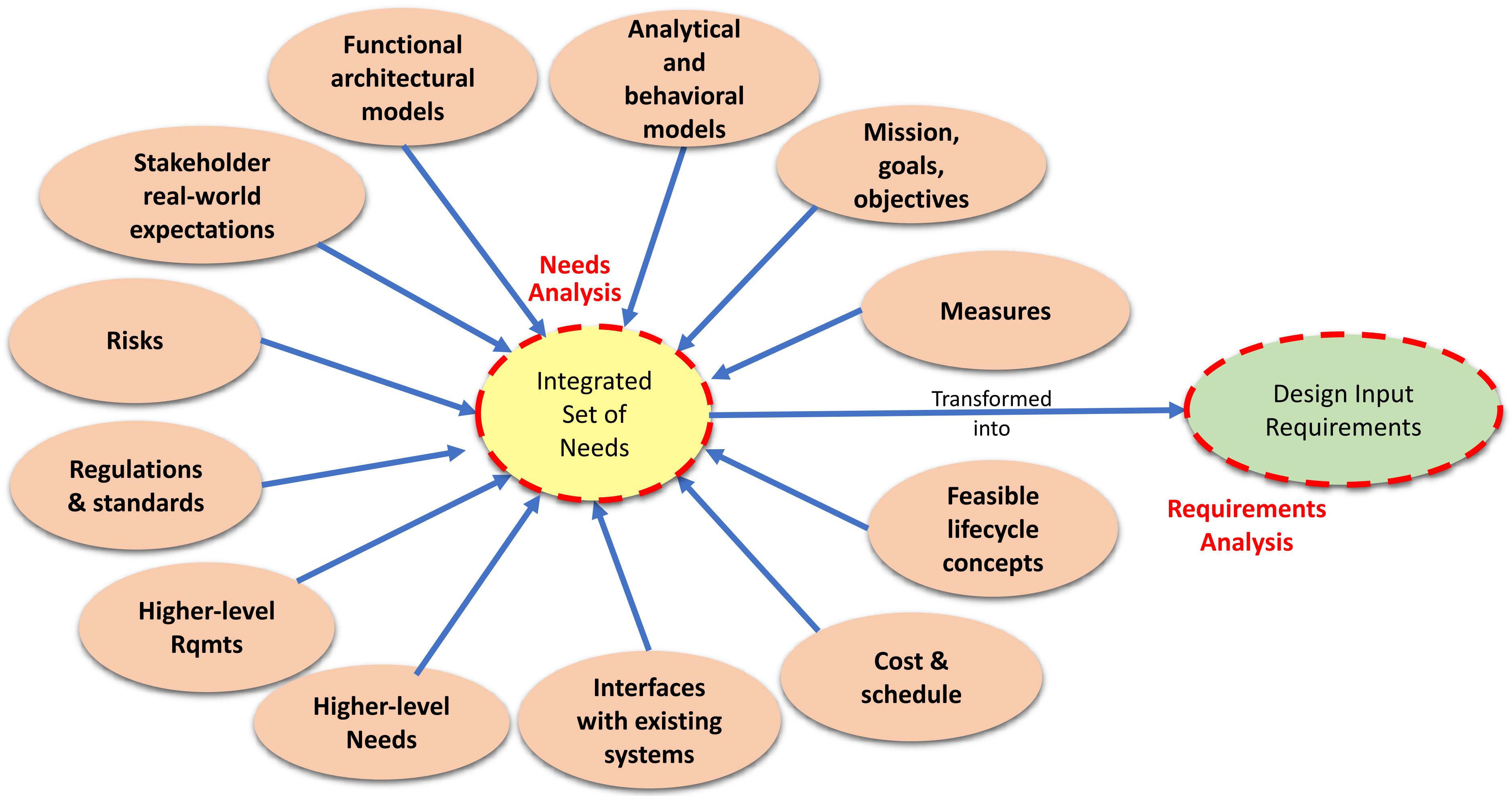
Section 6.2: Prepare for Design Input Requirements Definition



Traceability and Feedback
Baselined Integrated Set of Needs
Section 4.0
Section 5.0
Section 14.0
Plan for Design Input Requirements Definition
Section 6.1
Transform Needs into Design input Requirements for the SOI
Section 6.2.1
Establish Traceability
Section 6.2.2
Assess interactions (interfaces) & define interface requirements
Section 6.2.3
Record Attributes
Section 6.2.4
Plan for System Verification
Section 6.2.5
Finalize Design Input Requirements Definition
Section 6.2.6
Perform Design Input Requirements Definition
Section 6.2
Verify & Validate the Set of Design Input Requirements
Section 7.0
Baseline and Manage the Design Input Requirements Definition Outputs
Section 6.3
Section 14.0
Needs, Requirement, Verification, & Validation Management
Section 14.0
or
Architecture, Flow down, Allocation, and Budgeting
Section 6.4
Design
Section 8.0
Production
Section 9.0
Lifecycle and Needs Definition
Section 4.0

Original figure created by L. Wheatcraft. Usage granted per the INCOSE Copyright Restrictions. All other rights reserved.

Section 6.2.1: Transforming Needs into Design Input Requirements


Functional architectural models
Analytical and behavioral models
Mission, goals, objectives
Stakeholder real-world expectations
Needs Analysis
Measures
Risks
Integrated Set of Needs
Transformed into
Design Input Requirements
Regulations & standards
Feasible lifecycle concepts
Requirements Analysis
Higher-level Rqmts
Cost & schedule
Higher-level Needs
Interfaces with existing systems

By doing the upfront analysis to define lifecycle concepts and an Integrated Set of Needs, the definition of the Design Input Requirements will be much easier.

Original figure created by L. Wheatcraft. Usage granted per the INCOSE Copyright Restrictions. All other rights reserved.

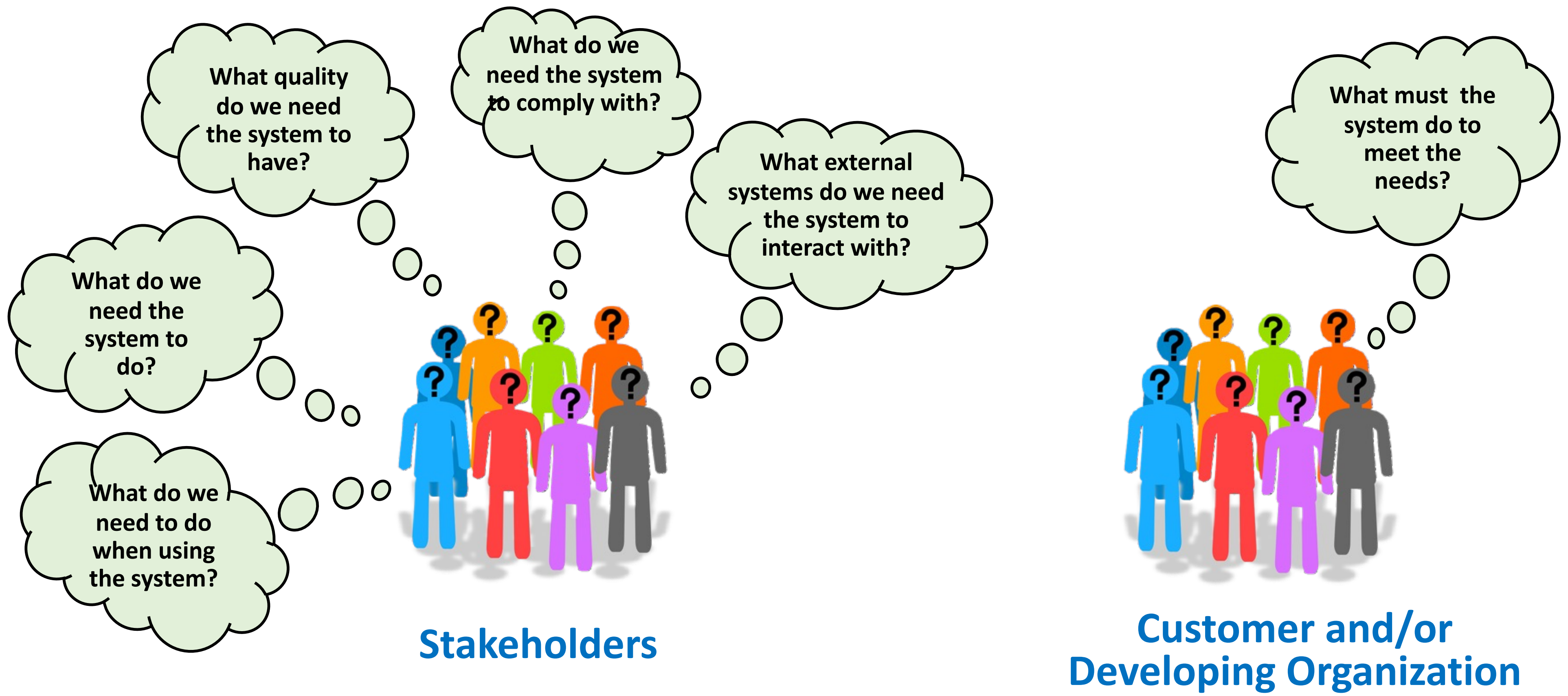
Needs vs Requirements

- Needs represent the stakeholder, customer/acquirer, user view of the system
  - 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 system is delivered – in the end **what will make the customer(s) happy**?
  - Agreed-to and baselined Integrated Set of Needs represents the scope of the project
  - The system will be validated against its Integrated Set of Needs

- Requirements represent the technical, developer view of the system
  - What must the realized system do in order to meet the needs?
  - Needs are transformed into design input requirements that will result in a design that, when realized, will meet the needs.
  - The system will be verified against its Design Input Requirements

Needs vs Requirements

Two Different Perspectives



What quality do we need the system to have?
What do we need the system to comply with?
What external systems do we need the system to interact with?
What do we need the system to do?
What do we need to do when using the system?
Stakeholders
What must the system do to meet the needs?
Customer and/or Developing Organization

Integrated Set of Needs **Vs.** Design Input Requirements

Original figure created by L. Wheatcraft. Usage granted per the INCOSE Copyright Restrictions. All other rights reserved.

Section 6.2.1: Transforming Needs into Design Input Requirements

Column A	Column B	Column C
Needs	Design input requirements	External Interface
Functional/Performance (Function)		
Need 1	Rqmt 1 (could be more than one)	Interface (if applicable)
Need 2	Rqmt 2 (could be more than one)	Interface (if applicable)
Operational (Fit)		
Need 3	Rqmt 3 (could be more than one)	Interface (if applicable)
Need 4	Rqmt 4 (could be more than one)	Interface (if applicable)
Physical Characteristics (Form)		
Need 5	Rqmt 5 (could be more than one)	……….
Need 6	Rqmt 6 (could be more than one)	………..
Quality (-ilities)		
Need 7	Rqmt 7 (could be more than one)	……….
Need 8	Rqmt 8 (could be more than one)	……….
Standards/Regulations (Compliance)		
Need 9	Rqmt 9 (could be more than one)	………
Need 10	Rqmt 10 (could be more than one)	………

Table 6-1: Needs-to-Requirements Transformation Matrix.

The project team does ***requirements analysis*** to transform the Integrated Set of Needs into a set of Design Input Requirements, asking: "What must the SOI do to fulfill each of the needs?"

The answer will be one or more Design Input Requirements that are necessary and sufficient to meet the parent need.

The Needs-to-Requirements Transformation Matrix is one tool that can be used to aid in the transformation.

Section 6.2.1: Transforming Needs into Design Input Requirements

- 6.2.1.1 Organizing the Sets of Design Input Requirements
  - Organizations need to define a template for how they will organize and manage requirements.
- 6.2.1.2 Considerations for Each Type of Requirement
  - Details concerning functional/performance, operational (fit), form, quality, and compliance*.
- 6.2.1.3 Guidelines when Formulating Design Input Requirement Statements
  - Tolerances and Ranges, Accuracy and Precision, System Lifetime and Expected Performance.
- 6.2.1.4 Appropriate to Level
  - Requirements defined for entities at the appropriate level.
- 6.2.1.5 Managing Unknowns
  - Managing TBDs, TBRs, etc.

* See presentation “Standards and Regulations Compliance” on the INCOSE RWG YouTube Channel https://youtu.be/LQmLt4eL_JA

Section 6.2.2: Establish Traceability

- The individual sets of lifecycle concepts, Integrated Set of Needs, Design Input Requirements, design output specifications, system validation artifacts, system verification artifacts do not exist in isolation, rather they represent a multi-level, 3-dimensional "spider web" of relationships which represents a data and information model of the integrated system.

- These relationships are documented via links that allow the relationships to be traced between the entities that are linked both vertically across levels and horizontally across the lifecycle.

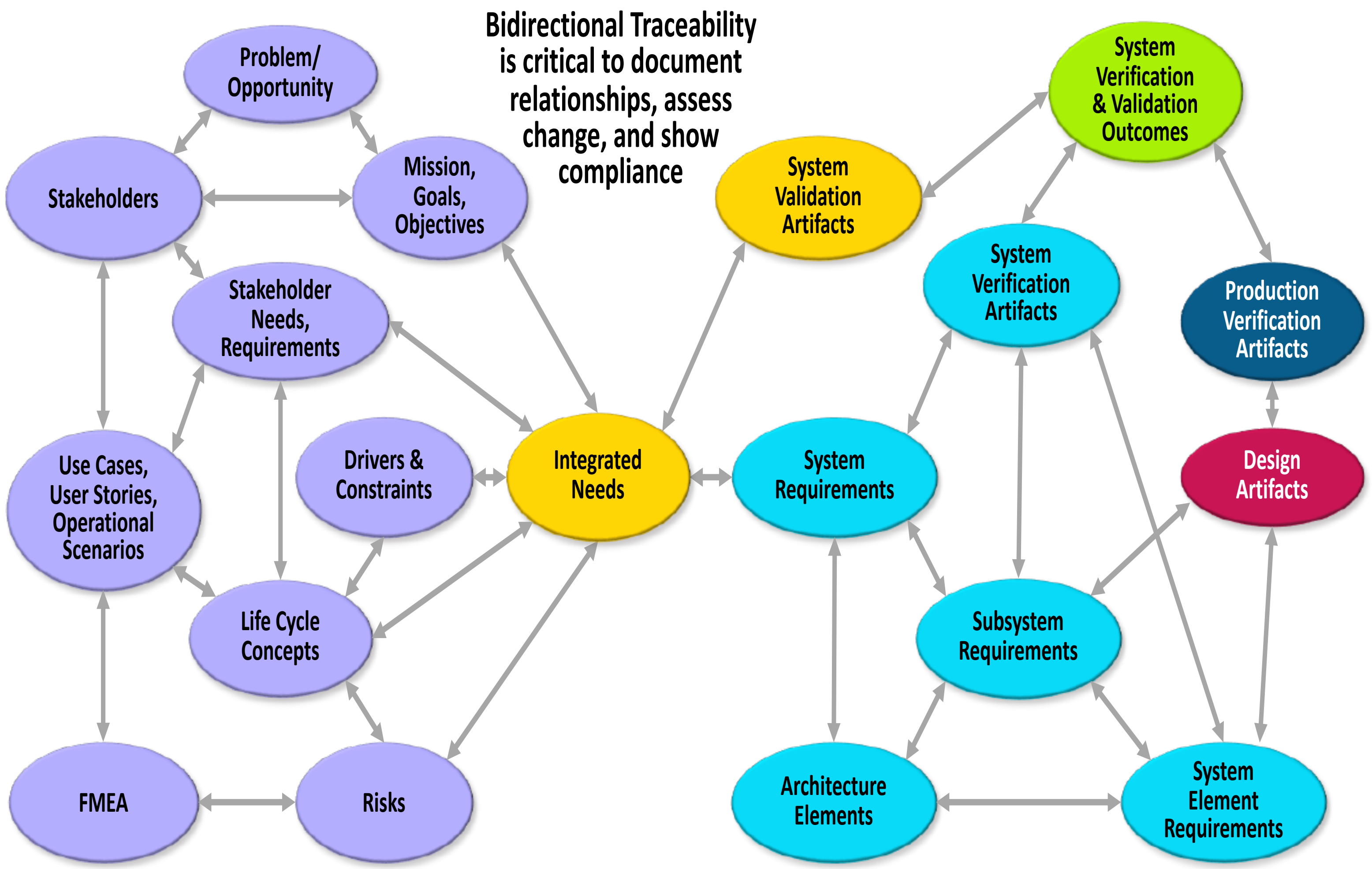
- Requirements can have various types of traceability, including:
  - Parent/Child
  - Source
  - Allocation
  - Peer – Peer (Dependencies)
  - Interface Definition
  - Design
  - System Verification
  - System Validation
  - Model Entity

Traceability Relationship Model

At the beginning of your project, you should choose to develop your product using a data-centric practice of SE.

A first step is selecting your SE toolset and determining what data and information will be developed and managed within the toolset.

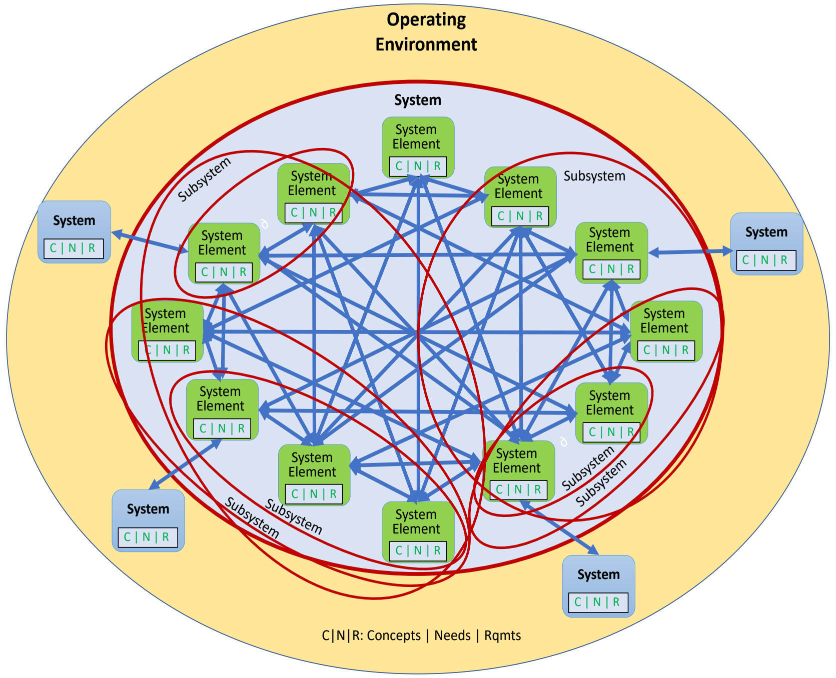
A second step is defining a traceability relationship model addressing which relationships will be established and managed via traceability.


Problem/ Opportunity
Bidirectional Traceability is critical to document relationships, assess change, and show compliance
System Verification & Validation Outcomes
Stakeholders
Mission, Goals, Objectives
System Validation Artifacts
System Verification Artifacts
Production Verification Artifacts
Stakeholder Needs, Requirements
Use Cases, User Stories, Operational Scenarios
Drivers & Constraints
Integrated Needs
System Requirements
Design Artifacts
Life Cycle Concepts
Subsystem Requirements
FMEA
Risks
Architecture Elements
System Element Requirements

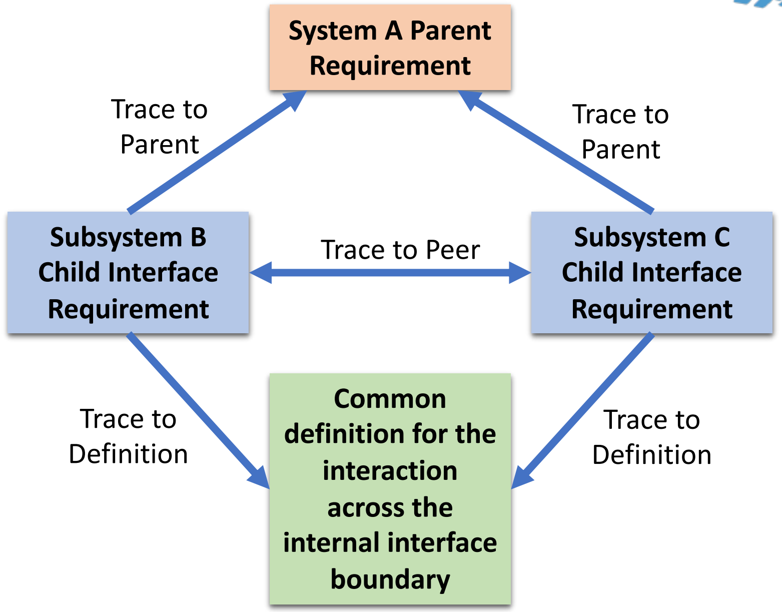
Section 6.2.3: Defining Interactions & Recording Interface Requirements

- The phrase “interface requirement” refers to the specific form or template for a functional/performance requirement that deals with an interaction of a system across an interface boundary with another system.
  - As such, interface requirements should not be considered a separate type of requirement when organizing the set of design input requirements - doing so often leads to confusion and duplication of requirements.
- Interface requirements may be included with the functional/performance requirements or “fit” operational requirements discussed previously.
- All interface requirements have the same general form: (if the interaction is based on some condition, then that condition would be included in the requirement text.)
  - [Condition] [The System] shall [interact (function verb/object) with] [Another System] as defined in [location where the interaction is defined].”
  - [Condition][The System] shall [use/provide from/to] [Another System] [something] having the characteristics defined in [location where the something is defined]”.
  - The word “interface” is not included in an interface requirement as a noun or a verb.

Assessing Interactions Across Interface Boundaries

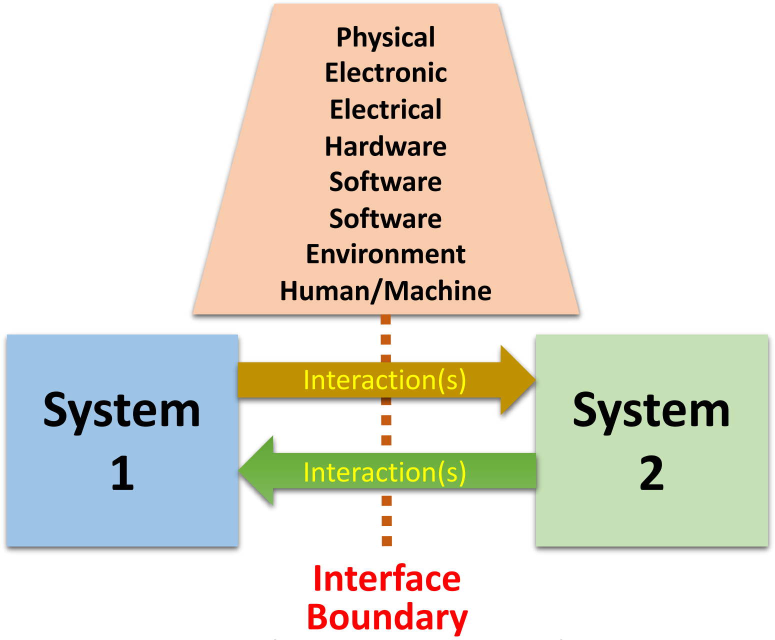


Original figure created by L. Wheatcraft. Usage granted per the INCOSE Copyright Restrictions. All other rights reserved.



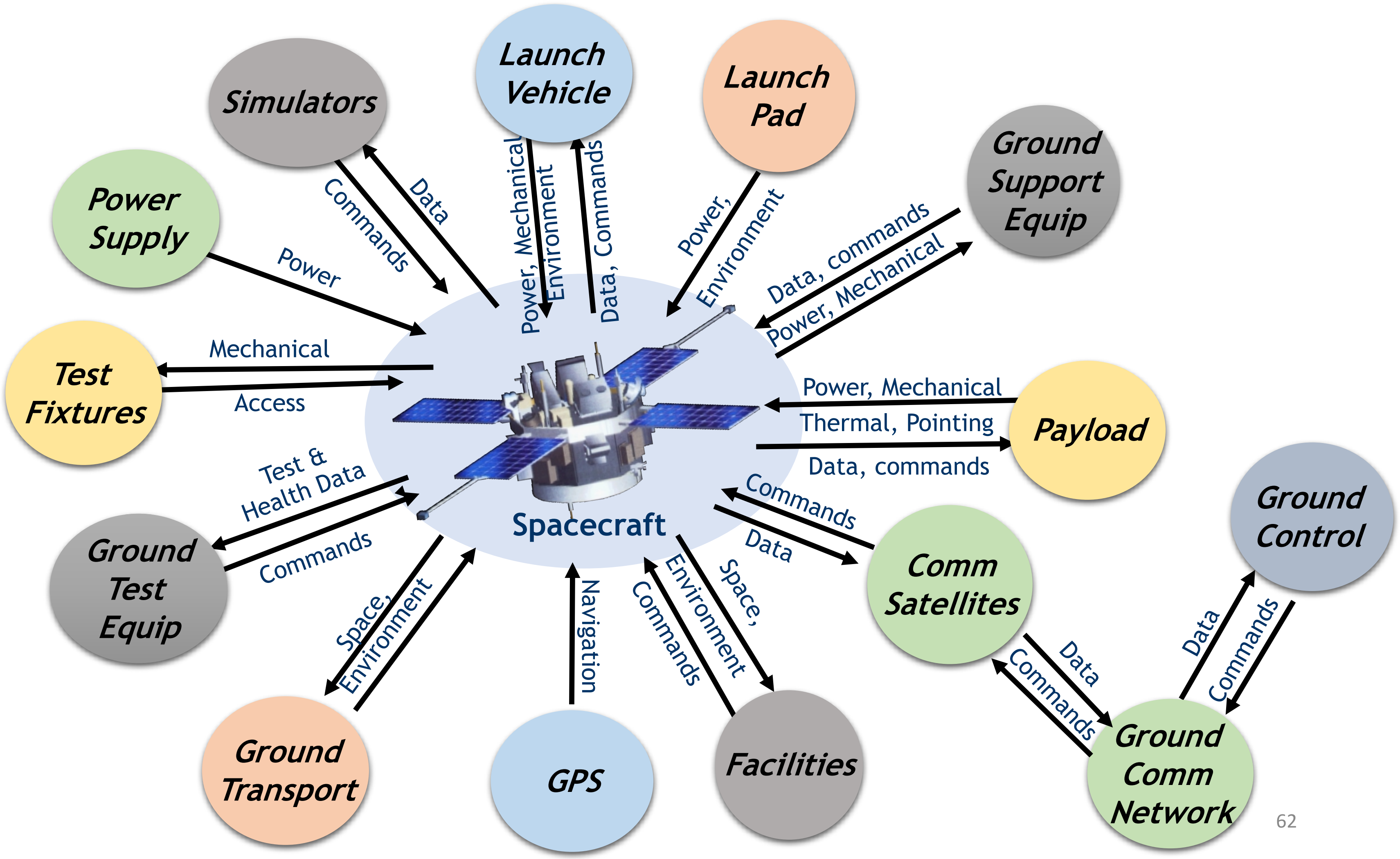
System A Parent Requirement
Trace to Parent
Trace to Parent
Subsystem B Child Interface Requirement
Trace to Peer
Subsystem C Child Interface Requirement
Trace to Definition
Trace to Definition
Common definition for the interaction across the internal interface boundary

Original figure created by L. Wheatcraft.
Usage granted per the INCOSE Copyright Restrictions. All other rights reserved.



Original figure created by L. Wheatcraft.
Usage granted per the INCOSE Copyright Restrictions. All other rights reserved.

External Interface Diagram



Original figure created by L. Wheatcraft. Usage granted per the INCOSE Copyright Restrictions. All other rights reserved.

Interface Requirements Audit

SOI Interface Requirements	SOI Interaction Definitions	External System Interaction Definition	External System	External Systems Interface Requirements
SOI IR 1	Def 111	Def 111	ES1	ES1 IR 1
SOI IR 2	**Def 112**	**Def 123**	ES1	ES1 IR 2
SOI IR 3	Def 113		ES1	**Missing**
SOI IR 4	**TBD**	**TBR**	ES1	ES1 IR 4
SOI IR 5	**Missing**	Def 134	ES2	ES2 IR 1
Missing		Def 135	ES2	ES2 IR 2
SOI IR 6	Def 114	Def 114	ES2	ES2 IR 3
SOI IR 7	Def 115		ES2	**Missing**
SOI IR 8	**Def 116**	**Def 128**	ES3	ES3 IR 1
SOI IR 9	Def 117		ES3	**Missing**
SOI IR 10	**Missing**	Def 131	ES3	ES3 IR 1
SOI IR 11	**TBD**	**TBR**	ES3	ES3 IR 3
Missing		Def 131	ES4	ES4 IR 1
SOI IR 12	Def 138	Def 138	ES4	ES4 IR 2
SOI IR 13	Def 119		ES4	**Missing**
Missing		Def 135	ES4	ES4 IR 3

Table 6-4: Example Interface Requirements Audit.

Presentation on the INCOSE RWG YouTube Channel "Everything You Wanted to Know About Interfaces but Were Afraid to Ask!" https://youtu.be/7qcoSeBEJ5Y

Section 6.2.4: Use of Attributes to Develop & Manage Design Input Requirements

A1 - *Rationale*: intent of the requirement statement.
A2 - *Trace to Parent*: The parent need or parent requirement from which the requirement was transformed
A3 - *Trace to Source*: Where the requirement originated: stakeholder, user story, scenario, use case, constraint, risk, lifecycle concept, analysis, model, etc.
A5 - *Allocation/Budgeting*: Subsystem or system element at the next level of the architecture to which the requirement is being allocated/budgeted
A26 - *Stability*: stable, likely to change, and incomplete.
A28 - *Requirement Verification Status*: true/false, yes/no, or not started, in work, complete, and approved.
A29 - *Requirement Validation Status*: true/false, yes/no, not started, in work, complete, and approved.
A30 - *Status of the requirement* (in terms of maturity): draft, in development, ready for review, in review and approved. (A28 is a dependent on A26 and A27)
A31 - *Status of implementation*: at higher levels of the architecture a trace to the implementing child requirements. At the bottom level, a trace to the design description that implements the intent of the requirement.
A32 - *Trace to Interface Definition*
A33 - *Trace to Peer Requirements*
A34 - *Priority*: Relative importance of the requirement.
A35 - *Criticality*: Achievement of the requirement is critical to the SOI being able to meet its intended use in the operational environment.
A36 - *Risk* (of implementation): one or more risk factors associated with being able to achieve the requirement.
A37 - *Risk* (mitigation): The requirement is linked to a risk the project has decided to mitigate within the SO design. Often related to safety, security, quality.
A38 - *Key Driving Requirement* (KDR): Implementing the requirement could have a significant impact or cost and/or schedule.

(See Section 15, for a more detailed discussion on attributes, definitions of each, & guidance concerning the use of attributes.)

Section 6.2.5: Plan for System Verification

- The outcome of all successful projects is a verified and validated SOI that has been accepted by the customer or has been approved for use by the public.
  - System verification is obtaining the evidence needed to show that the SOI satisfies its set of Design Input Requirements.
- The GtWR includes the characteristic, C7 - Verifiable, for well-formed design input requirement statements.
  - "Verifiable" means each requirement statement is structured and worded such that its realization by the design and resulting SOI can be verified to have been met to the customer's or regulator's satisfaction at the level the requirement exists.
- A best practice to ensure the Design Input Requirements are "verifiable", is to plan for how the project will verify that the system will meet each requirement within the sets of the design input requirements during system verification activities discussed in Section 10 and 11 of the NRM and the GtVV.

Section 6.2.5: Plan for System Verification

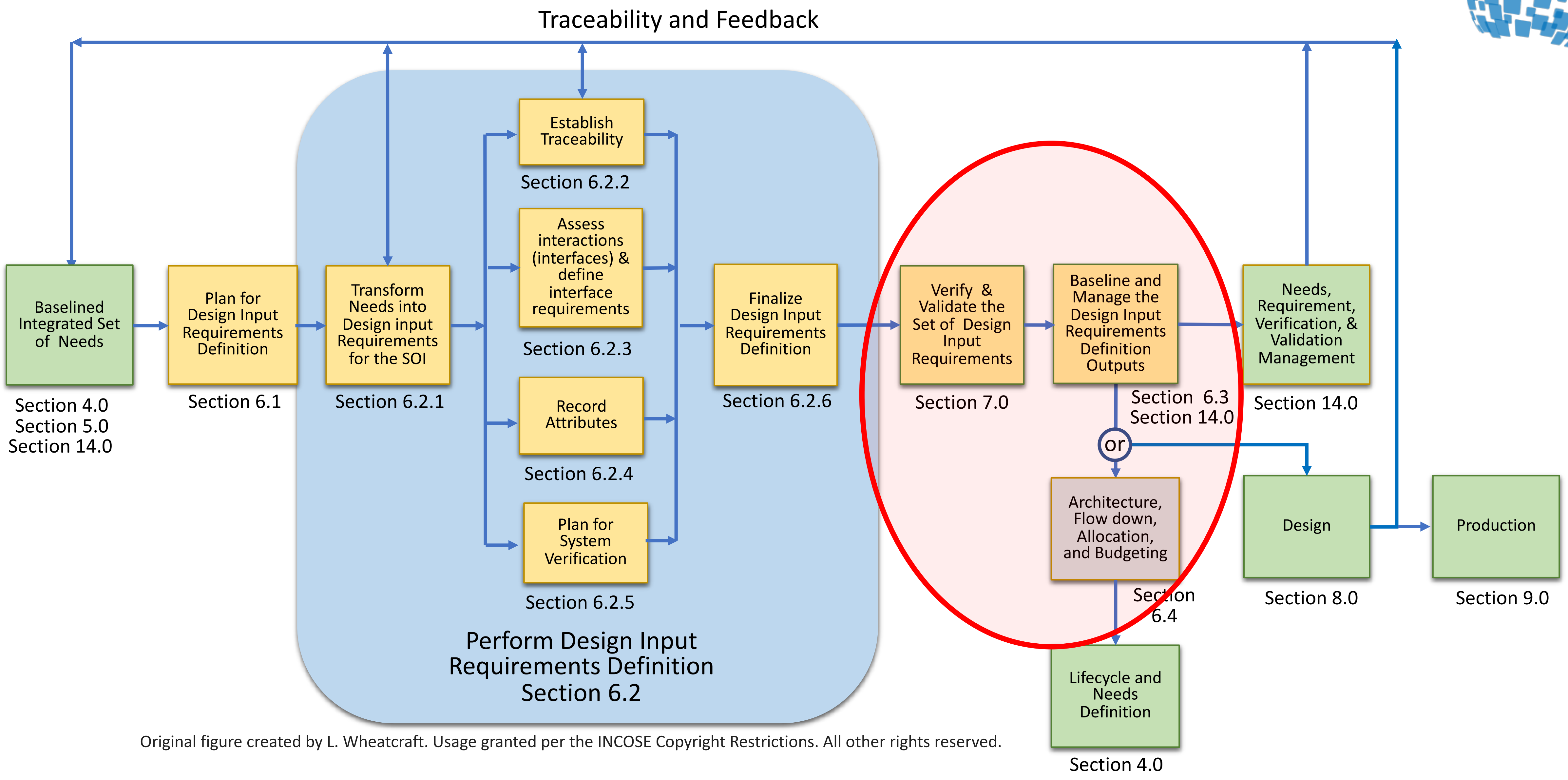
- Information that should be defined concerning system verification for each requirement statement includes system verification Method, Strategy, Success Criteria, and the organization responsible for the planning and execution of the system verification activities.
- This information can be defined within the system verification attributes that should be included (along with the other attributes discussed in Section 6.2.4) within each design input requirement expression. These verification attributes include as a minimum:
  - A6 - System Verification Success Criteria
  - A7 - System Verification Strategy
  - A8 - System Verification Method
  - A9 - System Verification Responsible Organization
  - A12 - Condition of Use: operational conditions of use expected in which the requirement applies

Note: Refer to the Section 10 for a detailed discussion concerning the system verification Method, Strategy, and Success Criteria and Section 15 for a discussion on the use of attributes.

Section 6.2.6: Finalize Design Input Requirements Definition

- Record the Design Input Requirements
  - The set of design input requirements must be recorded within the SOI's integrated dataset in a form and media suitable for review and feedback from the stakeholders as well as support traceability across the lifecycle and the requirement verification and requirement validation activities.
- Assess completeness, consistency, and correctness,
  - Well-formed sets design input requirements have the characteristics C10 – Complete, C11- Consistent, C15 – Correct.
- Assess feasibility and risk
  - Well-formed requirement statements have the characteristic C6 - Feasible, and sets of design input requirements have the characteristic, C12 – Feasible.

Were we are:



Traceability and Feedback
Baselined Integrated Set of Needs
Section 4.0
Section 5.0
Section 14.0
Plan for Design Input Requirements Definition
Section 6.1
Transform Needs into Design input Requirements for the SOI
Section 6.2.1
Establish Traceability
Section 6.2.2
Assess interactions (interfaces) & define interface requirements
Section 6.2.3
Record Attributes
Section 6.2.4
Plan for System Verification
Section 6.2.5
Finalize Design Input Requirements Definition
Section 6.2.6
Perform Design Input Requirements Definition
Section 6.2
Verify & Validate the Set of Design Input Requirements
Section 7.0
Baseline and Manage the Design Input Requirements Definition Outputs
Section 6.3
Section 14.0
or
Architecture, Flow down, Allocation, and Budgeting
Section 6.4
Lifecycle and Needs Definition
Section 4.0
Needs, Requirement, Verification, & Validation Management
Section 14.0
Design
Section 8.0
Production
Section 9.0

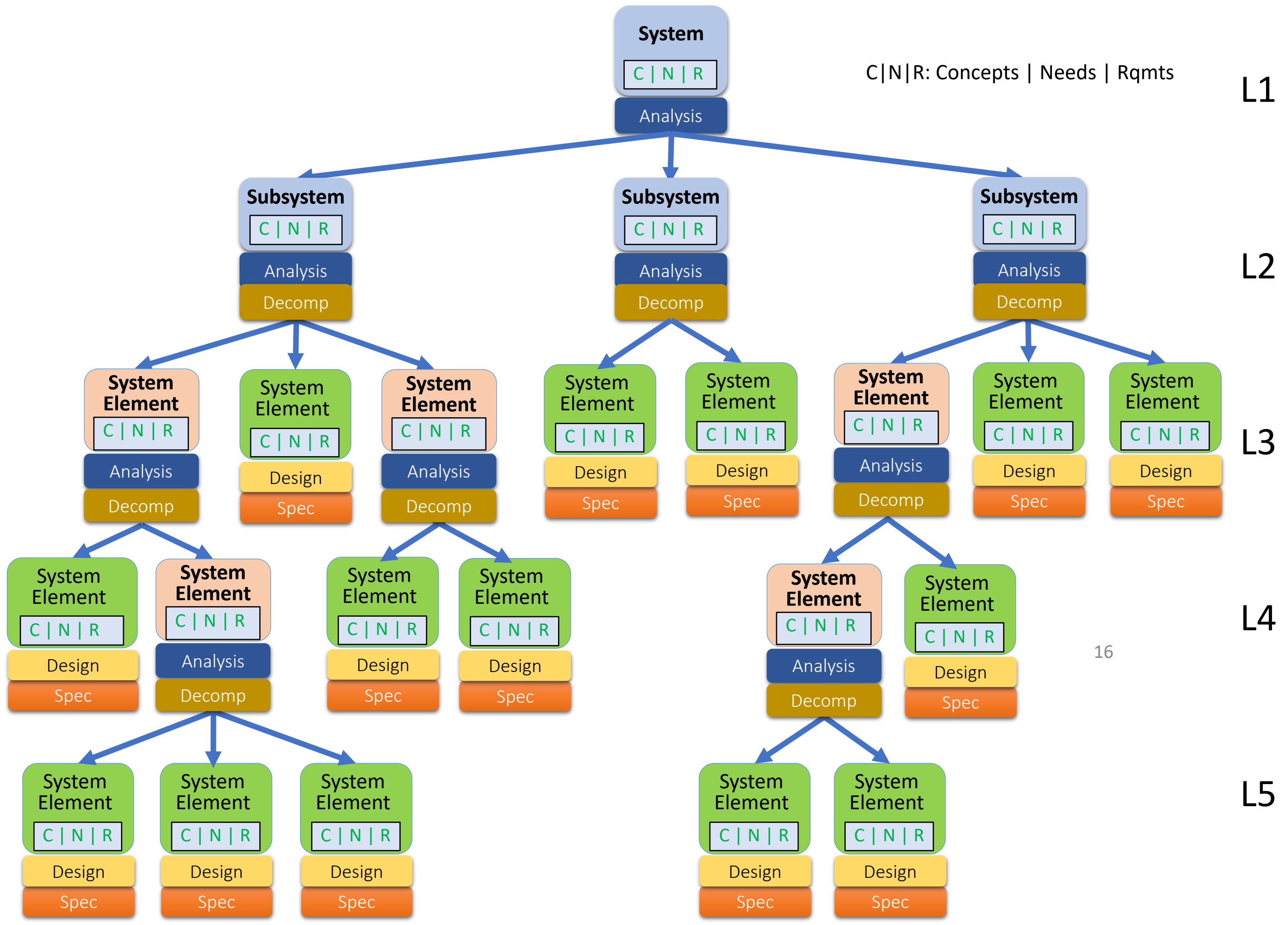
Original figure created by L. Wheatcraft. Usage granted per the INCOSE Copyright Restrictions. All other rights reserved.

Section 6.3: Baseline and Manage Design Input Requirements

- The approval and baselining of the sets of Design Input Requirements is discussed in Section 14, Needs, Requirements, Verification, and Validation Management.
- As part of the approval and baselining of the sets of Design Input Requirements, the project team will need to complete the Design Input Requirements verification and validation activities **defined in Section 7**.
- These activities are critical to ensuring
  - Individual requirements have the GtWR characteristics C1 - Necessary, C3 - Unambiguous, C4 - Complete, C6 - Feasible, and C8 – Correct.
  - Sets of requirements have the GtWR characteristics C10 – Complete, C11 – Consistent, C12 – Feasible, C13 – Comprehensible, C14 - Able to be Validated, and C15 - Correct.

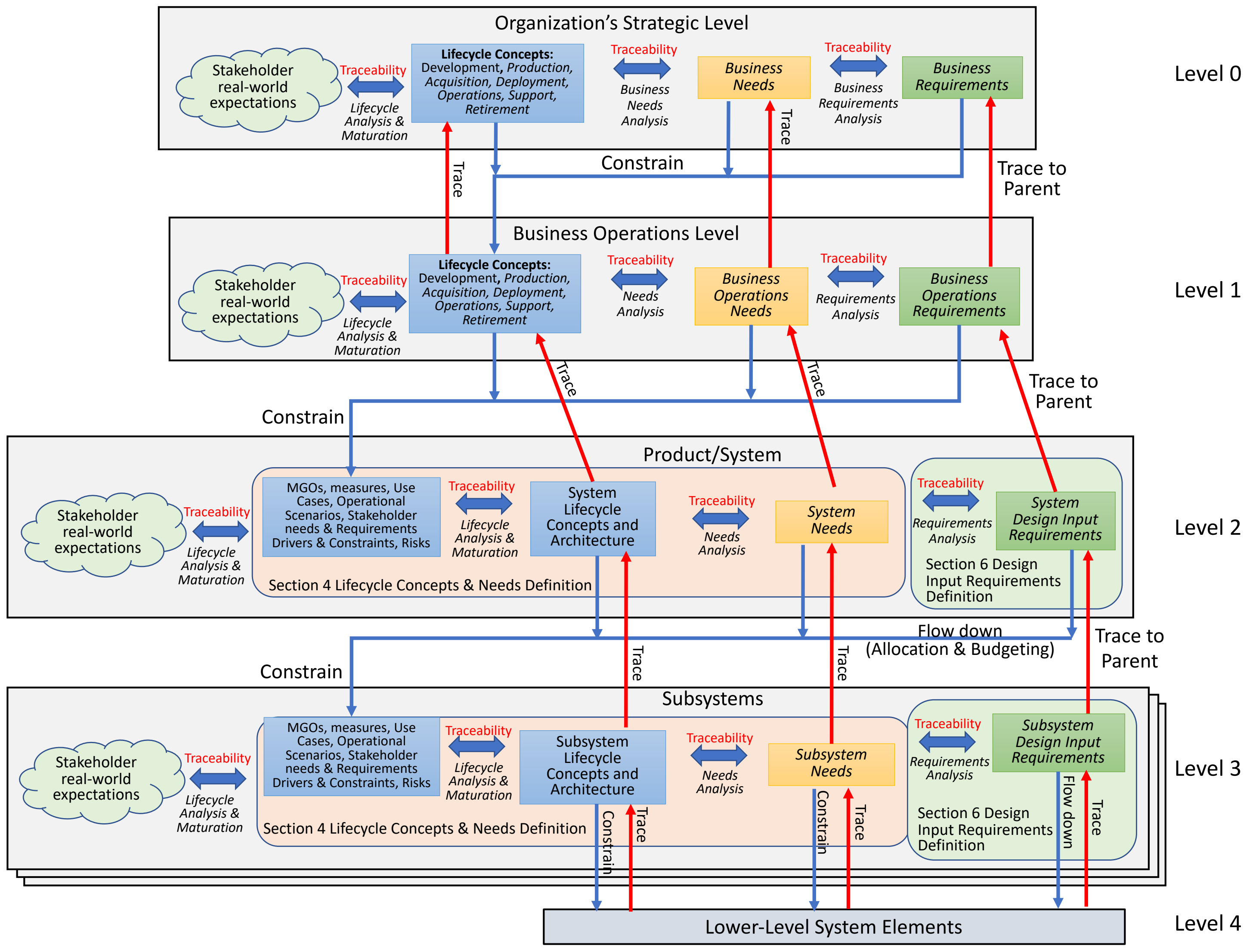
Section 6.4: Architectural Levels, Flow down, Allocation, & Budgeting

- Unless a SOI requires no further elaboration as discussed in Section 2.3.2, once its set of Design Input Requirements have been defined, verified, validated, and baselined, the project team will flow the requirements down to the subsystems and system elements at the next level of the physical architecture via allocation and budgeting.
- Architecting and Design Input Requirement definition go together iteratively and recursively as the project team defines the levels of the architecture.



Original figure created by M. Ryan and L. Wheatcraft. Usage granted per the INCOSE Copyright Restrictions. All other rights reserved.

Section 6.4.1:Moving Between Levels of the Physical Architecture



Organization's Strategic Level
Stakeholder real-world expectations
Traceability
Lifecycle Analysis & Maturation
Lifecycle Concepts: Development, Production, Acquisition, Deployment, Operations, Support, Retirement
Traceability
Business Needs Analysis
Business Needs
Traceability
Business Requirements Analysis
Business Requirements
Trace
Constrain
Trace to Parent
Level 0
Business Operations Level
Stakeholder real-world expectations
Traceability
Lifecycle Analysis & Maturation
Lifecycle Concepts: Development, Production, Acquisition, Deployment, Operations, Support, Retirement
Traceability
Needs Analysis
Business Operations Needs
Traceability
Requirements Analysis
Business Operations Requirements
Level 1
Trace
Trace
Trace to Parent
Constrain
Product/System
Stakeholder real-world expectations
Traceability
Lifecycle Analysis & Maturation
MGOs, measures, Use Cases, Operational Scenarios, Stakeholder needs & Requirements Drivers & Constraints, Risks
Traceability
Lifecycle Analysis & Maturation
System Lifecycle Concepts and Architecture
Traceability
Needs Analysis
System Needs
Section 4 Lifecycle Concepts & Needs Definition
Traceability
Requirements Analysis
System Design Input Requirements
Section 6 Design Input Requirements Definition
Level 2
Flow down (Allocation & Budgeting)
Trace to Parent
Trace
Trace
Constrain
Subsystems
Stakeholder real-world expectations
Traceability
Lifecycle Analysis & Maturation
MGOs, measures, Use Cases, Operational Scenarios, Stakeholder needs & Requirements Drivers & Constraints, Risks
Traceability
Lifecycle Analysis & Maturation
Subsystem Lifecycle Concepts and Architecture
Traceability
Needs Analysis
Subsystem Needs
Section 4 Lifecycle Concepts & Needs Definition
Traceability
Requirements Analysis
Subsystem Design Input Requirements
Section 6 Design Input Requirements Definition
Constrain
Trace
Constrain
Trace
Flow down
Trace
Level 3
Lower-Level System Elements
Level 4

Original figure created by M. Ryan and L. Wheatcraft. Usage granted per the INCOSE Copyright Restrictions. All other rights reserved.

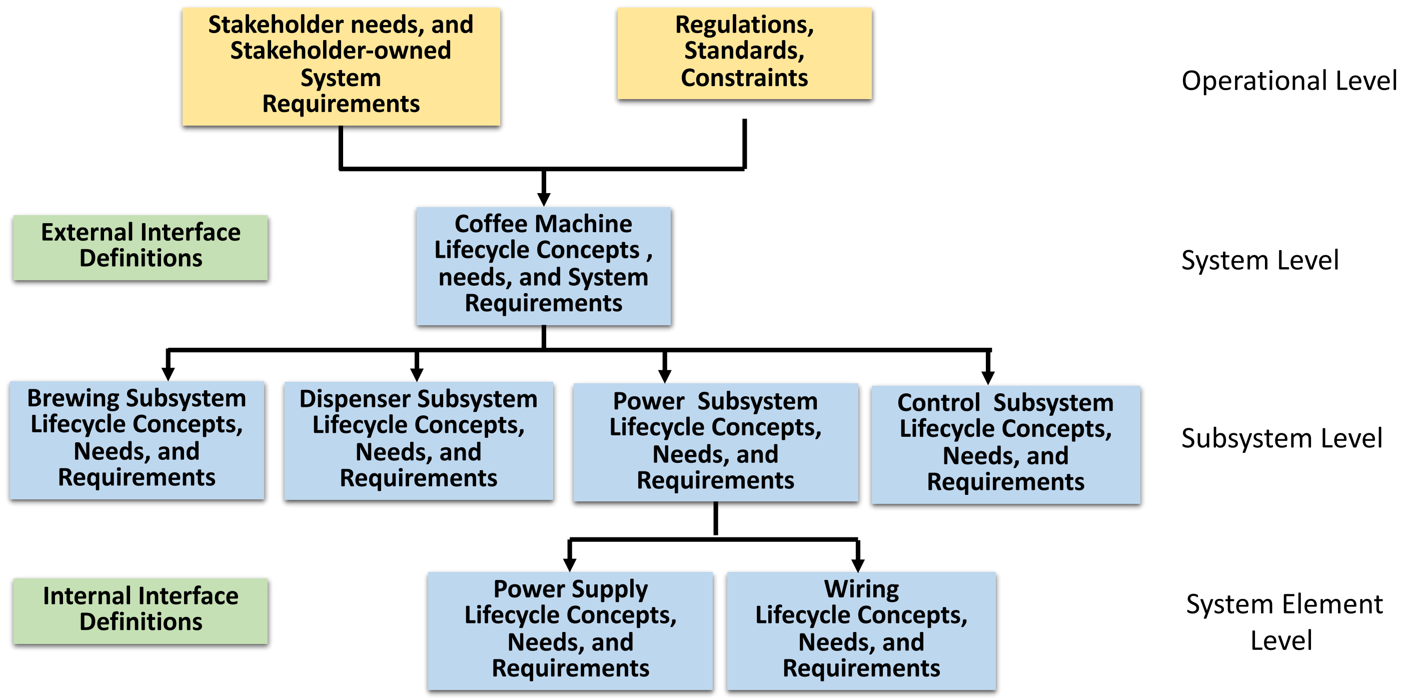
Requirements at one level are allocated/budgeted to the subsystems and system elements at the next level

These requirements are inputs to the lifecycle concepts, needs, and requirements definition process for each of the subsystems and system elements to which the requirements were allocated/budgeted

Process repeats until a buy, build, code, or reuse decision is made.

Section 6.4.2: Product Breakdown Structure and Document Tree

- From a PM perspective, the physical architecture can be mapped to a PBS.
  - The PBS is similar in concept to a WBS.
  - The system along with each subsystem and system element within the system physical architecture represents an entity within the PBS.
  - Each of these entities are represented by a budget, schedule, development concept, procurement concept, and a set of both project work products and engineering artifacts



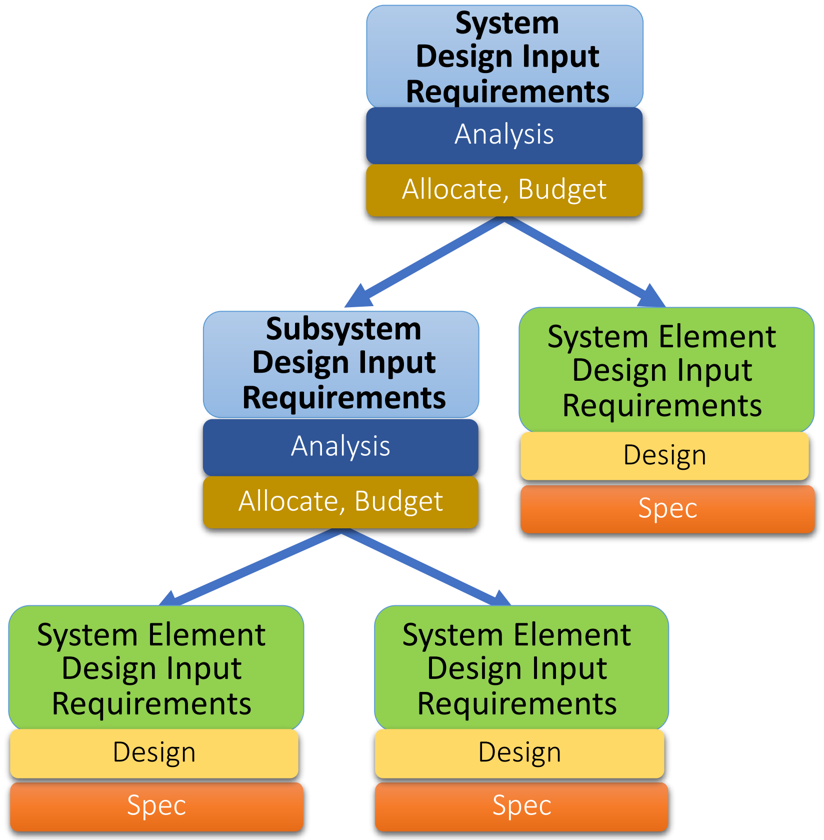
Stakeholder needs, and Stakeholder-owned System Requirements
Regulations, Standards, Constraints
Operational Level
External Interface Definitions
Coffee Machine Lifecycle Concepts , needs, and System Requirements
System Level
Brewing Subsystem Lifecycle Concepts, Needs, and Requirements
Dispenser Subsystem Lifecycle Concepts, Needs, and Requirements
Power Subsystem Lifecycle Concepts, Needs, and Requirements
Control Subsystem Lifecycle Concepts, Needs, and Requirements
Subsystem Level
Internal Interface Definitions
Power Supply Lifecycle Concepts, Needs, and Requirements
Wiring Lifecycle Concepts, Needs, and Requirements
System Element Level

Original figure created by T. Katz and L. Wheatcraft. Usage granted per the INCOSE Copyright Restrictions. All other rights reserved.

- To completely describe the SOI, lifecycle concepts, needs, and requirements sets for each entity within the PBS are needed.
  - This results in a hierarchical family of artifacts, historically referred to as a “document tree”.
  - In today’s data-centric practice of SE each of the artifacts are represented a hierarchical sets of artifacts.

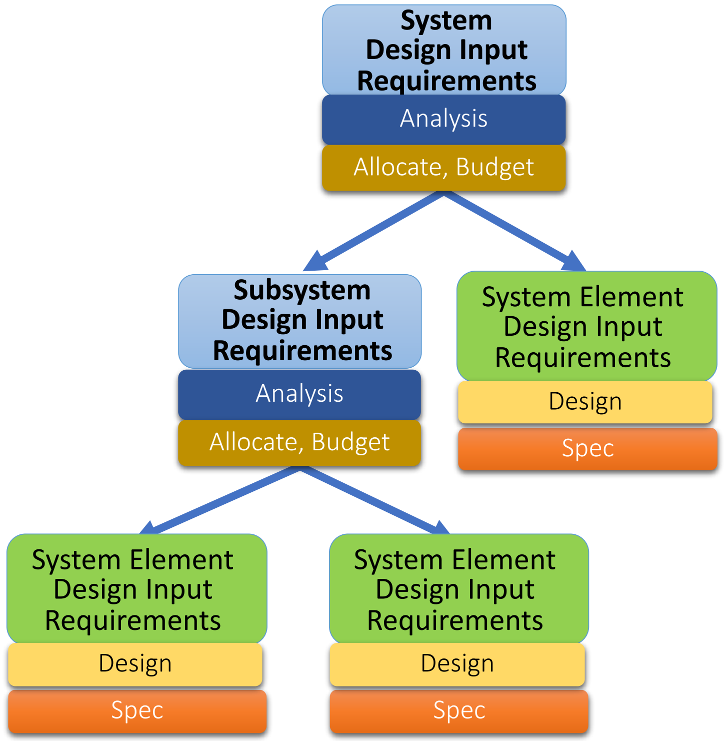
Section 6.4.3: Allocation - Flow Down of Requirements

- Allocation is the process by which the design input requirements defined for an entity at one level of the physical architecture are assigned (flow down) to the entities at the next lower level of the architecture that have a role in the implementation of the allocated requirement.
- Based on analysis of the design input requirements and the functions of the system whose requirements are being allocated, the Architecture Definition Process decomposes the system into subsystems and system elements, resulting in the next level of the SOI physical architecture.
- For each subsystem or system element that has a role in meeting the allocated parent requirement, the requirement will be allocated to those subsystems or system elements.


System Design Input Requirements
Analysis
Allocate, Budget
Subsystem Design Input Requirements
Analysis
Allocate, Budget
System Element Design Input Requirements
Design
Spec
System Element Design Input Requirements
Design
Spec
System Element Design Input Requirements
Design
Spec

Original figure created by L. Wheatcraft.
Usage granted per the INCOSE Copyright Restrictions. All other rights reserved.

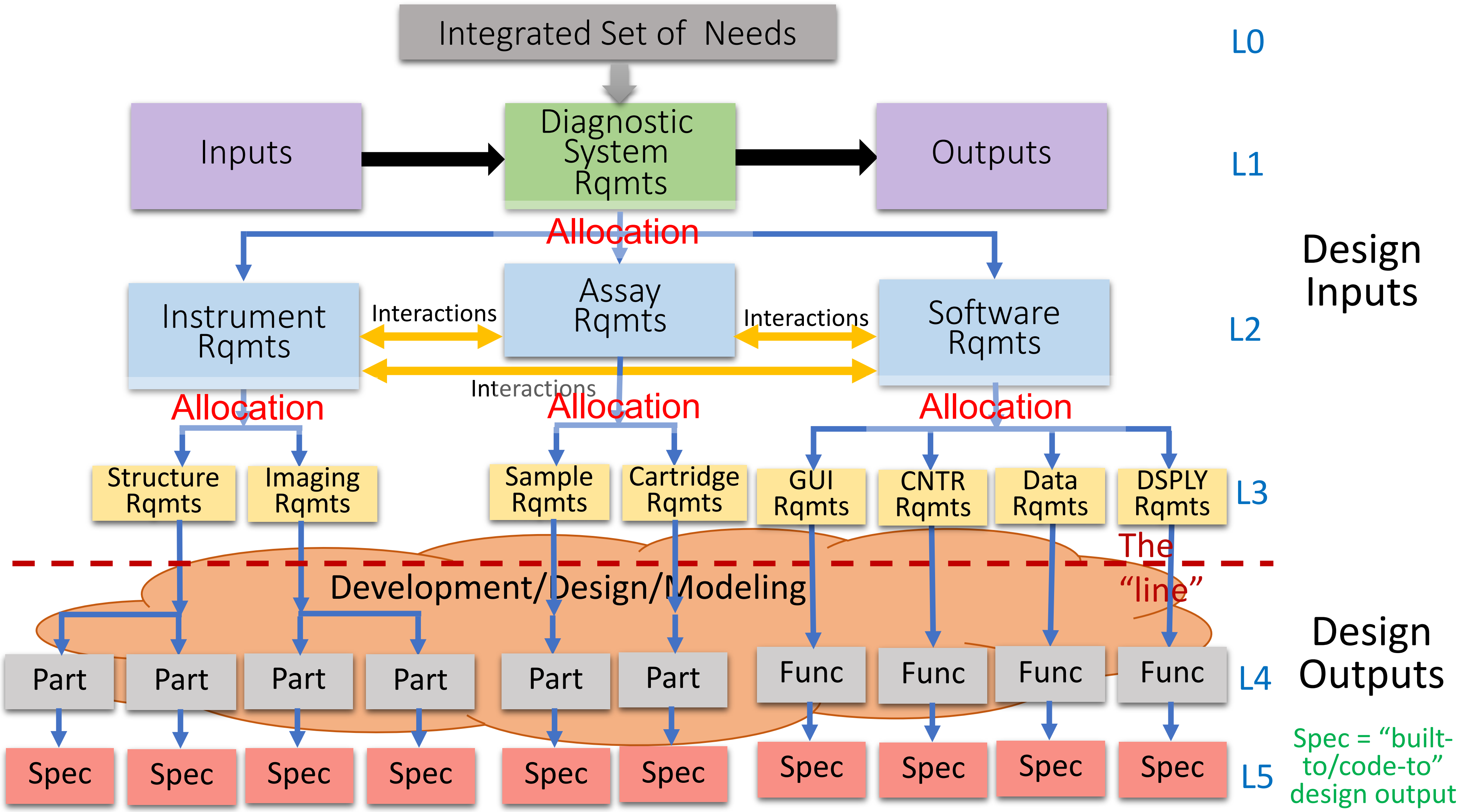
Section 6.4.3: Allocation - Flow Down of Requirements


System Design Input Requirements
Analysis
Allocate, Budget
Subsystem Design Input Requirements
Analysis
Allocate, Budget
System Element Design Input Requirements
Design
Spec
System Element Design Input Requirements
Design
Spec
System Element Design Input Requirements
Design
Spec

Original figure created by L. Wheatcraft.
Usage granted per the INCOSE Copyright Restrictions. All other rights reserved.

- The allocated requirements are inputs to the Lifecycle Concepts and Needs Definition activities for these entities.
  - The resulting Integrated Set of Needs are transformed into a set of Design Input Requirements for the subsystem or system element the higher-level parent requirements were allocated to.
- The allocated requirements are referred to as “parents” to the resulting “child” requirements derived to meet the intent of the allocated parent.
- When a parent is allocated to multiple lower-level entities, the is often an interaction (interface) between those entities.

Example: Medical Diagnostic System



Original figure created by L. Wheatcraft. Usage granted per the INCOSE Copyright Restrictions. All other rights reserved.

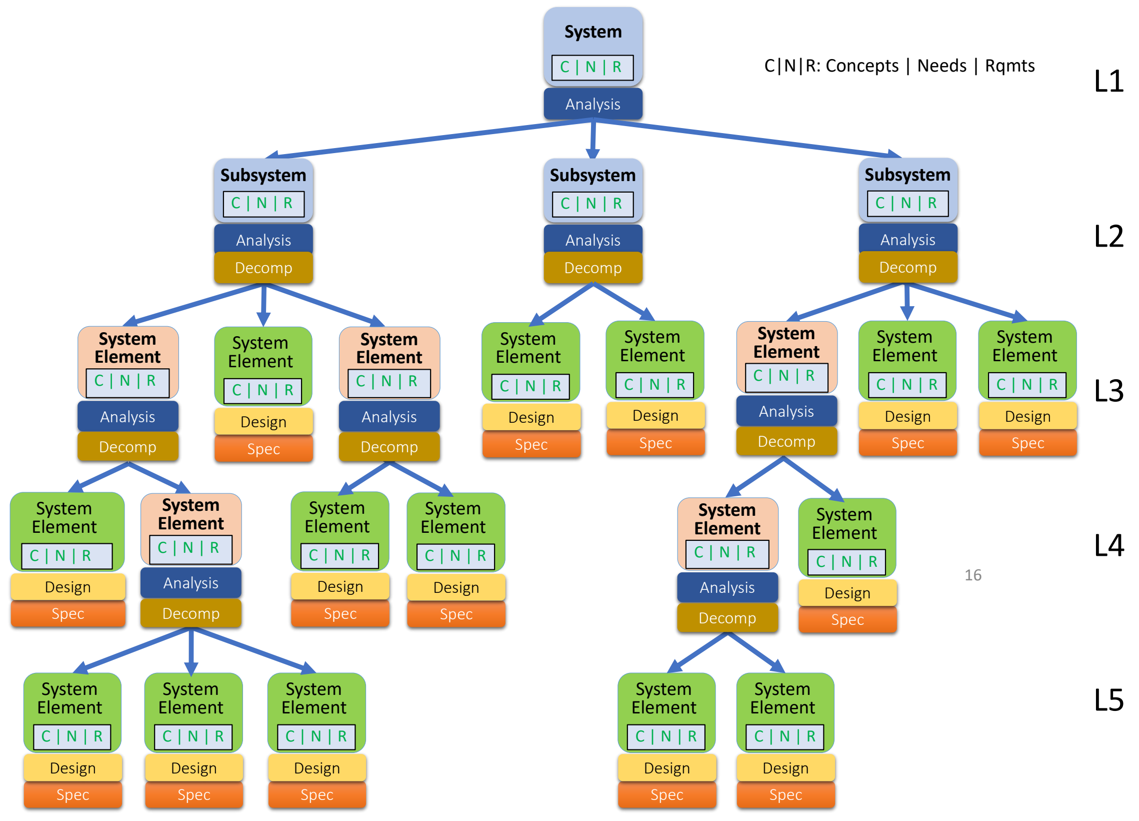
Allocation using an Allocation Matrix

Column A	Column B	Column C	Column D	Column E	Column F	Column G
System A Design Input Requirements	Instrument systems		Assay Systems		Software Systems	
	Instrument System 1	Instrument System 2	Assay System 1	Assay System 2	S/W System 1	S/W System 2
Functional/Performance					X	X
Rqmt 1	X	X			X	X
Rqmt 2		X	X	X	X	X
Operational (Fit)						
Rqmt 3	X	X			X	
Rqmt 4			X	X		X
Physical Characteristics (Form)						
Rqmt 5	X	X	X	X	X	X
Rqmt 6	X	X		X		
Quality (-ilities)						
Rqmt 7	X	X	X	X	X	X
Rqmt 8	X	X	X	X	X	X
Standards/Regulations (Compliance)						
Rqmt 9	X	X		X	X	X
Rqmt 10	X	X	X	X	X	X

Table 6-5: Example Allocation Matrix.

Section 6.4.4: Defining Child Requirements that Meet the Intent of the Allocated Parents.

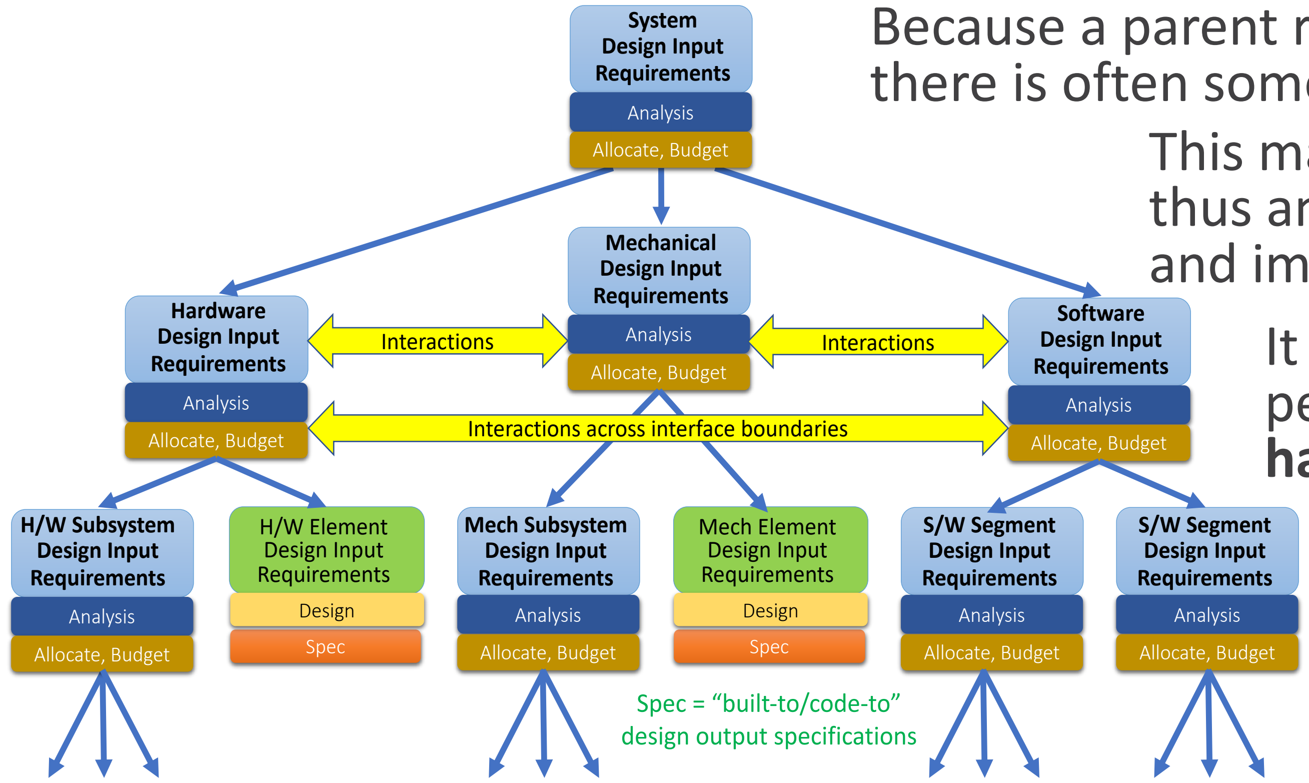
- Once the parent requirements have been allocated, they become constraints for each of the receiving entities (subsystems or system elements).
- Project teams for the receiving entity will address in their lifecycle concept analysis and maturation activities, the specific "role" they play in the implementation of the intent of each of the allocated parent requirements.



Original figure created by M. Ryan and L. Wheatcraft. Usage granted per the INCOSE Copyright Restrictions. All other rights reserved.

- The roles will be first communicated within the entity's Integrated Set of Needs and then transformed into the entity's set of design input requirements.
- The "role" must take into consideration the "role" of the other subsystems or system elements at the same level that were allocated the same parent requirement.

Dependencies and Interactions


System Design Input Requirements
Analysis
Allocate, Budget
Mechanical Design Input Requirements
Analysis
Allocate, Budget
Hardware Design Input Requirements
Analysis
Allocate, Budget
Software Design Input Requirements
Analysis
Allocate, Budget
Interactions
Interactions
Interactions across interface boundaries
H/W Subsystem Design Input Requirements
Analysis
Allocate, Budget
H/W Element Design Input Requirements
Design
Spec
Mech Subsystem Design Input Requirements
Analysis
Allocate, Budget
Mech Element Design Input Requirements
Design
Spec
S/W Segment Design Input Requirements
Analysis
Allocate, Budget
S/W Segment Design Input Requirements
Analysis
Allocate, Budget
Spec = "built-to/code-to" design output specifications

Original figure created by L. Wheatcraft. Usage granted per the INCOSE Copyright Restrictions. All other rights reserved.

Because a parent requirement is allocated to multiple entities, there is often some degree of dependency between entities.

This may involve an interaction between entities and thus an interface whose interaction must be defined and implemented.

It will be common to have a functional/ performance requirement allocated to **both hardware and software**.

Each of the child requirements for each of the receiving entities contribute to the realization of the allocated parent requirement.

Each of the entity project teams will need to determine their entity's specific role and how each entity will interact such that the intent of the allocated parent requirement is met.

This may be a new concept for software engineers experienced in developing standalone software applications, but not have experience in developing embedded software dependent on hardware systems.

When defining the child requirements DO NOT just copy and paste the parent requirement and change the noun.

Section 6.4.4: Defining Child Requirements that Meet the Intent of the Allocated Parents.

- Use of models
  - Dependencies and interactions are much easier to determine and assess within the functional, analytical, and behavioral models.
  - Dependencies and interactions will be discovered and managed within the models helping to ensure consistency of the requirements within a set as well as consistency with requirements that have a dependency with requirement sets for other subsystems and system elements.

Section 6.4.5: Budgeting of Performance, Resource, and Quality Requirements

- Allocation involves more than just “flowing down” requirements from one level to another. There are two types of allocation.
  - *Assignment of responsibility* - the receiving entity has some role in meeting the intent of the allocated parent requirements.
  - *Allocation of some quantity* - such as utilization, performance, quality, or some physical attribute.
    - Physical attributes include mass, volume, etc.
    - Performance is associated with functional requirements in terms of how well, how fast, how many, etc. For example, accuracy, precision, time, bandwidth, consumption of a consumable, or power use.
- Budgets need to be managed and controlled at the system level.
  - Results in some lower-level entities to be suboptimal in order to optimize the integrated system.
- In a data-centric practice of SE budgets are managed and controlled within a model of the system.

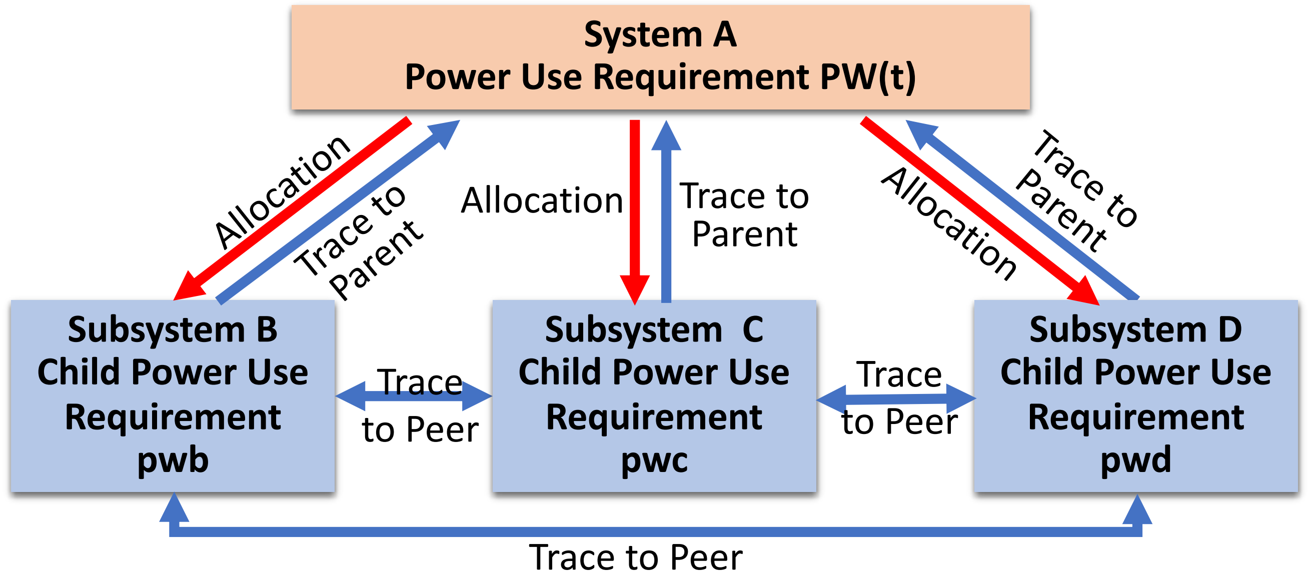
Section 6.4.5: Budgeting of Performance, Resource, and Quality Requirements

A critical concept associated with budgeting is that the budgeted quantities result in requirements that have a dependency - a change in one will result in the need to change another.

Because of these dependencies, establishing traceability between the child requirements and their allocated parent as well as between peer requirements is critical.

Because of the dependencies, it is useful to view allocation of resources as an equation.

Changes to any variables on the right side of the equation will require a change in the other variables to keep the equation balanced.



Sys A PW(t) = SSB (pwb) + SSC (pwc) + SSD (pwd)

SS = Subsystem, SE = System element   PW(t) = Maximum power use

Pwx = Allocated value

Original figure created by L. Wheatcraft. Usage granted per the INCOSE Copyright Restrictions. All other rights reserved.

Section 6.4.6: Budget Management: Margins and Reserves

- Budgets are established as limits within which a quantity is managed.
- Given there is uncertainty with the budgets, there is inherent risk to the project being able to stay within the allocated budgeted values.
- One way to help manage those risk is the use of margins and reserves.
- A major problem when defining and managing design input requirements is a failure of systems engineers to appreciate the concept of managing resource margins and reserves.
  - Development/technical margin
  - Operational margin/Operational capability
  - Management Reserve

Section 6.4.6: Budget Management: Margins and Reserves

- The size of the margins and reserves are based on the risk associated with projects.
- The early establishment of adequate margins and reserves and the effective management of them throughout the project's lifecycle play a critical role in the ability of the project to deliver a winning system.
- Failing to define margins places the project at great risk of cost overruns and schedule slips.
- When defining the values within the design input requirements, it is critical these values take into consideration the margins and reserves defined and being managed by the project.

Section 6.4.7: Use of Traceability and Allocation to Manage Requirements

- Combining the concept of allocation with the concept of traceability provides the project team a powerful way to manage the design input requirements, especially across levels and across subsystems and system elements within a specific level.
- As requirements are developed and flow down from one level to another, it is critical that allocation and traceability is assessed for completeness, correctness, and consistency.
- These assessments are not only needed while defining the sets of design input requirements but is a major function of managing the sets of needs and sets of design input requirements, especially when assessing changes.

Section 6.4.7: Use of Traceability and Allocation to Manage Requirements

- Common issues.
  - Requirements not allocated
  - Requirements not allocated correctly
  - Parent requirements with no child requirements
  - Needs with no implementing design input requirements
  - Orphan needs that do not trace to a source.
  - Orphan requirements that do not trace a need, parent, or a source
  - Needs, sources, or requirements with incorrect or missing implementing child requirements.
  - Requirements with an incorrect parent or source.
  - Sets of child requirements are not necessary and sufficient to implement the parent requirement, need, or source from which it was transformed/derived.

This are discussed in detail within Section 14, Managing Needs and Requirements

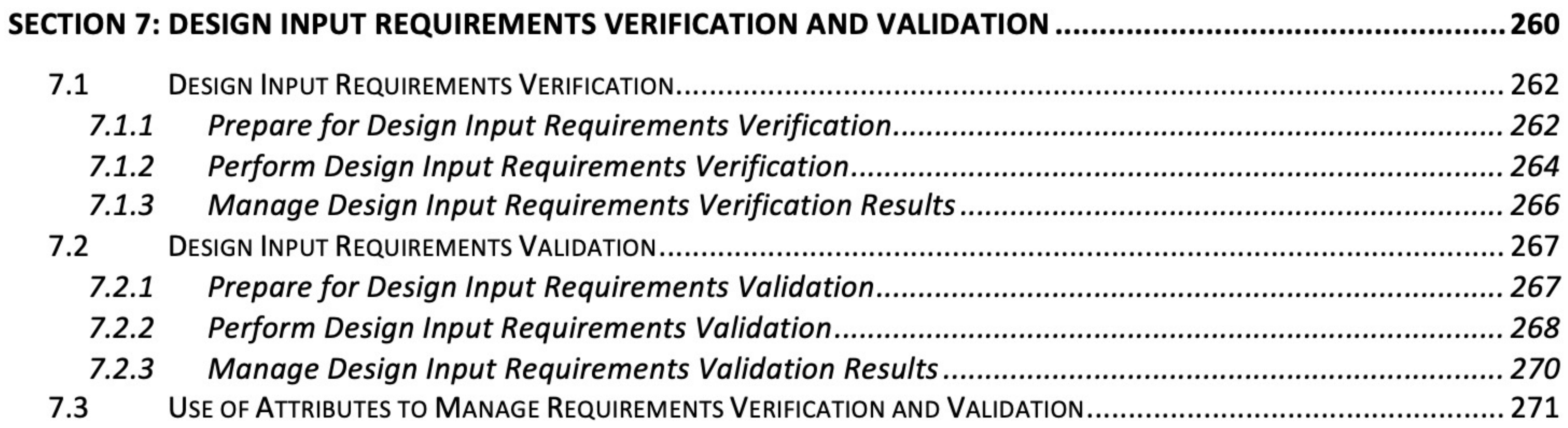
Section 6.5: Summary of Design Input Requirements Definition

- The focus of the Design Input Requirements Definition activities is to define well-formed sets of design input requirements for the integrated system as well as each subsystem and system element within the physical architecture of the SOI.
  - These sets of design input requirements represent the allocated baseline to which the Design Definition Process will implement.
- Well-formed refers to the quality of the sets of design input requirements in terms of content as well as the structure as defined in the INCOSE GtWR.
- The goal is to have well-formed sets of design input requirements that clearly communicate the intent of the needs to users of the requirements.
- Ensuring the sets of design input requirements have the characteristics of well-formed design input requirements as defined in the INCOSE GtWR is necessary to ensure high-quality, requirements that will not be as volatile as many organizations currently experience.

Section 6.5: Summary of Design Input Requirements Definition

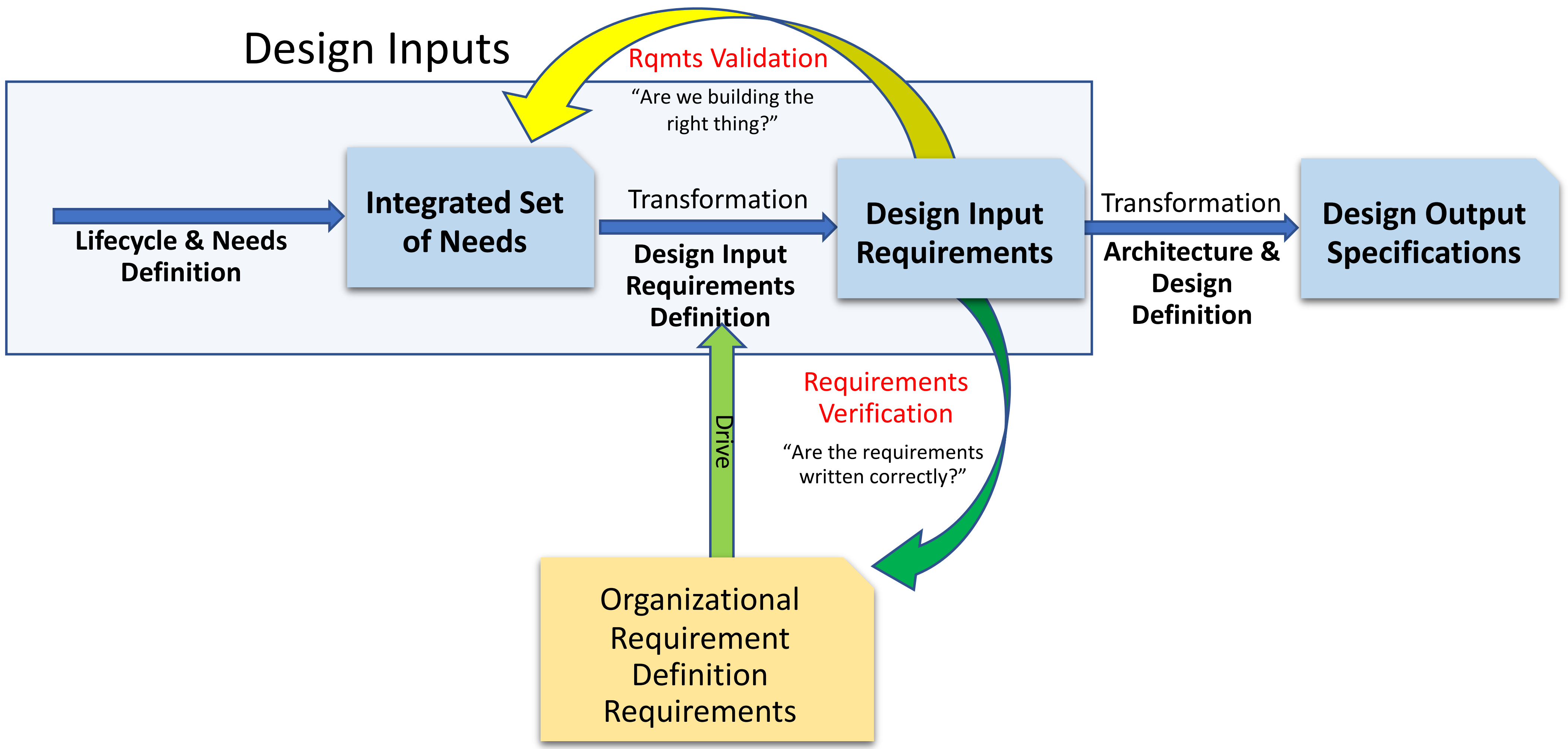
- Projects often ask the question: “How do we know requirements are “done” enough to proceed with design?”
  - There is always a trade-off between “better” and “good-enough”.
  - One definition of “good enough” is: the point where the cost of potential changes is less than the effort needed to define every requirement.
  - There really is not a simple indicator, the decision should be knowledge driven and not schedule driven.
- Baselining poorly formed sets of design input requirements often results in the accumulation of technical debt that will be more costly in terms of both schedule and budget rather than spending the time and effort that will result in well-formed sets of design input requirements as discussed in this guide.

Table of Contents



- Do the individual requirement expressions have the characteristics defined in the INCOSE GfWR: *Necessary*, *Appropriate*, *Unambiguous*, *Complete*, *Singular*, *Feasible*, *Verifiable*, *Correct, and Conforming?*
- Does the set of design input requirements have the characteristics defined in the INCOSE GfWR: *Complete*, *Consistent*, *Feasible*, *Comprehensible*, *Able to be validated?*
- Have attributes been defined for each requirement statement?
- Are all requirement statements traceable to their parent, source, or dependent peer requirements?
- Does the set of design input requirements communicate the intent of the integrated set of system needs from which they were transformed?

Requirements Verification and Validation


Design Inputs
Rqmts Validation
"Are we building the right thing?"
Lifecycle & Needs Definition
Integrated Set of Needs
Transformation
Design Input Requirements Definition
Design Input Requirements
Transformation
Architecture & Design Definition
Design Output Specifications
Drive
Requirements Verification
"Are the requirements written correctly?"
Organizational Requirement Definition Requirements

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

Questions and Discussion

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.