Requirements: The Essentials

Sponsored by the INCOSE Los Angeles Chapter

Rick Hefner, PhD, Caltech CTME
rhefner@caltech.edu

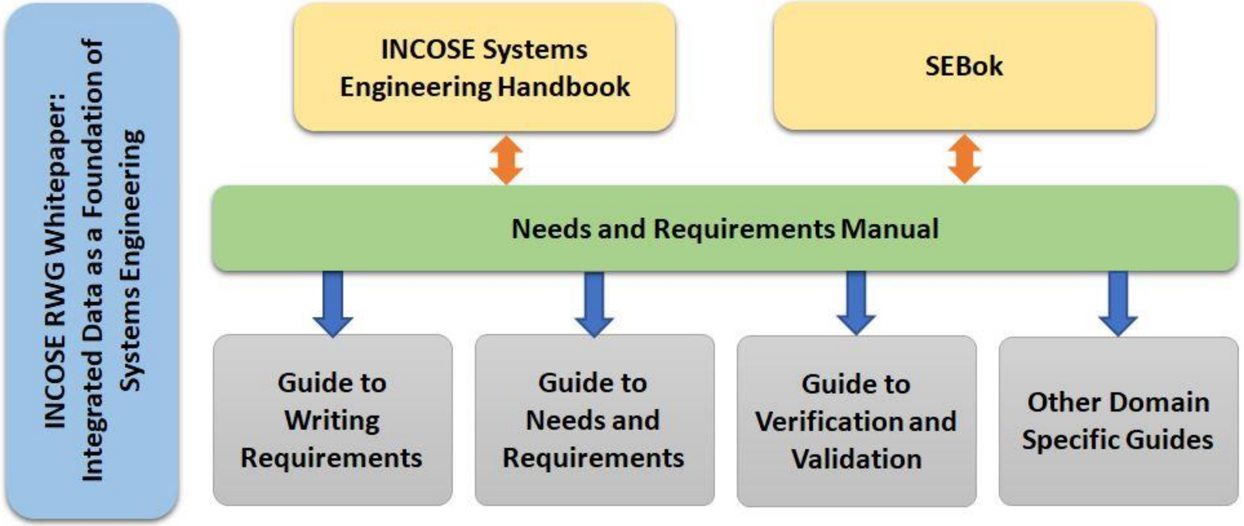
Background

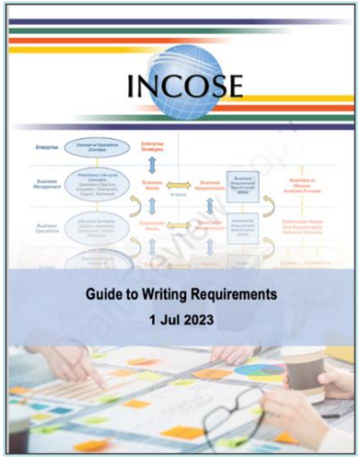
- Requirements are one of the most critical aspects of modern systems development and one of the least understood
- This presentation provides a comprehensive overview of industry best practices for Requirements Engineering, covering elicitation, analysis, validation, specification, allocation, and verification
- Attendees will leave with a thorough understanding of techniques and tools for handling common requirements challenges

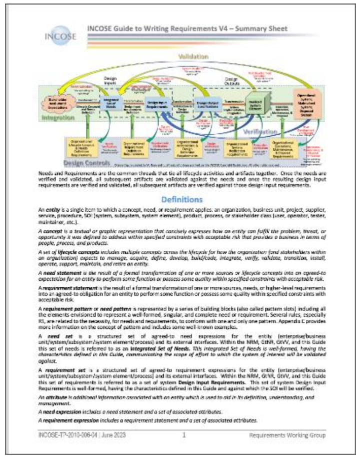
Material is taken from Caltech professional training (http://ctme.caltech.edu) and INCOSE publications (e.g., SEBoK, http://sebokwiki.org)

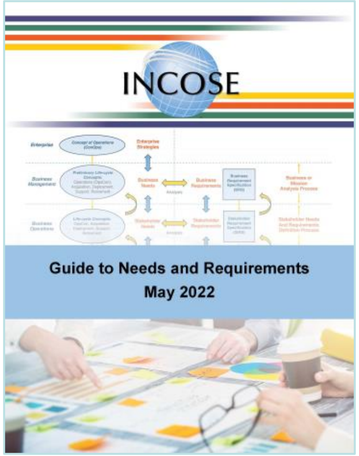
INCOSE Requirement Working Group (RWG)

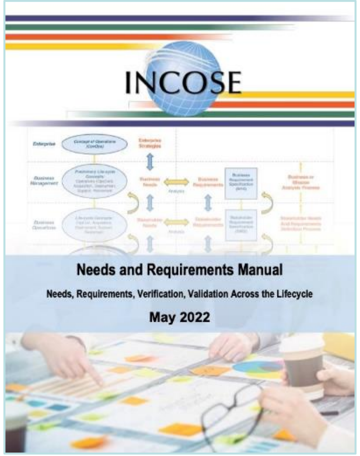
https://www.incose.org/communities/working-groups-initiatives/requirements

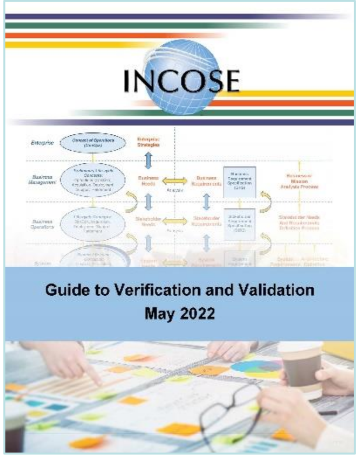








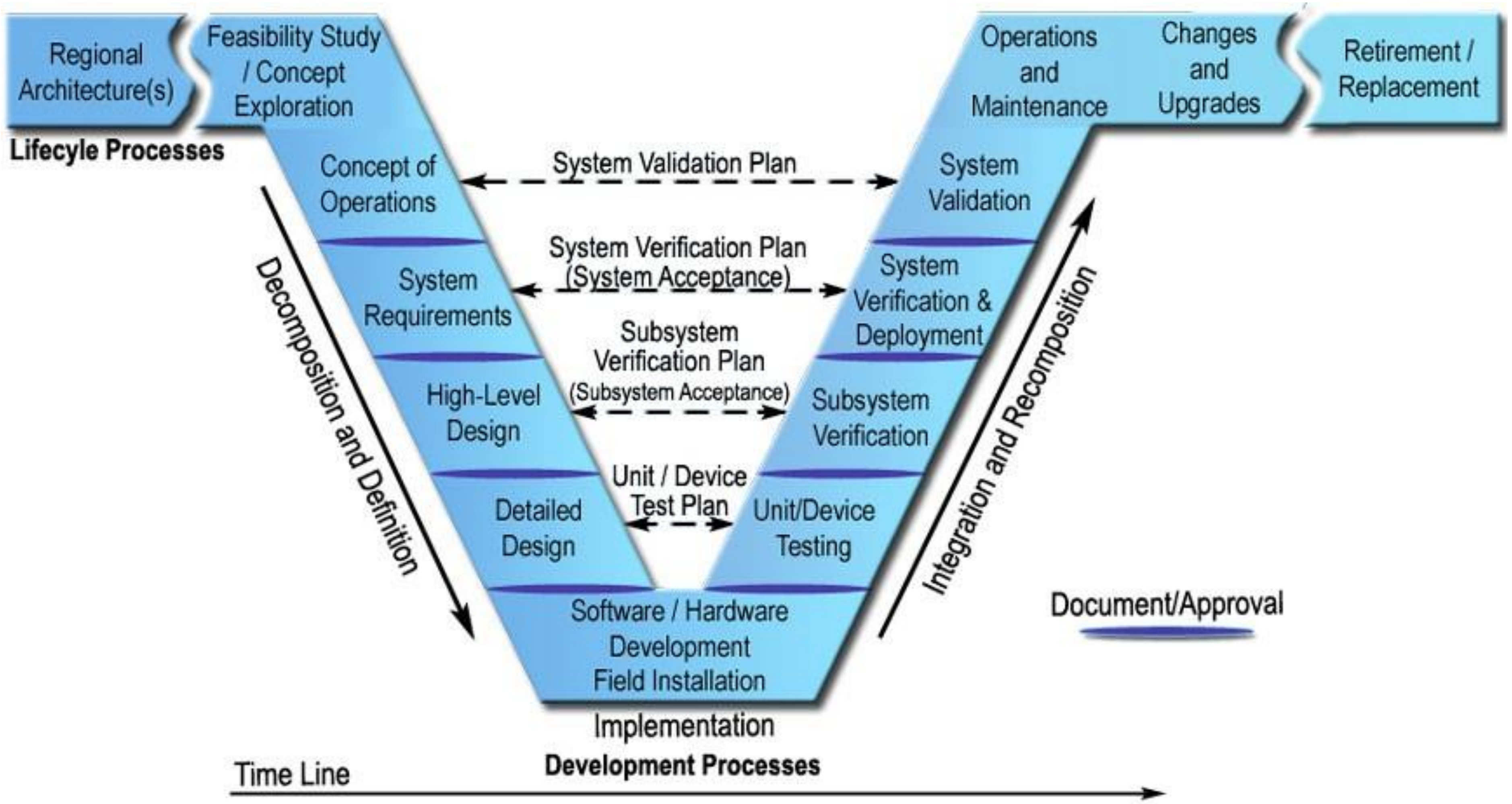




Topics

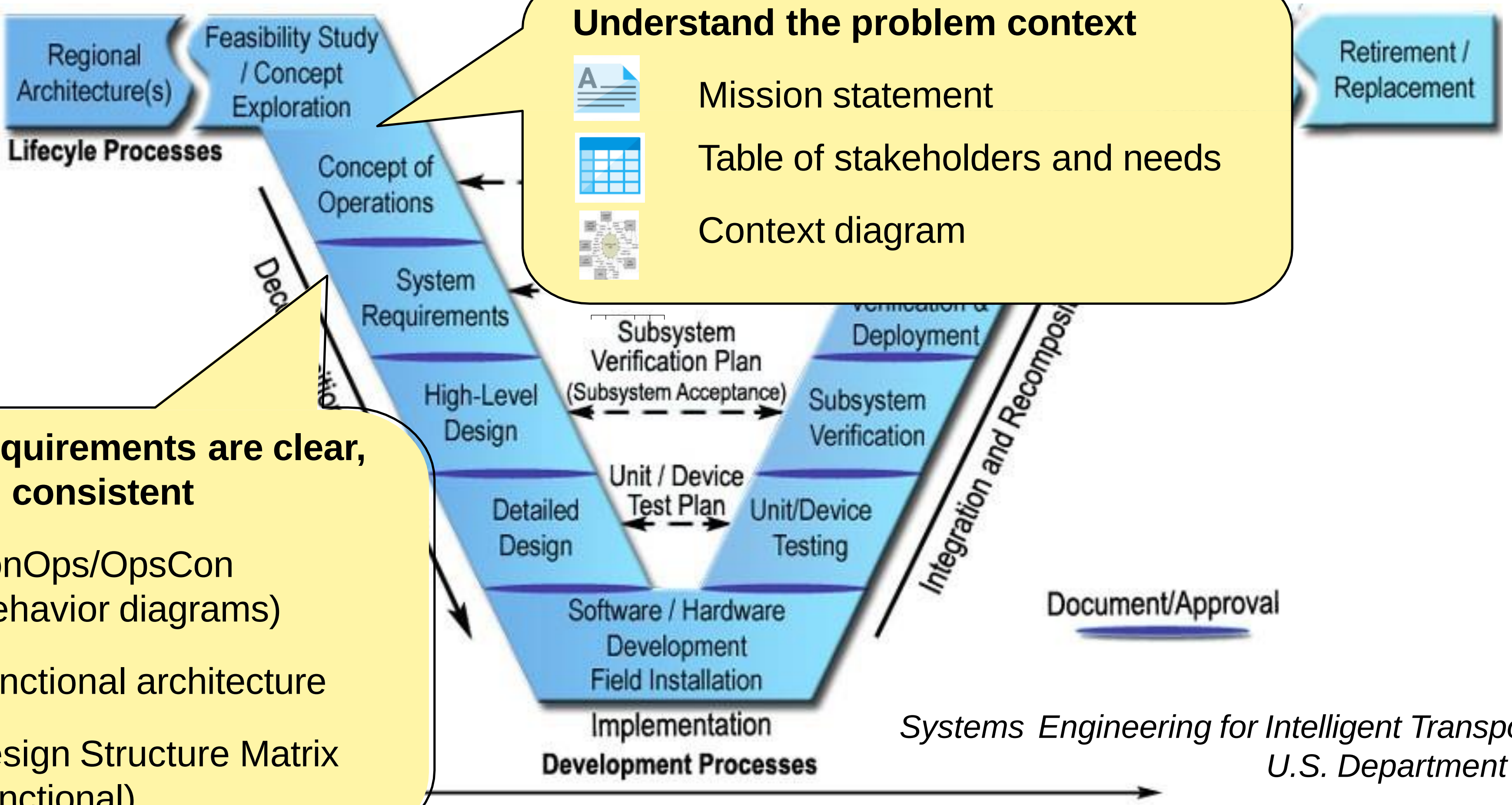
- **Introduction to Requirements Engineering**
- Requirements Elicitation
- Requirements Analysis
- Requirements Validation
- Requirements Specification and Allocation
- Requirements Verification
- MBSE Approaches to Requirements
- Keys to Requirements Engineering

The Systems Engineering V-Model



Systems Engineering for Intelligent Transportation Systems, U.S. Department of Transportation

Key SE Representations and Their Purpose



Understand the problem context

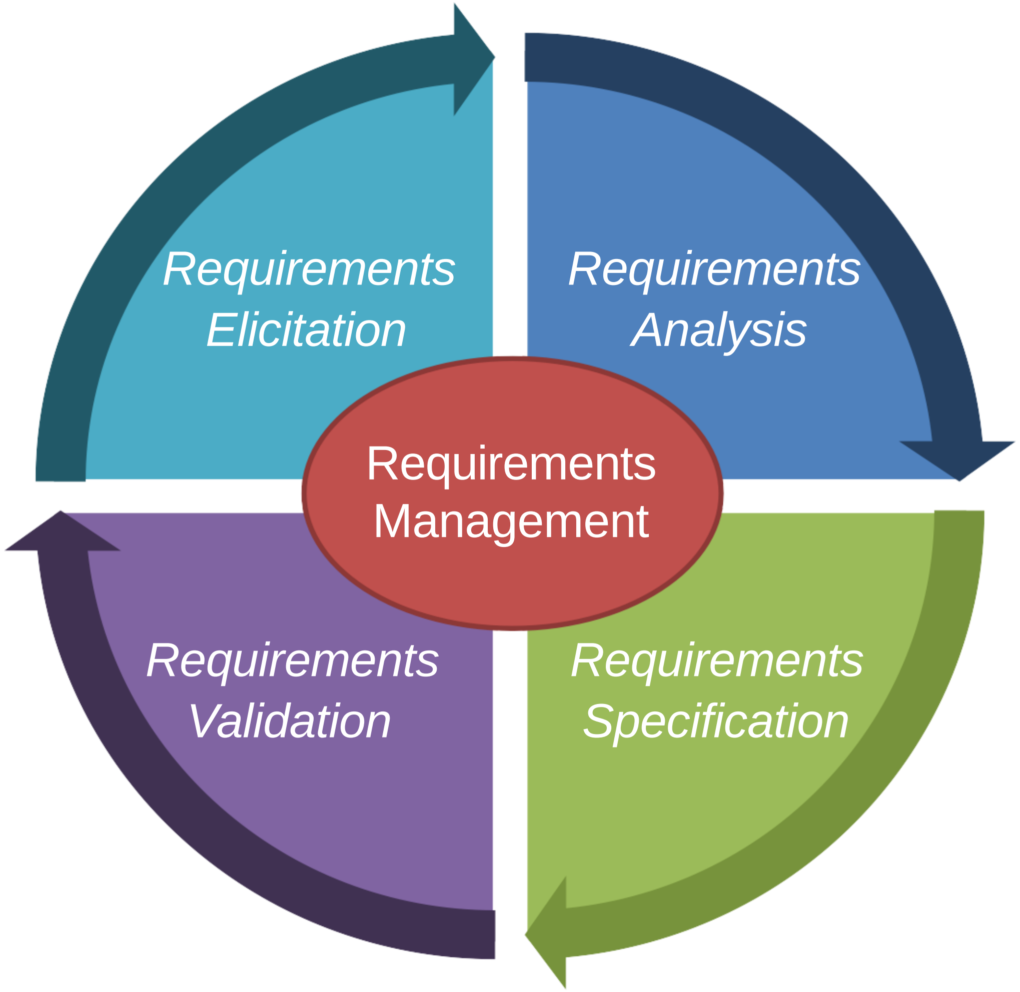
- Mission statement
- Table of stakeholders and needs
- Context diagram

Ensure requirements are clear, complete, consistent

- ConOps/OpsCon (behavior diagrams)
- Functional architecture
- Design Structure Matrix (functional)

Systems Engineering for Intelligent Transportation Systems, U.S. Department of Transportation

Requirements Engineering is An Iterative Process

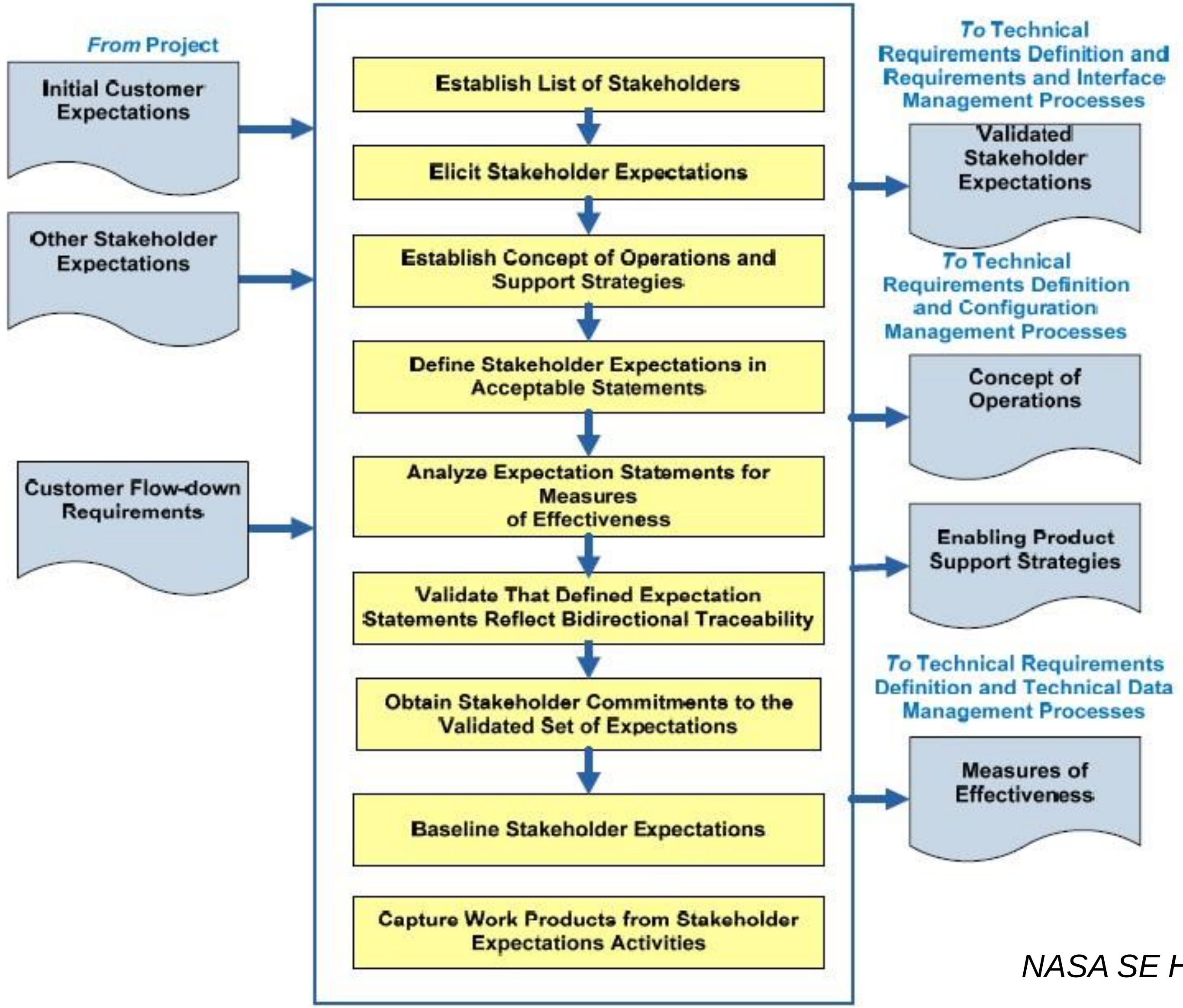


Discussion: What Requirements Challenges Have You Experienced?

Topics

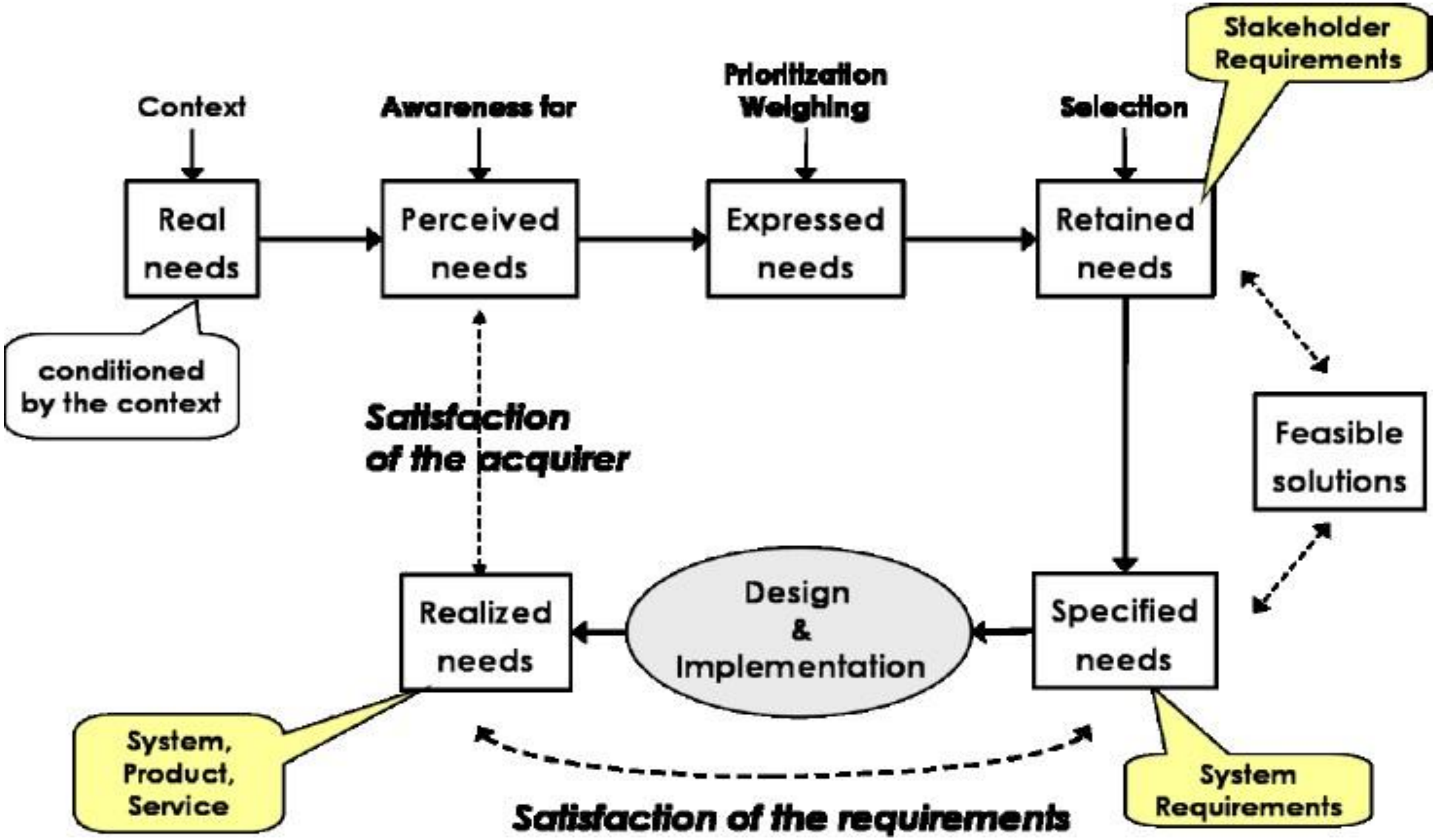
- Introduction to Requirements Engineering
- **Requirements Elicitation**
- Requirements Analysis
- Requirements Validation
- Requirements Specification and Allocation
- Requirements Verification
- MBSE Approaches to Requirements
- Keys to Requirements Engineering

Stakeholder Expectations Definition Process



NASA SE Handbook

Cycle of Stakeholder Needs



Cited in SEBoK. Permission granted by Singery'Com

Topics

- Introduction to Requirements Engineering
- Requirements Elicitation
- **Requirements Analysis**
- Requirements Validation
- Requirements Specification and Allocation
- Requirements Verification
- MBSE Approaches to Requirements
- Keys to Requirements Engineering

Example Types of Requirement Categories

Derived from the INCOSE Needs and Requirements Manual

Category	Description
Function/ Performance	The primary functions and associated performance that the SoI needs to perform in terms of its intended use. The functions address the capabilities and features the stakeholders expect the SoI to have; performance addresses how well, how many, how fast attributes of the function. Many of the primary functions involve interactions (interfaces) between the SOI and systems external to the SOI. All critical and high priority needs would be included in this category.
Fit/Operational	Requirements dealing with functions that deal with a secondary or enabling capabilities, functions, and interactions between the SoI and external systems needed for the system to accomplish its primary functions. This includes functions concerning the ability of the system to interface with, interact with, connect to, operate within, and become an integral part of the macro system it is a part. Fit includes human system interactions and interfaces as well as both the induced and natural **environments** (conditions of operations, transportation, storage, maintenance). For example, needs associated with safety, security, power, cooling, transportation and handling, storage, maintenance, and disposal.
Form	Physical Characteristics. The shape, size, dimensions, mass, weight, and other observable parameters and characterizes that uniquely distinguish a system. For software, form could address programming language, lines of code, memory requirements.
Quality	Fitness for use. For example, various “-ilities” such as reliability, testability, operability, availability, maintainability, operability, supportability, manufacturability, and interoperability.
Compliance	Conformance with design and construction standards and regulations.

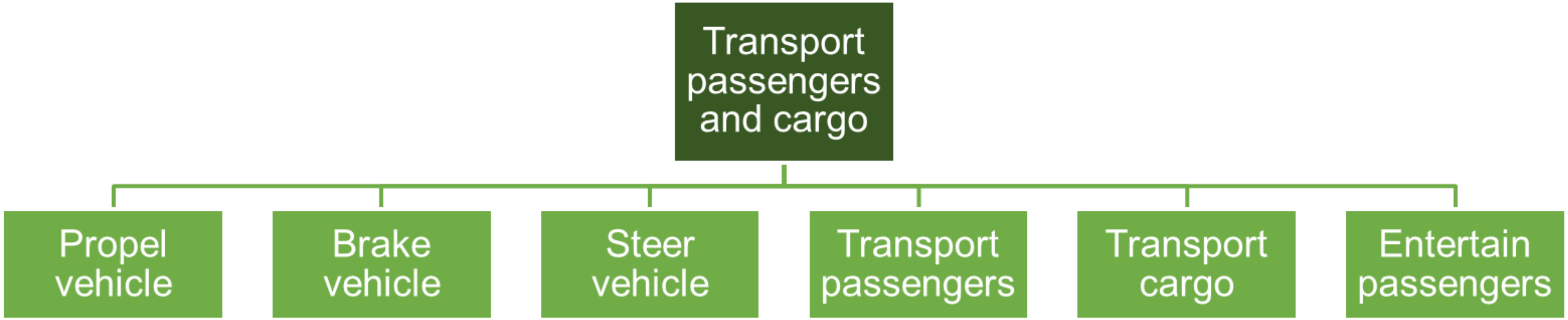
Product Requirement Types in Industry

Functional • What must the system do?	**Performance** • How well must it be done?
Design Constraints • What design characteristics must be followed or achieved? • Typically set by a higher authority for business reasons	**Quality Attributes** • How will the users determine the quality of the system, given the other requirements? • Usability, maintainability, etc. • Specification must include agreement on how it will be measured

Requirements – Passenger Auto

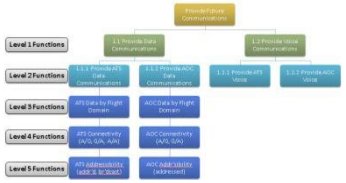
Requirement
Functional: The auto shall be capable of travelling in reverse.
Performance: The auto shall be capable accelerating from a stop to 60 mph in 10 seconds.
Design Constraint: The auto shall use Firestone tires.
Quality Attribute: The auto shall be highly reliable. > Overall reliability of the auto shall be .999 as measured by standard XXX.

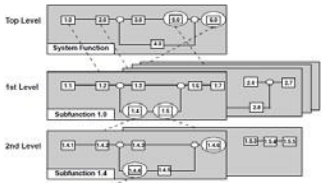
Functional Hierarchy – Passenger Car

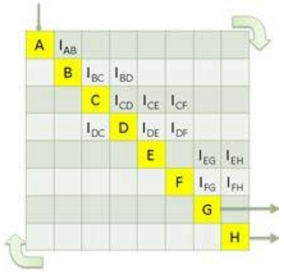


Example Functional Analysis Methodologies

- Functional hierarchy
  - Use to describe top-down definition of system functions
- Functional flow block diagram
  - Used to show the sequence of all functions to be accomplished by a system
- Design Structure Matrix (DSM)
  - Used to develop functional or physical interfaces







By iterating among the different representations, the list of functions can be checked for completeness and consistency

Topics

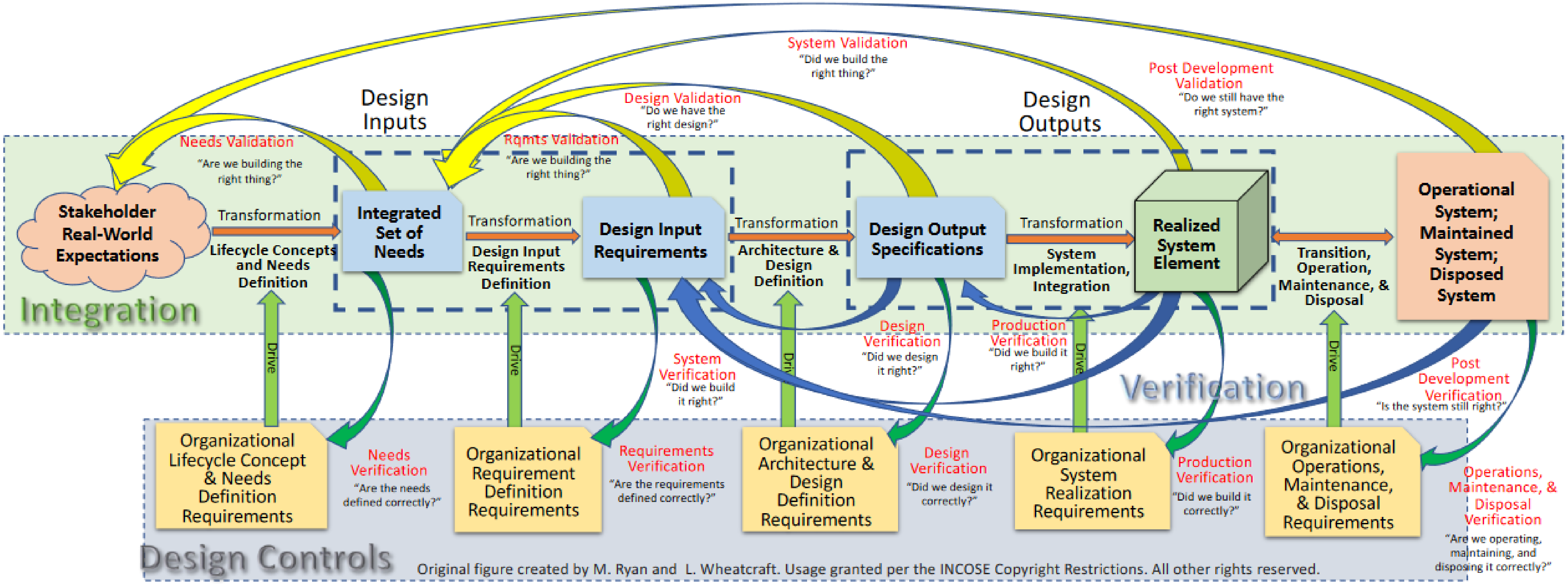
- Introduction to Requirements Engineering
- Requirements Elicitation
- Requirements Analysis
- **Requirements Validation**
- Requirements Specification and Allocation
- Requirements Verification
- MBSE Approaches to Requirements
- Keys to Requirements Engineering

Verification and Validation

- **Verification** – proving the specified requirements have been met (Did we build the system right?)
- **Validation** – determining to what extent the user needs have been met (Did we build the right system?)

- These definitions reflect the industry consensus – some texts (and organizations) use these terms differently/backwards!

Verification and Validation (Incremental and End-Item)



INCOSE Guide for Writing Requirements

Requirement Validation Activities

- Conduct requirements reviews to validate that requirements are correct, unambiguous, complete, consistent, ranked for importance, verifiable (testable), modifiable, and traceable
- Use prototyping to demonstrate assumptions and confirm mutual interpretations
- Validate the concept of operations developed during analysis
- Plan how each requirement will be verified (establish acceptance tests)

How would the dealer validate the passenger car requirements?

Topics

- Introduction to Requirements Engineering
- Requirements Elicitation
- Requirements Analysis
- Requirements Validation
- **Requirements Specification and Allocation**
- Requirements Verification
- MBSE Approaches to Requirements
- Keys to Requirements Engineering

TABLE OF CONTENTS

3 ATS SERVICE REQUIREMENTS

3.1 Flight Planning Services

3.1.1 Flight Plan Support

Safe and efficient use of the nation's airspace requires the provision of flight planning capabilities. Flight planning requires information such as expected route, altitude, and time of flight, as well as the anticipated flight conditions including weather, navigation systems, available routes, special use airspace, and flow control conditions.

3.1.1.1 Accept proposed flight plans

The NAS shall accept VFR flight plans. *(00340)*
The NAS shall accept IFR flight plans. *(00350)*
The NAS shall accept flight plans up to 24 hours in advance of proposed departure time. *(00380)*
The NAS shall accept proposed flight plans from external interfaces. *(00430)*
The NAS shall accept flight plans from specialist. *(00550)*
The NAS shall accept proposed flight plans from specialists. *(00190)*
The NAS shall accept amendments to proposed flight plans from users. *(00200)*
The NAS shall accept amendments to proposed flight plans from specialists. *(00210)*
The NAS shall acquire flight plan reservations from military users. *(16060)*
The NAS shall acquire requests for special movement activities by military aircraft. *(16270)*
The NAS shall accept flight plans in NAS format. *(00260)*
The NAS shall accept flight plans in ICAO format. *(00290)*

3.1.1.1.1 Accept altitude reservation requests

The NAS shall accept altitude reservation requests. *(15740)*
The NAS shall acquire altitude reservations. *(15760)*
The NAS shall store altitude reservations. *(15770)*

Sample Specification: National Airspace System

Requirement Syntax

<requiree> shall <active verb> <object> <qualifier>

- Example: The autonomous taxi shall unlock the passenger door(s) upon arriving at the destination.

- <requiree> is typically the system-of-interest
- The optional <qualifier> may identify when/where/how the requirement applies (e.g., “upon arriving at the destination”)
- One requirement per sentence, straightforward language
- “Shall”, not “will” or “must” (convention)
- “Shall not” is permissible
- “Should” or “may” indicate preferences, but are not binding

Characteristics of Good Requirements

Individual Requirements

- C1- Necessary
- C2 – Appropriate
- C3 – Unambiguous
- C4 – Complete
- C5 – Singular
- C6 – Feasible
- C7 – Verifiable
- C8 – Correct
- C9 - Conforming

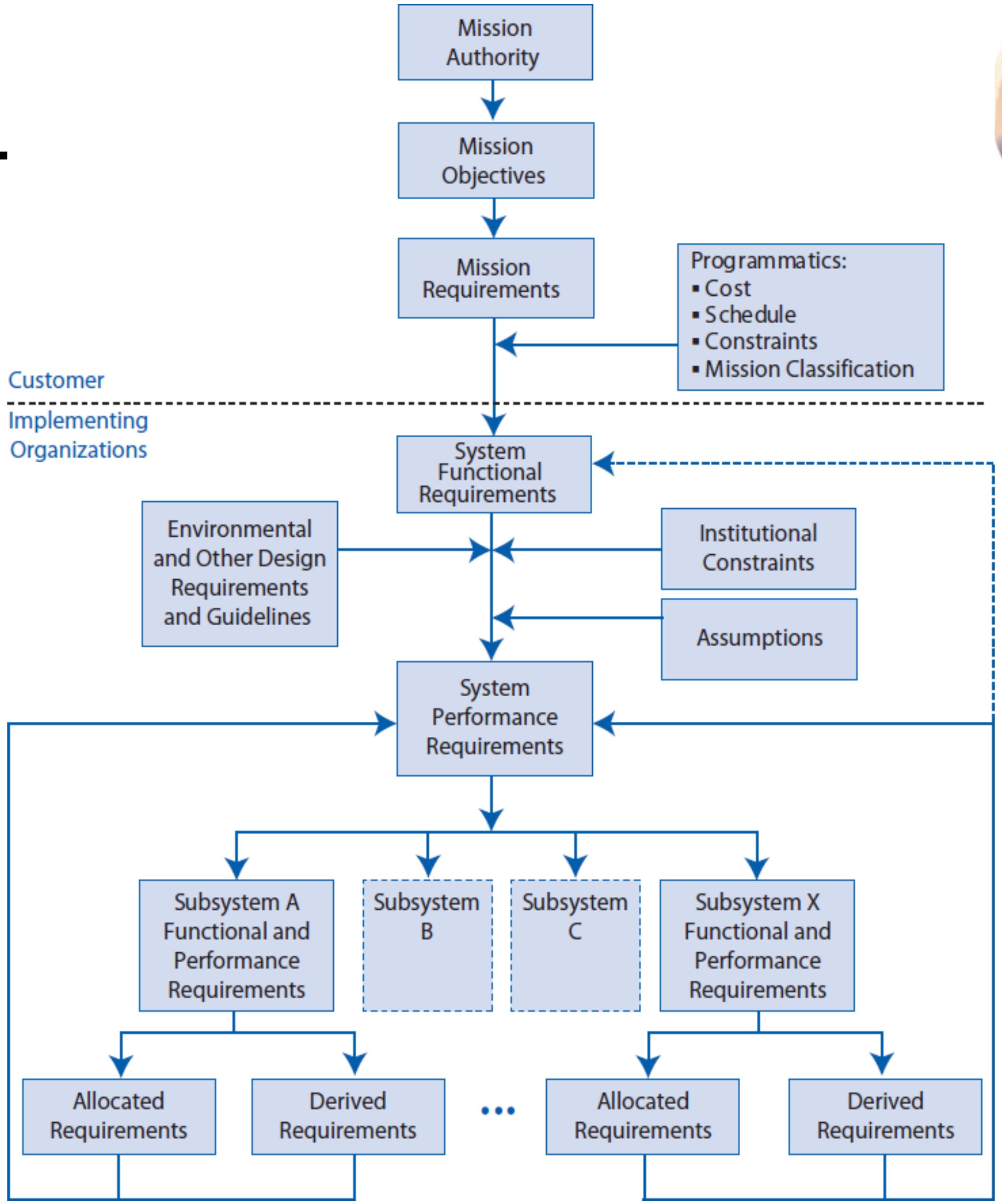
Sets of Requirements

- C10 – Complete
- C11 – Consistent
- C12 – Feasible
- C13 – Comprehensible
- C14 – Able to Be Validated

INCOSE Guide for Writing Requirements

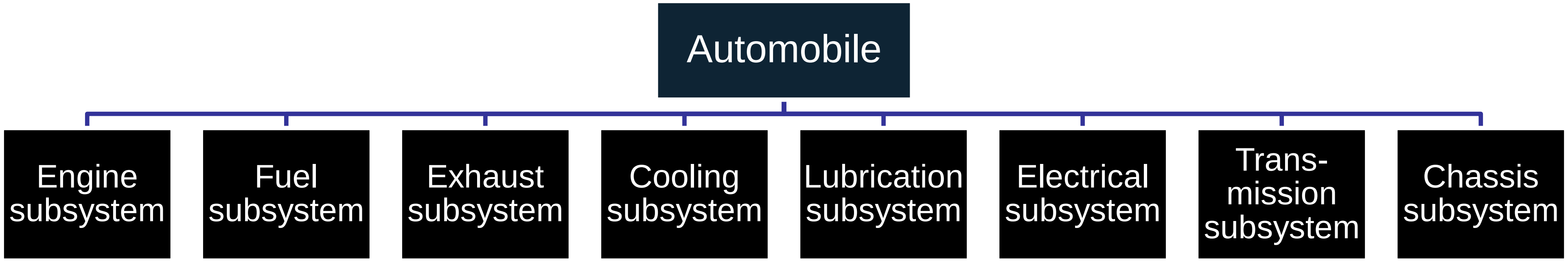
Requirements Allocation

- Requirements are decomposed in a hierarchical structure
- High-level requirements are decomposed and allocated to the design elements – if each element can meet its allocated requirements, the top-level system will meet its requirements
- The process is repeated, as requirements are further decomposed and allocated among the elements and sub- elements

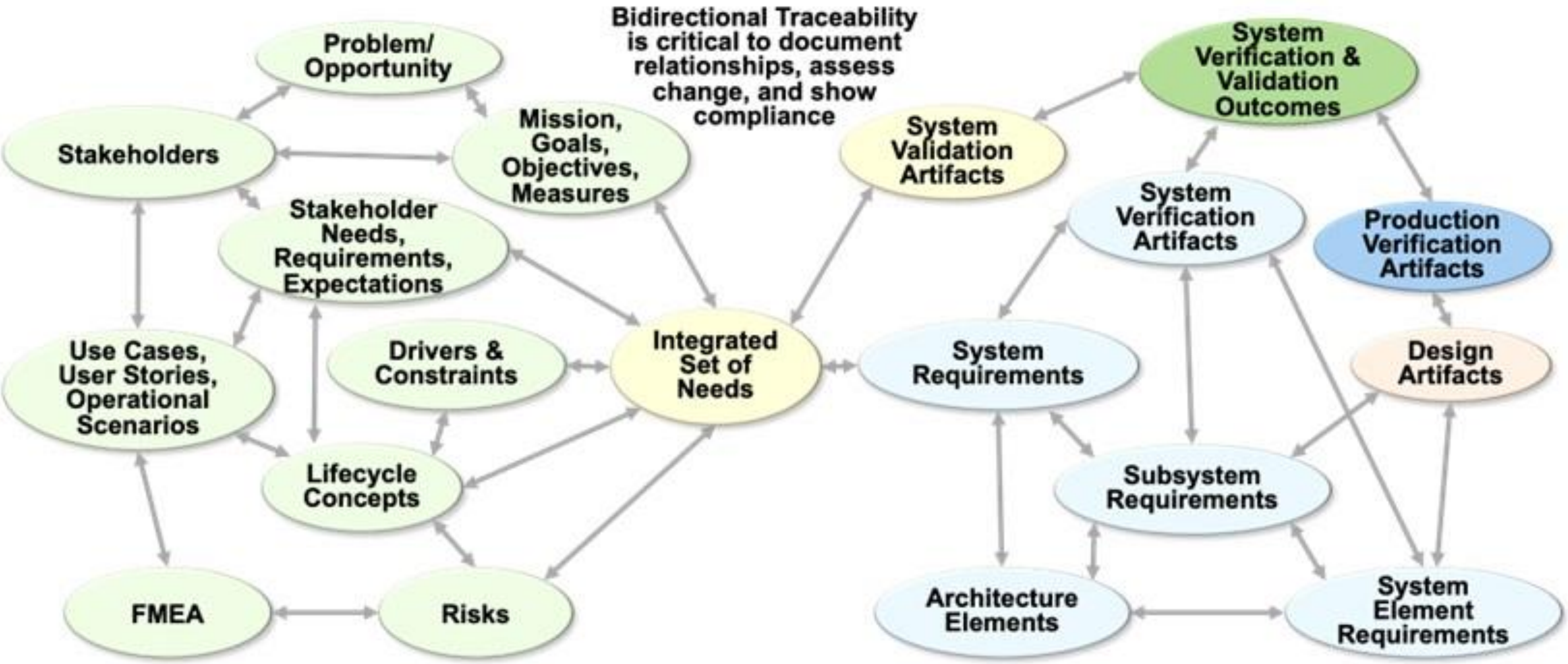


NASA Systems Engineering Handbook

Requirements Allocation – Passenger Car



Requirements Traceability



Original figure created by L. Wheatcraft.

SEBoK

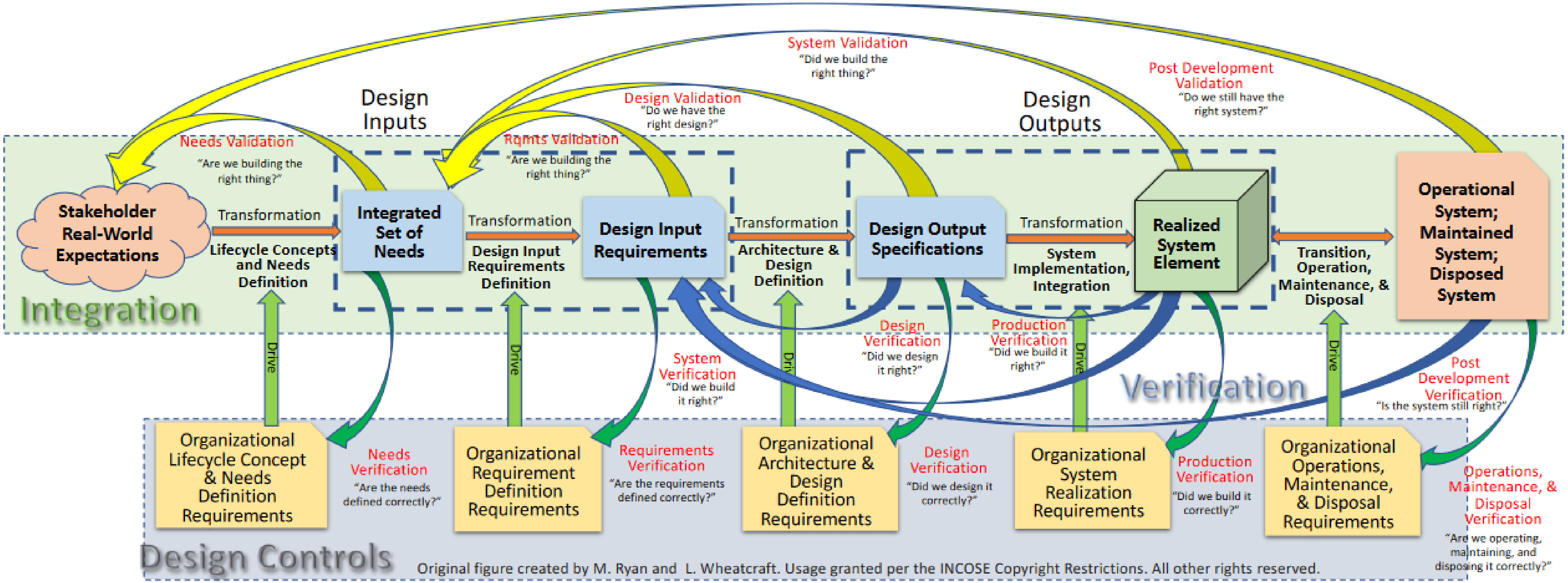
Topics

- Introduction to Requirements Engineering
- Requirements Elicitation
- Requirements Analysis
- Requirements Validation
- Requirements Specification and Allocation
- **Requirements Verification**
- MBSE Approaches to Requirements
- Keys to Requirements Engineering

Verification and Validation

- **Verification** – proving the specified requirements have been met (Did we build the system right?)
- **Validation** – determining to what extent the user needs have been met (Did we build the right system?)

Verification and Validation (Incremental and End-Item)



INCOSE Guide for Writing Requirements

Requirements Types and Typical Verification Methods

Functional What must the system do?	**Demonstration** Use of system, subsystem, or component operation to show that a requirement can be achieved
Performance How well must it be done?	**Test** Use of system, subsystem, or component operation to obtain detailed data to verify performance or to provide sufficient information to verify performance through further analysis
Design Constraints What design characteristics must be followed or achieved?	**Inspection** Visual examination
Quality Attributes How will the users determine the quality of the system, given the other requirements?	**Analysis** Use of mathematical modeling and analytical techniques to predict the compliance of a design to its requirements based on calculated data or data derived from lower level component or subsystem testing

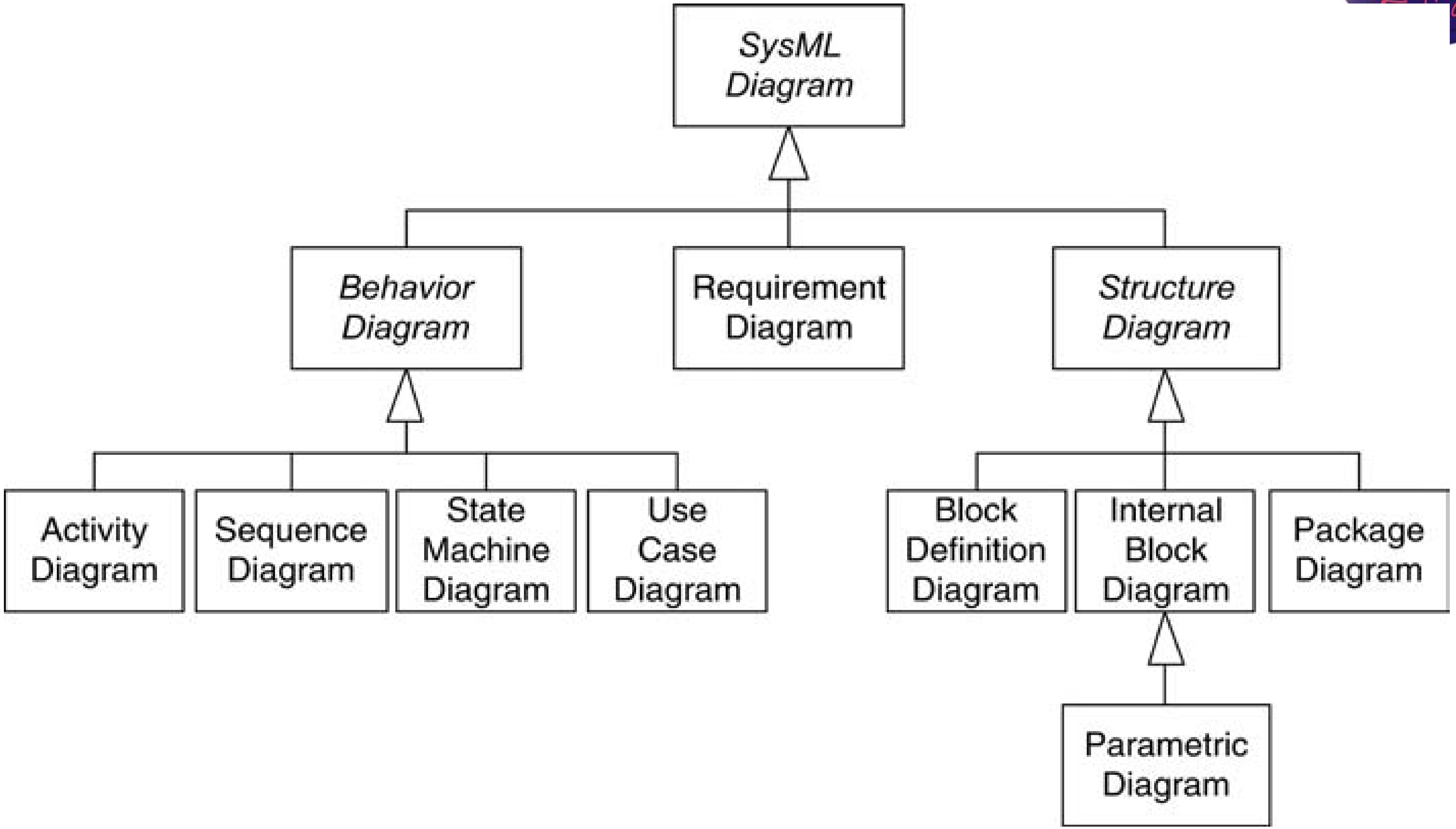
Verification Example – Passenger Car

Requirement	Verification
Functional: The auto shall be capable of travelling in reverse.	**Demonstration:** A demo where you simply show the auto can travel in reverse.
Performance: The auto shall be capable accelerating from a stop to 60mph in 10 seconds.	**Test:** The auto's acceleration is measured with a stopwatch.
Design Constraint: The auto shall use Firestone tires.	**Inspection:** Observe the tires used on the car.
Quality Attribute: The auto shall be highly reliable. > Overall reliability of the auto shall be .999 as measured by standard XXX.	**Analysis:** A model is built according to standard XXX, calibrated to physical measurements, and used to compute an reliability number.

Topics

- Introduction to Requirements Engineering
- Requirements Elicitation
- Requirements Analysis
- Requirements Validation
- Requirements Specification and Allocation
- Requirements Verification
- **MBSE Approaches to Requirements**
- Keys to Requirements Engineering

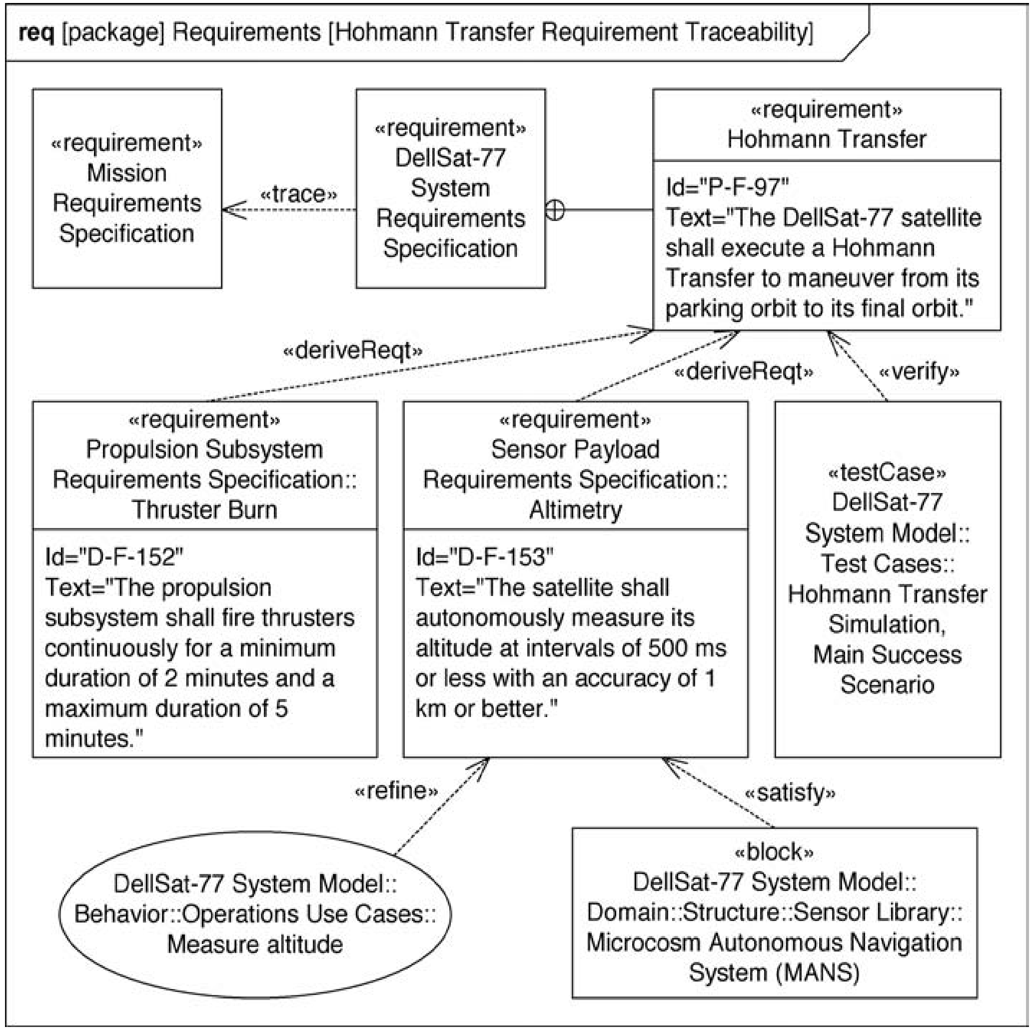
SysML Diagram Taxonomy



Requirements Diagram

Used to display text-based requirements, the relationships between requirements, and the relationships between requirements and other model elements

- **Trace** - A modification to the supplier element (↑) *may* result in the need to modify the client element (tail end)
- **Derive** – Client requirement is derived from the supplier requirement
- **Refine** – Client element is more concrete (i.e., less abstract) than the supplier element
- **Satisfy** – Client requirement is fulfilled by the supplier element
- **Verify** – Client requirement is verified by the supplier test case



Topics

- Introduction to Requirements Engineering
- Requirements Elicitation
- Requirements Analysis
- Requirements Validation
- Requirements Specification and Allocation
- Requirements Verification
- MBSE Approaches to Requirements
- **Keys to Requirements Engineering**

Keys to Requirements Engineering

1. Good requirements engineering takes more time and attention than expected
2. Ensure you understand the problem context and identify all the stakeholders before writing requirements
3. Use requirements analysis and validation to ensure a complete, correct, and consistent set of requirements
4. Pay attention to the type and syntax of the requirements
5. Traceability is essential – each requirement should link back to a stakeholder need, and lower-level requirements should link to upper-level requirements