DEFINITIONS AND STRUCTURE OF REQUIREMENTS


WALLY, WE DON'T HAVE TIME TO GATHER THE PRODUCT REQUIRE-MENTS AHEAD OF TIME.
I WANT YOU TO START DESIGNING THE PRODUCT ANYWAY. OTHERWISE IT WILL LOOK LIKE WE AREN'T ACCOMPLISHING ANY-THING.
OF ALL MY PROJECTS, I LIKE THE DOOMED ONES BEST.
NEWS
www.unitedmedia.com
S. Adams
5/9/97 © 1997 United Feature Syndicate, Inc.

W	Dates	Topic	Ref	Activity
35	26-Aug	Course Introduction Systems Engineering Review Requirement Engineering (Generic Process)	[1][6] [7][9]	Groups & Names
36	2-Sep	Lego exercise 01 - First assessment	[2] [10]	Exercise Report
37	9-Sep	Definition and Structure of Requirements Functional and non-functional Requirements	[8][6] [10]	Exercise List
38	16-Sep	Elicitation Techniques (Interview / Rich Picture / use Cases)		Group Project Delivery
39	23-Sep	Rules for need and requirement statements	[2][3]	Exercise List
40	30-Sep	Attributes of need and requirement statements Boilerplates/patterns	[2]	Exercise List
41	7-Oct	[Carrer Explo]		* Reading activity
42 *	14-Oct	Midterm Session		Self-paced concept questions
		[Fall break]		

W	Dates	Topic	Ref	Activity
43	21-Oct	MBSE Trends	**	Group Project Delivery
44	28-Oct	Traceability - Organization	[5][	Group Project Delivery
45	4-Nov	Metadata: Rationale + Verification and Validation (Success Criteria)	[2][4]	Group Project Delivery
46	11-Nov	[Veterans Day]		* reading activity
47	18-Nov	Lego Exercise 02	[2] [10]	Exercise Report
48	25-Nov	System Requirement Review Presentation (G1-G4)		
		[Thanksgiving]		
48	2-Dec	System Requirement Review Presentation (G5-G8)		
49	9-Dec	Finals		Self-paced concept questions and a small case

REQUIREMENT IMPORTANCE

Importance of having good requirements

- Requirements tell you **what the system needs to do** (functional requirements).
- **How well** the system needs to do it (performance requirements)
- **What environment** the system has to work in (environmental requirements).
- What the system **must do to fit into the bigger system** (interface requirements).
- What **lower level subsystems/assemblies/components must do to fit** into the system and make it all work (allocation of requirements/resources).
- What you need to **do before you fly** (verification activities).
- And basically, **when you are done** (requirements are met).



Requirements are essential to:

- To show **results** the user want from the system.
- To show **traceability back to sources** and the history of changes.
- To show what the organization **needs**.
- To show what the system must **do**.
- To form a basis for the design and design optimization.
- To enable **a logical approach to change management**
- To **partition** the work out to contractors.
- To act as a **foundation** for testing and payment.
- To **test** the system or any of its parts during development.
- To **communicate the basis** about the system in non-technical terms to all participants.

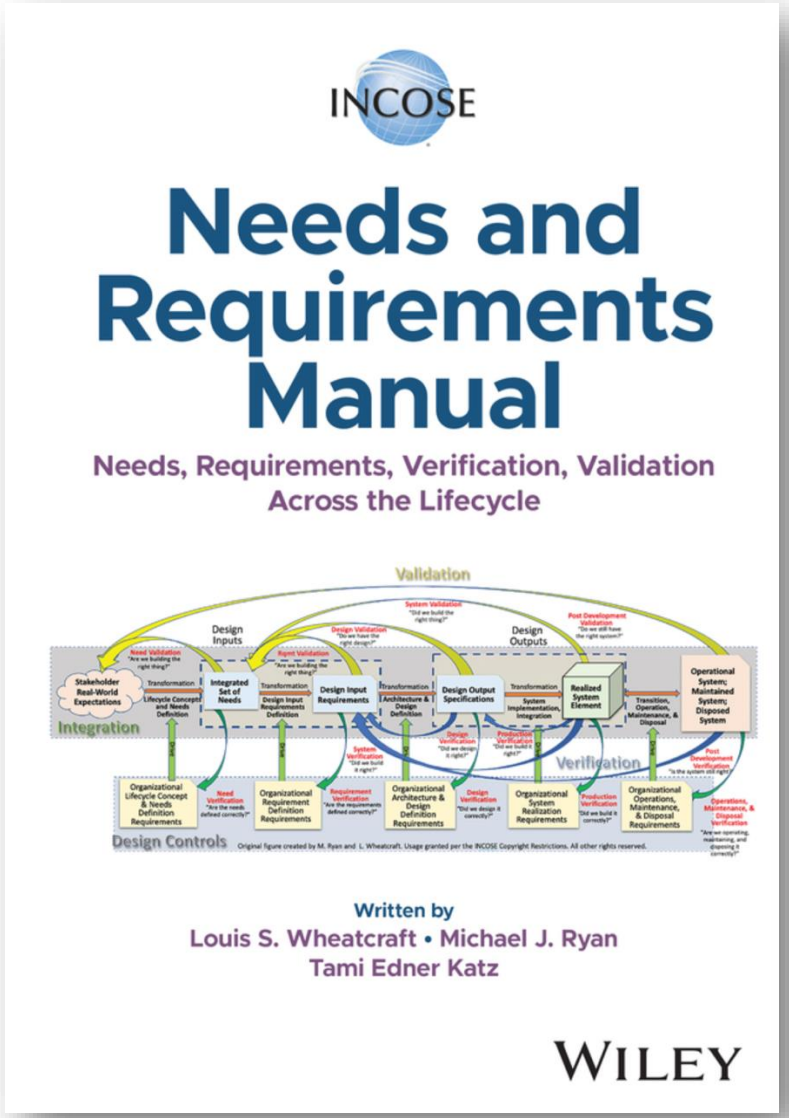
Class example so we can start: **functions to requirements**

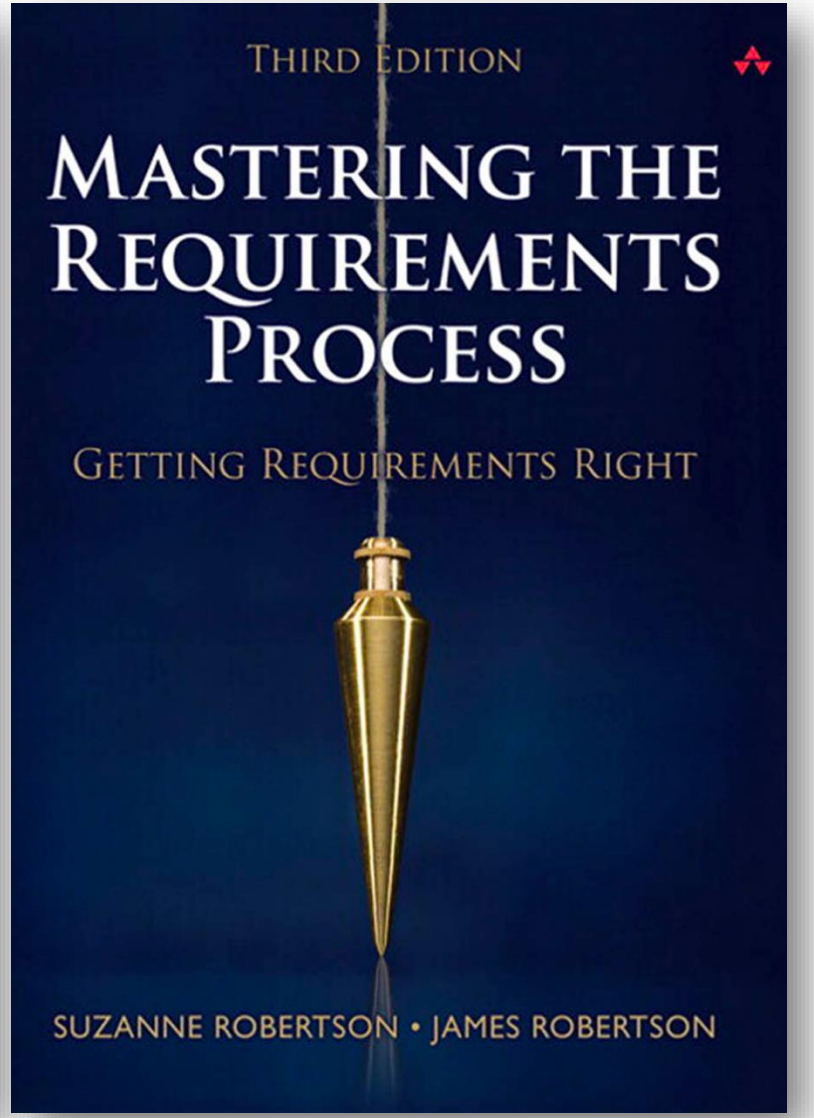
- **NEED STATEMENT: PORTABLE AUTOMOTIVE VEHICLE FOR OFF-ROAD**
  - Use Case: Camping Trip
    - Ops: Load Camping Gear On To Vehicle
      - Function: Provide Space To Store Camping Gear
        - Requirement: The vehicle shall be designed with a minimum cargo storage area of 11 cubic feet.
        - Rationale: Estimated calculation of space needed to accommodate customer's camping equipment
    - Ops: Load Personnel On To Vehicle
      - Function: Provide Capability To Carry Up To 4 People
    - Ops: Drive To Camp Site
      - Function: Provide Capability To Drive 20 Miles Roundtrip
        - Requirement: The vehicle driving range shall be a minimum of 40 miles.
        - Rationale: Customer stated the maximum range (out and back) of use as being 20 miles after getting the vehicle to where it would be used.
      - Function: Provide Capability To Utilize GPS
    - Ops: Get Stuck
      - Function: Provide Vehicle Capability To Get Un-Stuck
    - Ops: Unload Camping Gear/Camp
      - Function: Provide Electrical Interface For DC Powered Camp Eq.
        - Requirement: The vehicle shall provide, 12 VDC +/- 1.5 VDC, 12 Amps maximum, auxiliary utility ports.
        - Rationale: Based on power draw assessment of the connecting equipment
        - Requirement: The vehicle shall provide two utility ports for connecting DC powered camping equipment.
        - Rationale: Customer desired two specific ports to connect equipment simultaneously

Learning Objectives of today

- Remember the requirements definitions
- Be able to recognize functional / non-functional requirements

References for today's Learning Objectives





[02] INCOSE's Needs and Requirements Manual - https://erau.primo.exlibrisgroup.com/permalink/01ERAU_INST/15kv1ds/alma995949588205946

- Chapter 02

[10] Mastering the Requirement Process - Suzanne Robertson / James Robertson - https://erau.primo.exlibrisgroup.com/permalink/01ERAU_INST/15kv1ds/alma995930159005946

- Chapter 01, *10 and 11*

Puzzling (questionable / uncomfortable) truths

From the book: “Mastering the Requirements Process”

- Truth 01 - Requirements are not really about requirements.

- Truth 02 - If we must build a system, then it must be optimally valuable for its owner.

- Truth 03 - If your system does not have to satisfy a need, then you can build anything. However, if it is meant to satisfy a need, then you have to know what that need is to build the right system.

- Truth 04 - There is an important difference between building a system and solving a business problem. The former does not necessarily accomplish the latter.

- Truth 05 - The requirements do not have to be written, but they have to become known to the builders.

- Truth 06 - Your customer won’t always give you the right answer. Sometimes it is impossible for the customer to know what is right, and sometimes he just doesn’t know what he needs.

- Truth 07 - Requirements do not come about by chance. There needs to be some kind of orderly process for developing them.

- Truth 08 - You can be as iterative as you want, but you still need to understand what the business needs.

- Truth 09 - There is no silver bullet. All our methods and tools will not compensate for poor thought and poor workmanship.

- Truth 10 - Requirements, if they are to be implemented successfully, must be measurable and testable.

- Truth 11 - You, the requirement engineer, will change the way the user thinks about their problems, either now or later.

Truth 12 - Requirements are a human issue

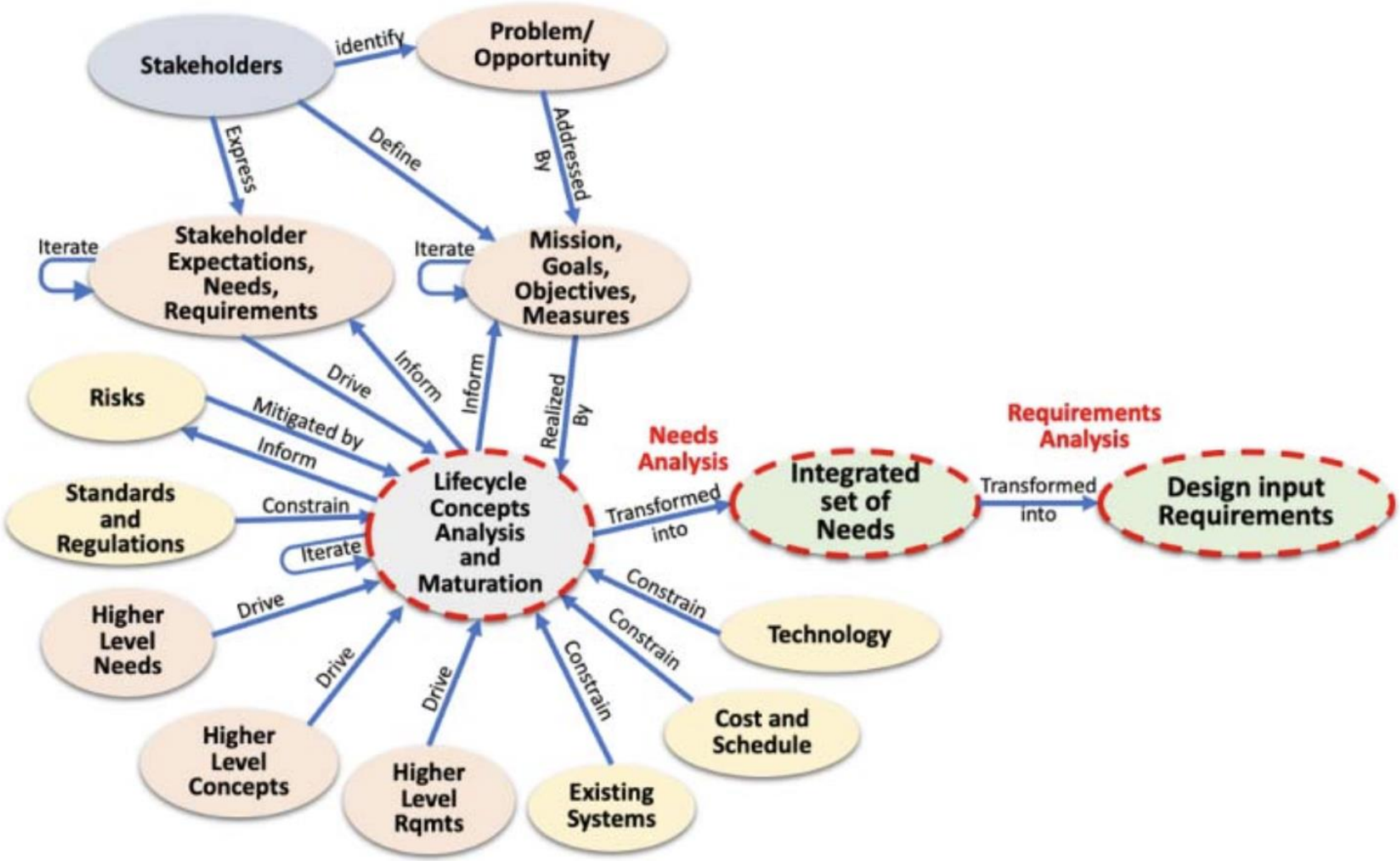
- Each requirement written is a collected/agreed statement.
- A requirements document has the force of a contract behind it, **but the needs that it expresses come from people**.
- **Systems' developers and engineers** work in close relationship with key stakeholders to find out more about the
  - **Problem to be solved**
  - **Systems capabilities and performances**
  - **Systems and Project constraints**
  - Etc.

**** we are going to do some experiments w/ AI, but in the end, YOU ARE THE RESPONSIBLE FOR THE REQUIREMENT STATEMENT / REQUIREMENT SET (DOC) BE CORRECT.**

INCOSE’s point of view

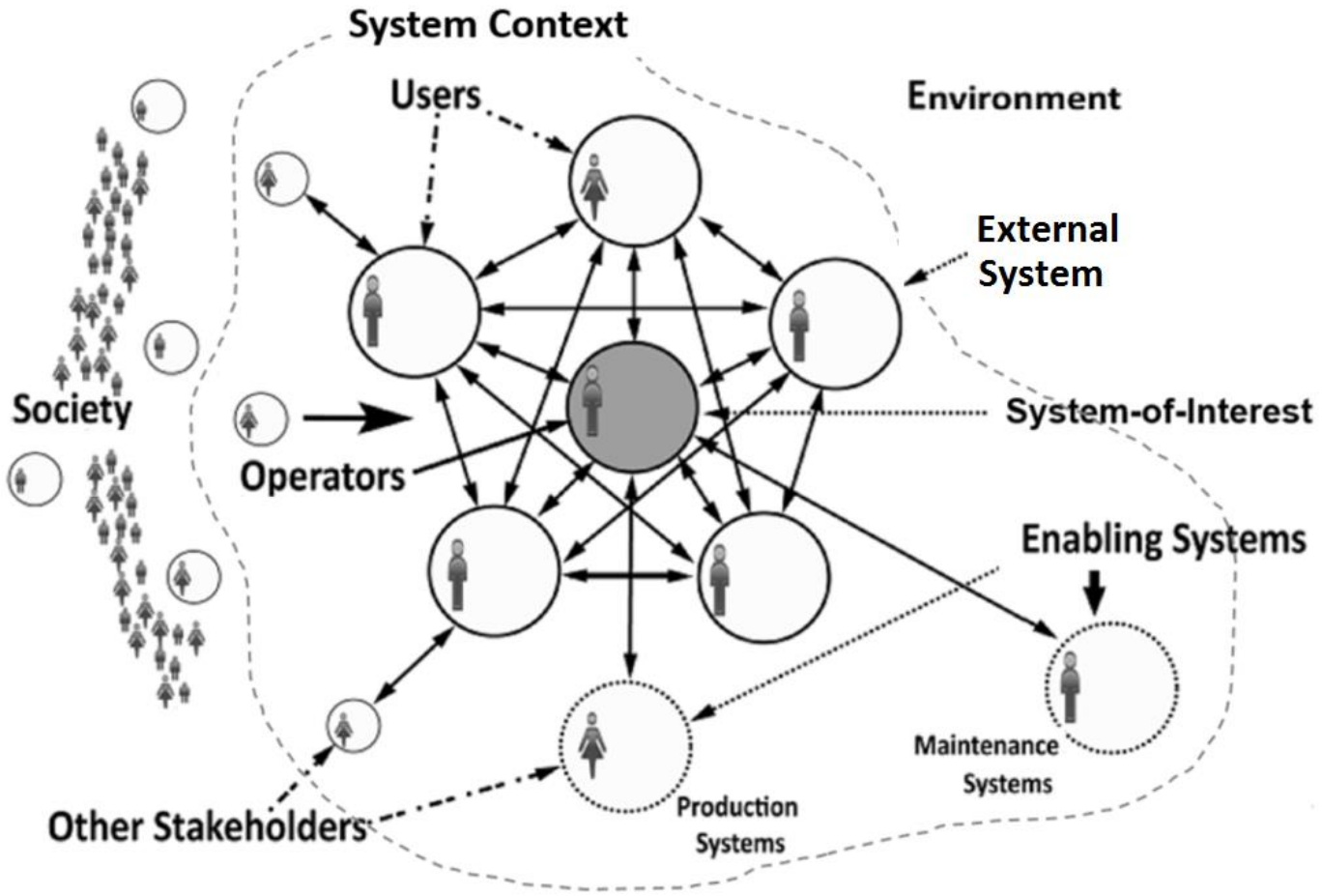
A more “structured eye”

Flow of expectations/needs into the requirements



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

Remember that we also have different types / levels of stakeholders

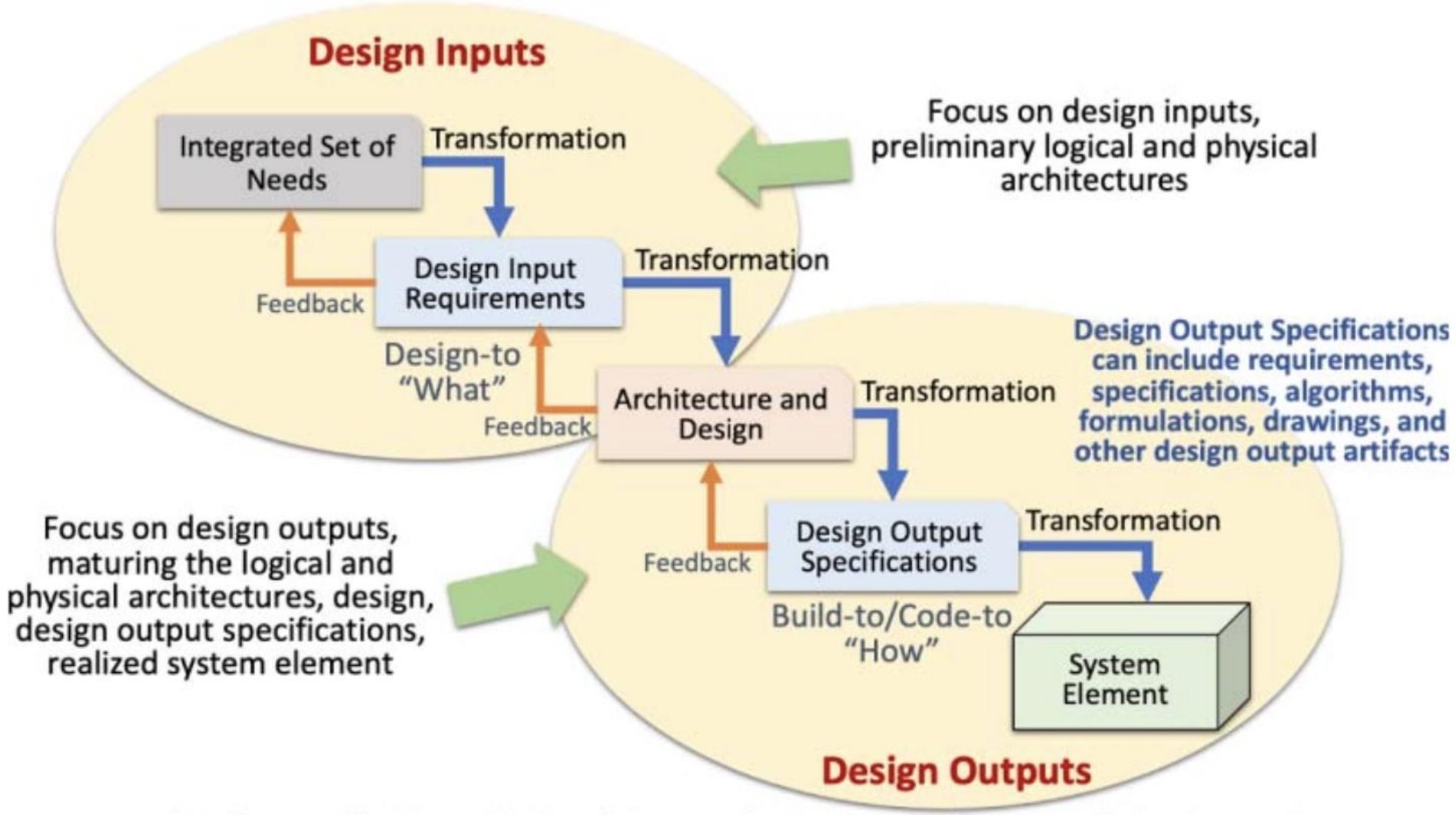


Needs level **vs** Requirements level



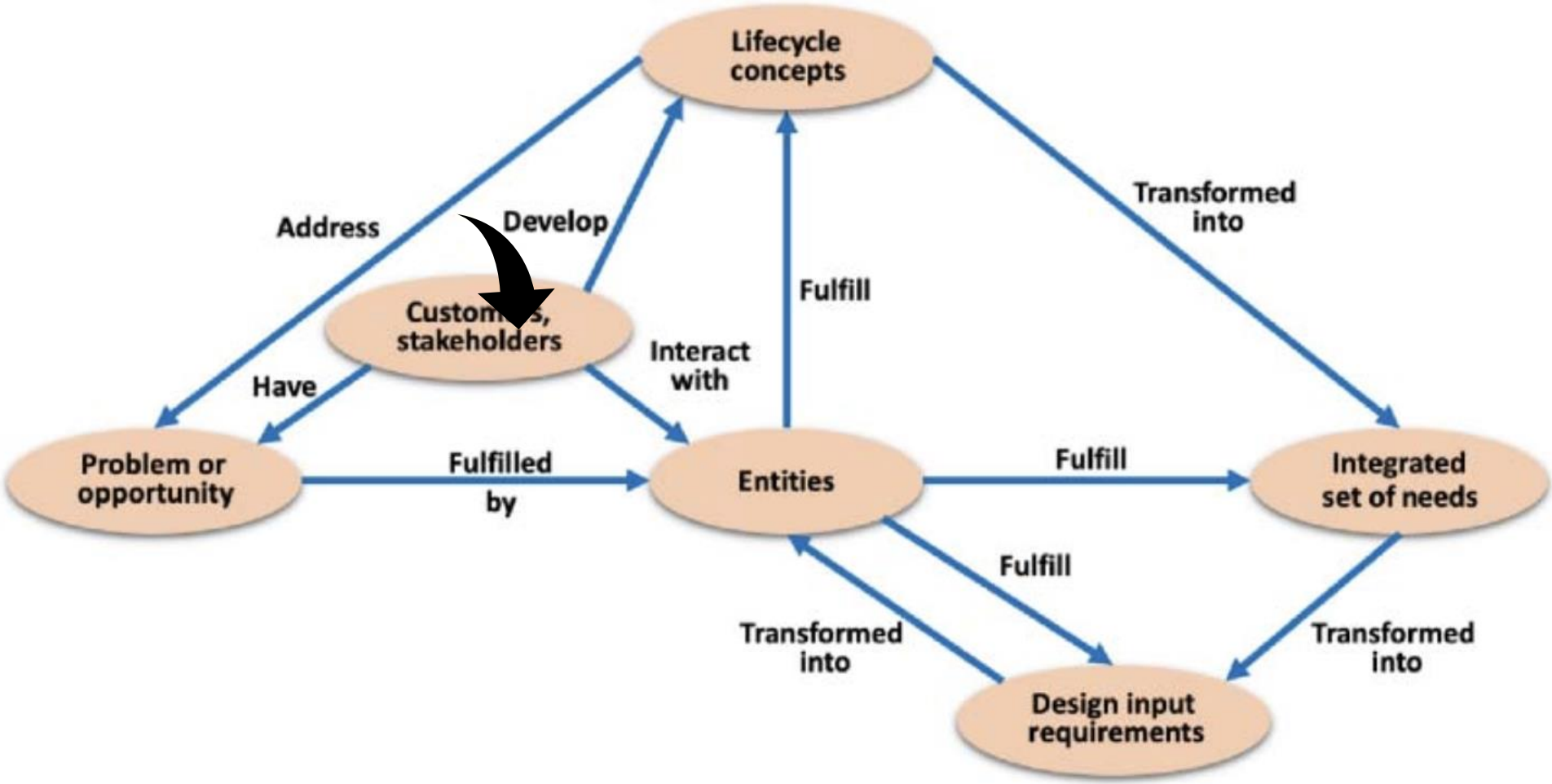
Original figure created by L. Wheatcraft. Usage granted per the INCOSE Copyright Restrictions. All other rights reserved.

"Traditional" ISO/IEC/IEEE 15288 sequence



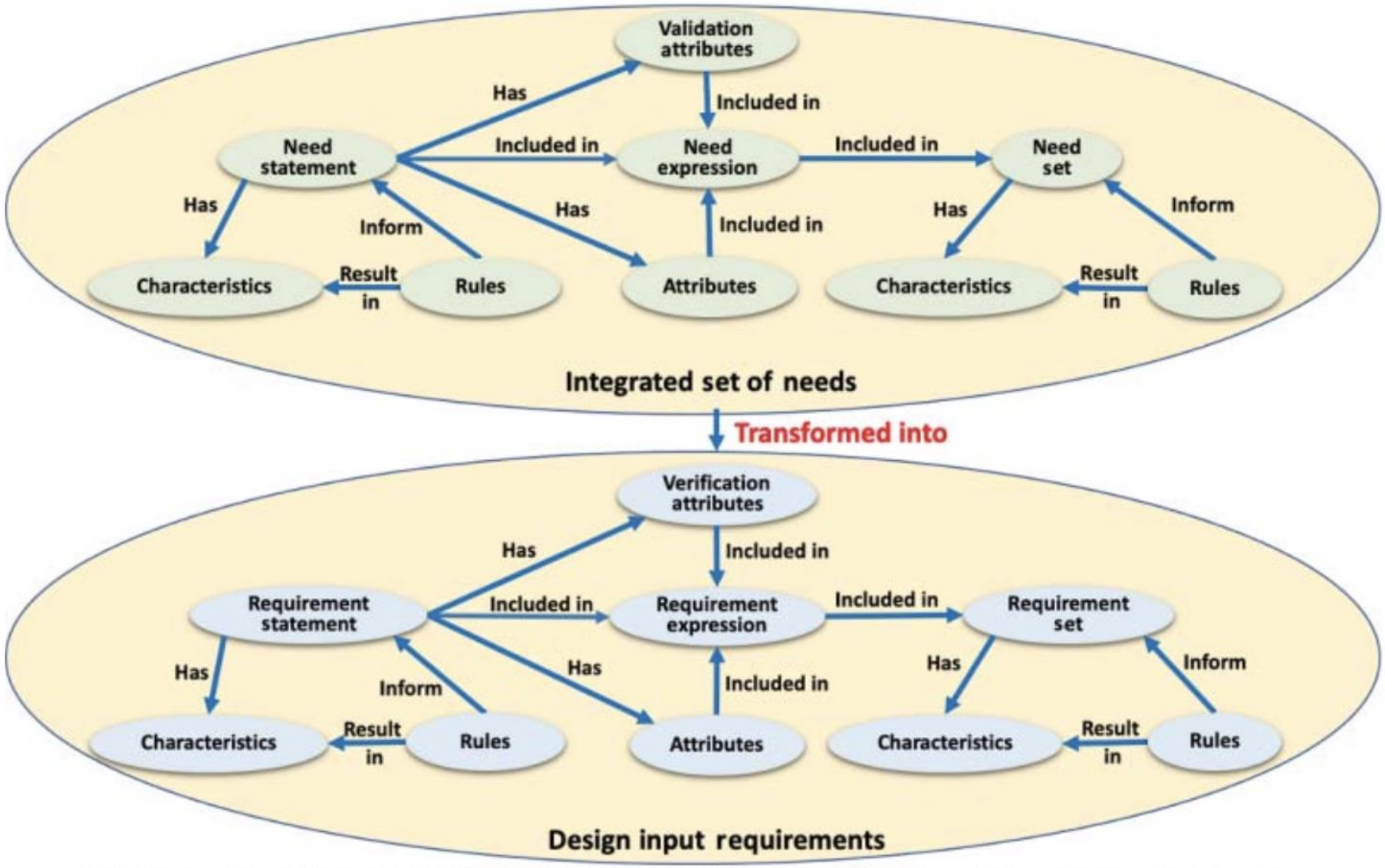
Original figure created by M. Ryan and L. Wheatcraft. Usage granted per the INCOSE Copyright Restrictions. All other rights reserved.

General ontology



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

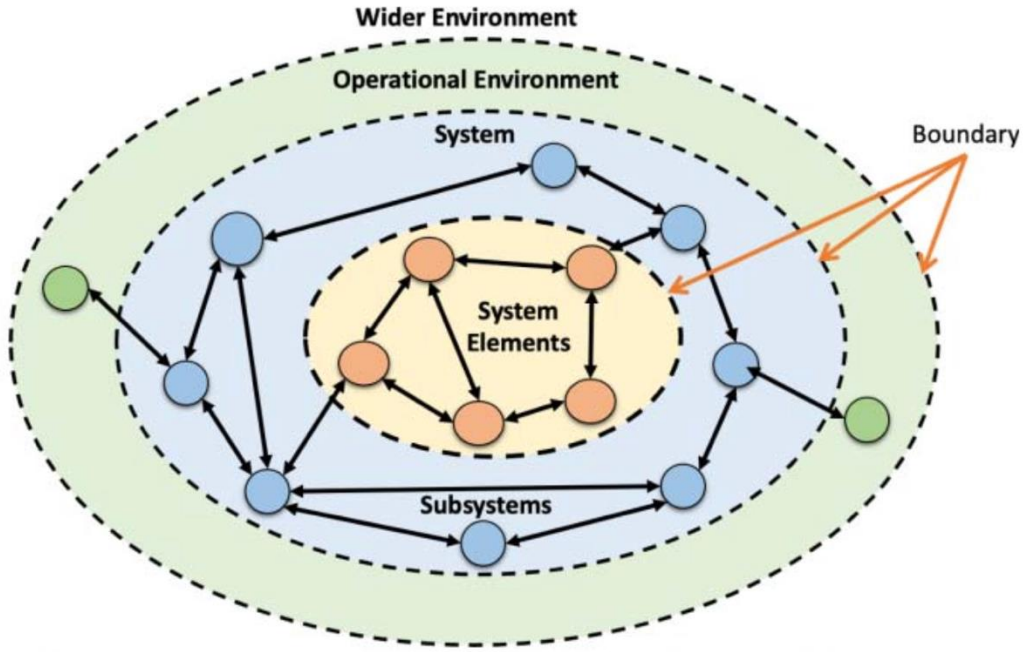
Each statement is not alone…



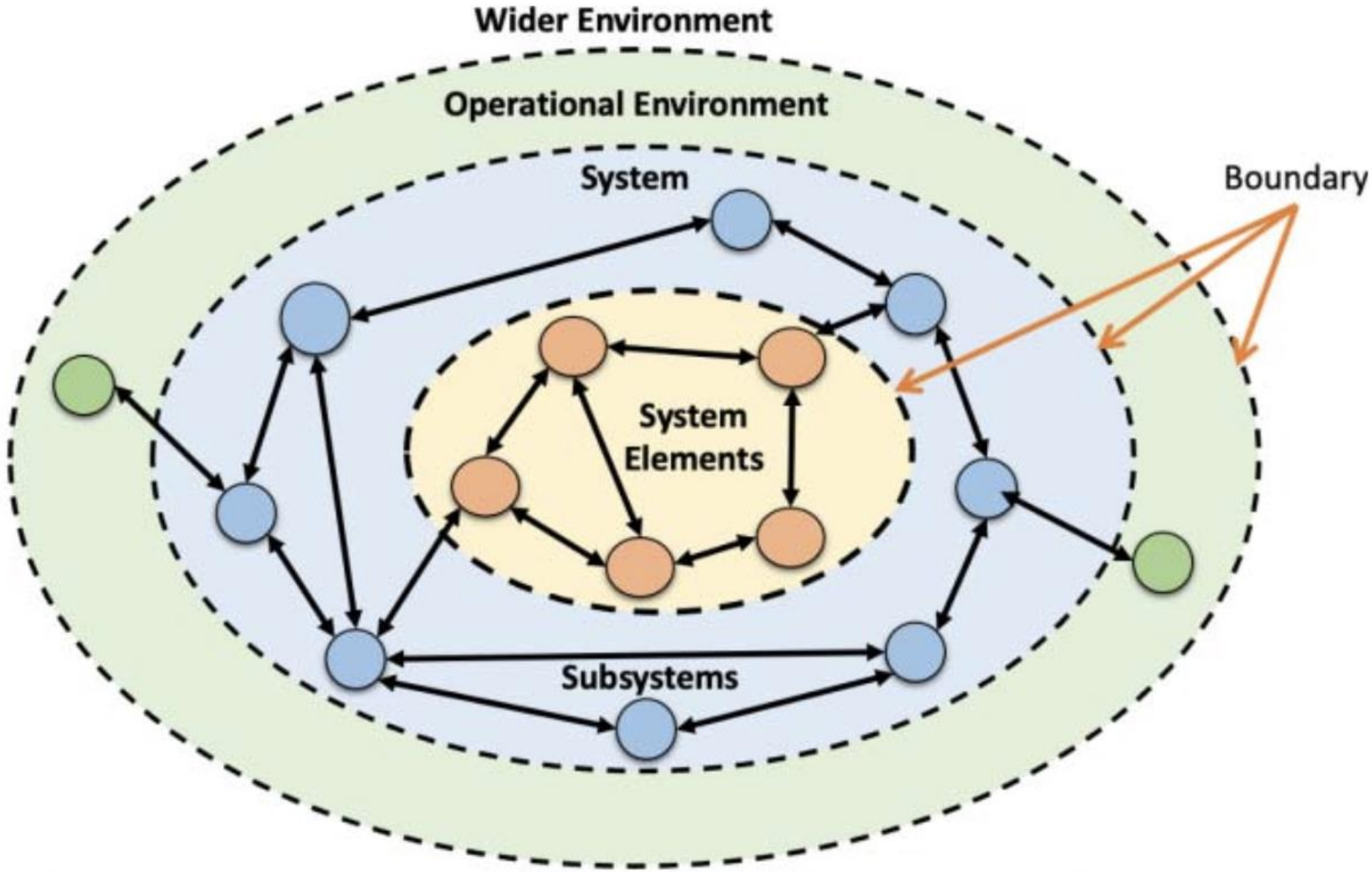
Original figure created by M. Ryan and L. Wheatcraft. Usage granted per the INCOSE Copyright Restrictions. All other rights reserved.

We end-up w/ the “whole”

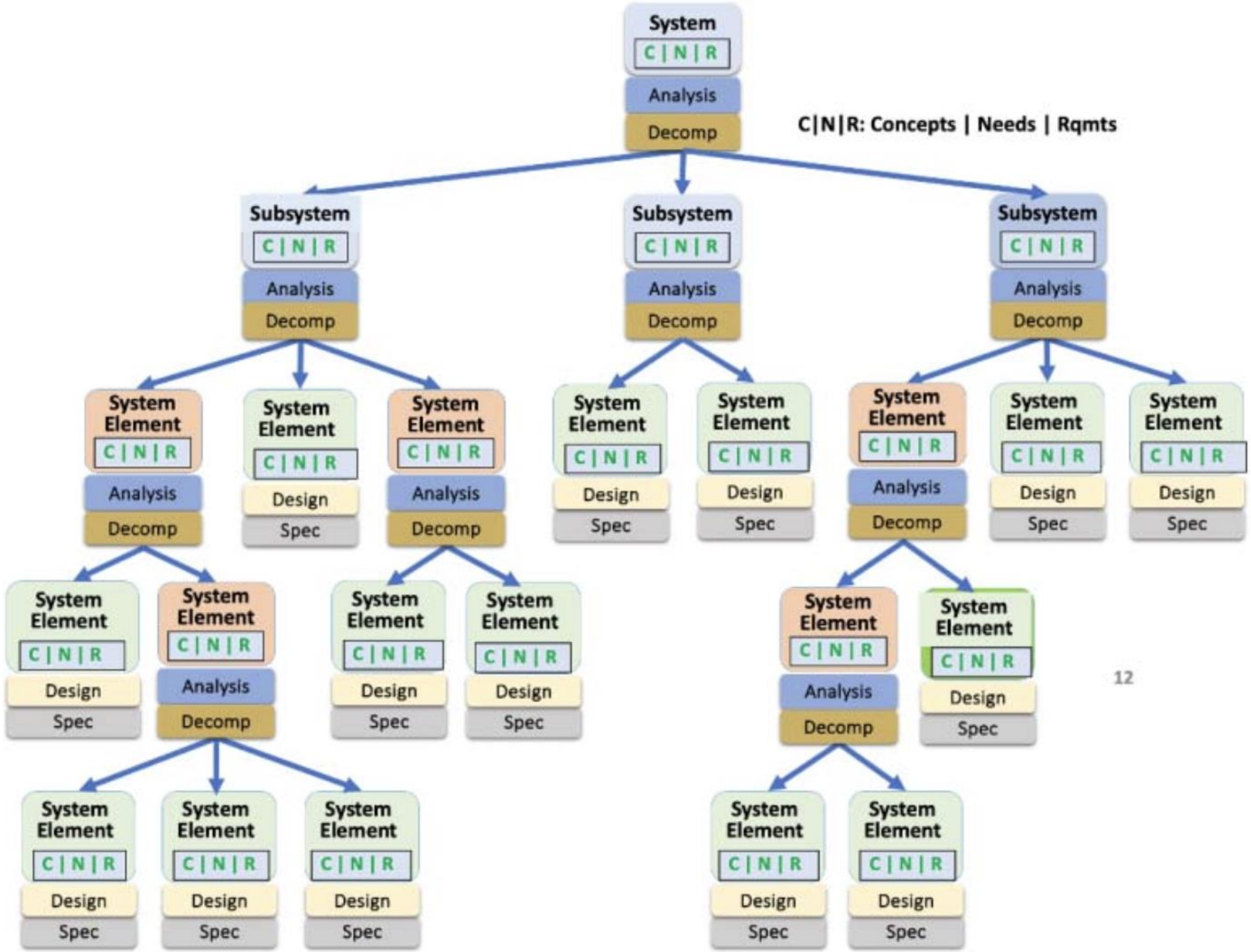
- A **need set** is a well-formed set of agreed-to-need **expressions for the entity and its external interfaces**.
- A **requirement set** is a well-formed set of agreed-to requirement **expressions for an entity and its external interfaces**.



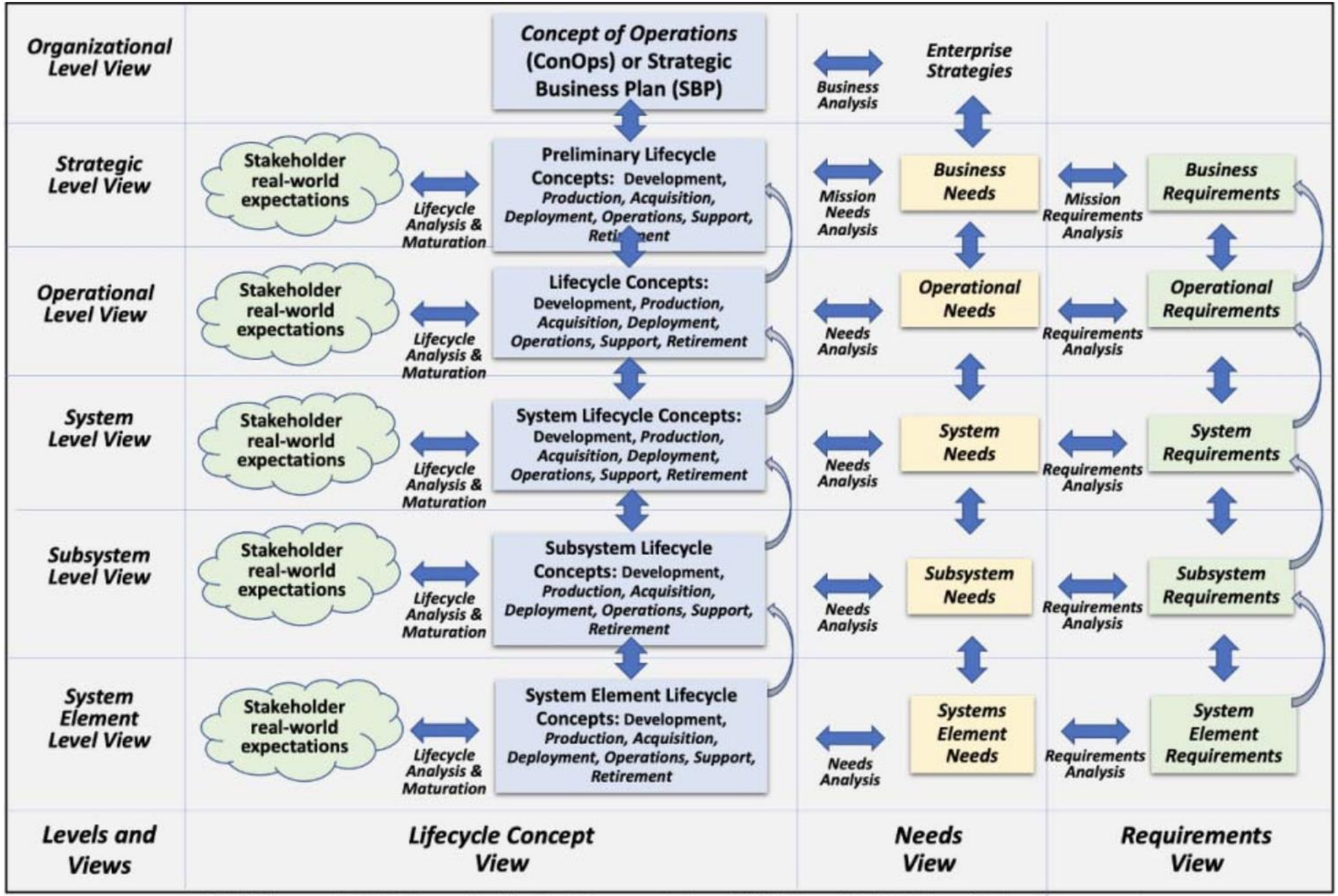
Original figure created by M. Ryan and L. Wheatcraft. Usage granted per the INCOSE Copyright Restrictions. All other rights reserved.



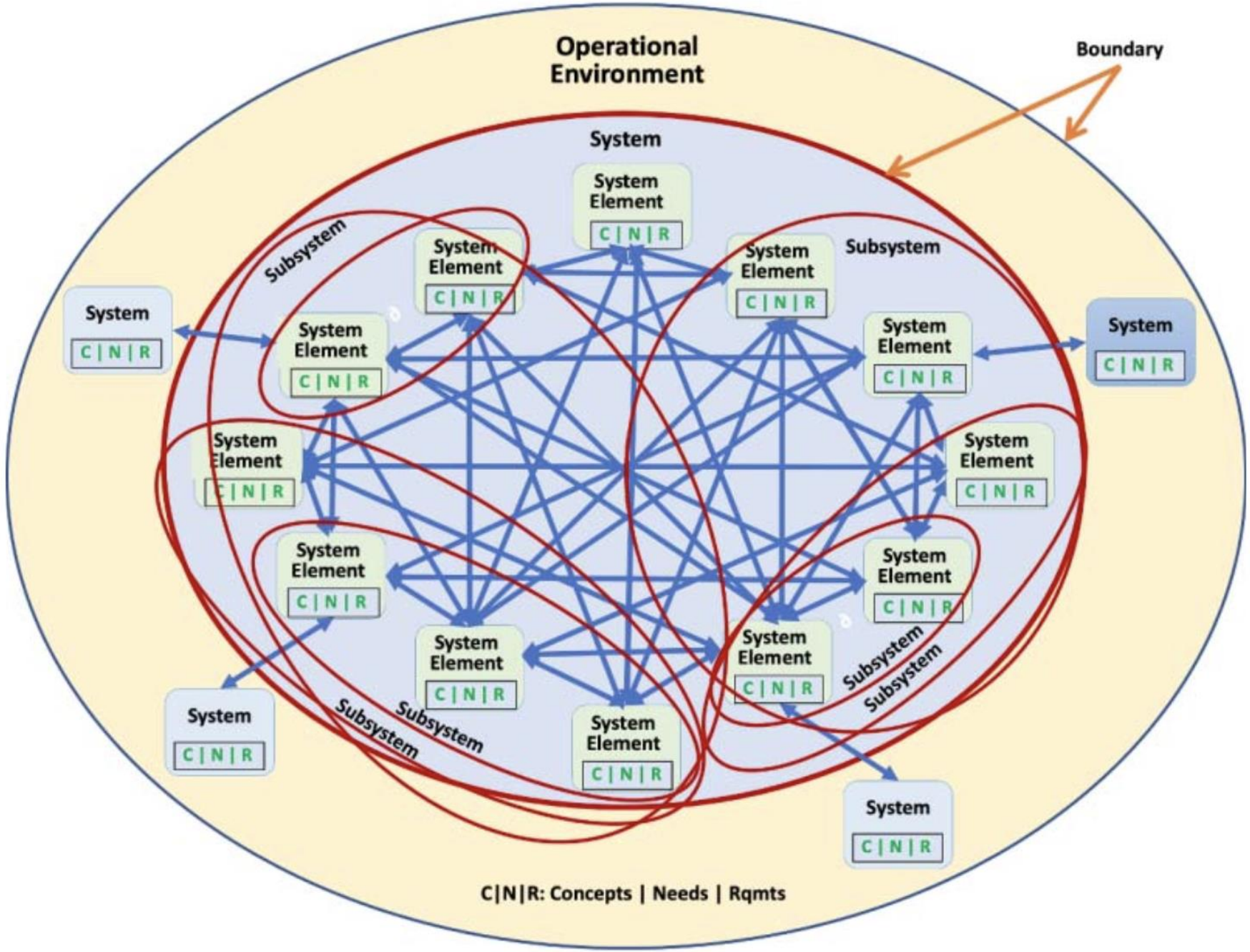
Original figure created by M. Ryan and L. Wheatcraft. Usage granted per the INCOSE Copyright Restrictions. All other rights reserved.



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



Original figure created by M. Ryan and 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.

What are requirements anyway?

requirement [ri-**kwahy*uh*** **r**-m*uh*** nt] SHOW IPA

SEE SYNONYMS FOR *requirement* ON THESAURUS.COM

noun

1. that which is required; a thing demanded or obligatory:
   One of the requirements of the job is accuracy.
2. an act or instance of requiring.
3. a need or necessity:
   to meet the requirements of daily life.

DEFINITIONS

Definitions from IEEE STD 1220-1994.

- **Requirement.** A statement identifying a capability, physical characteristic, or quality factor that bounds a product or process need for which a solution will be pursued.

- **Constraint.** A limitation or implied requirement that constrains the design solution or implementation of the systems engineering process, is not changeable by the performing activity, and is generally non allocable.

- **Specification.** A document that fully describes a physical element or its interfaces in terms of requirements (functional, performance, constraints and physical characteristics) and the qualification conditions and procedures for each requirement.

TO DO OR NOT TO DO

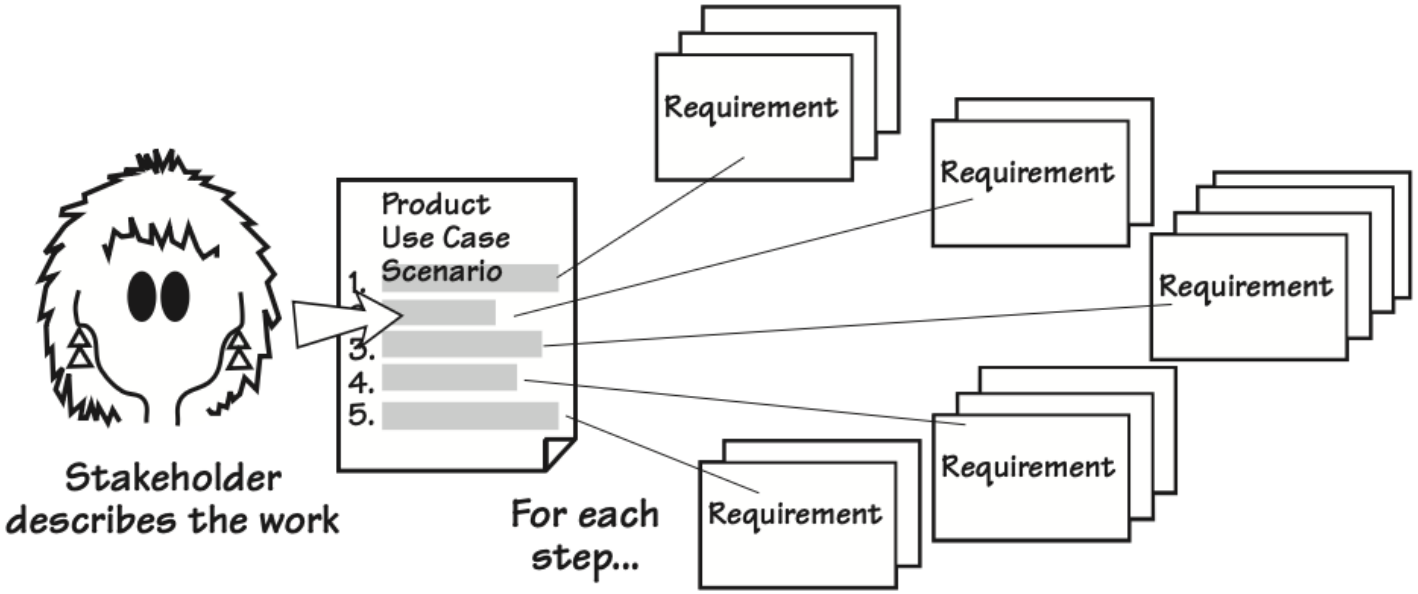
- **Functional Requirements** describe what the system should do and **Non-functional Requirements** place constraints on how these functional requirements are implemented.



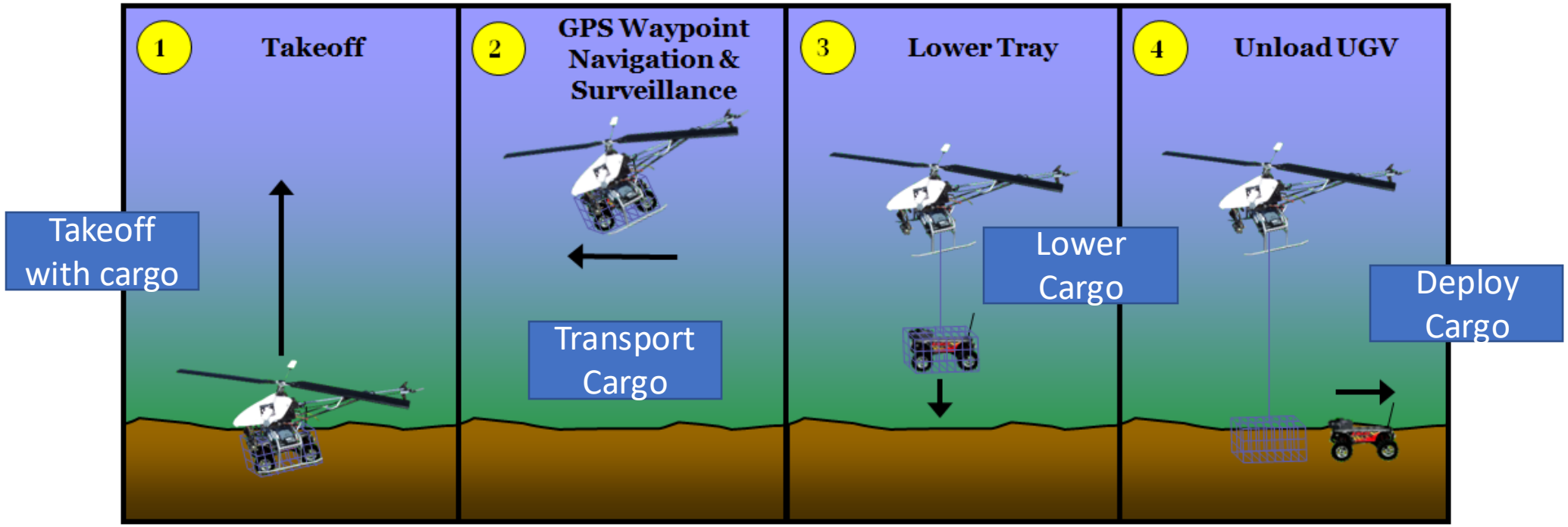
Functional Requirements (FR)

- **Functional requirements specify what the system must do - the actions it must perform to satisfy the fundamental reasons for its existence.**

- *The <system> shall predict which road sections will freeze within the selected time parameters.*



From CONOPs (1/2)



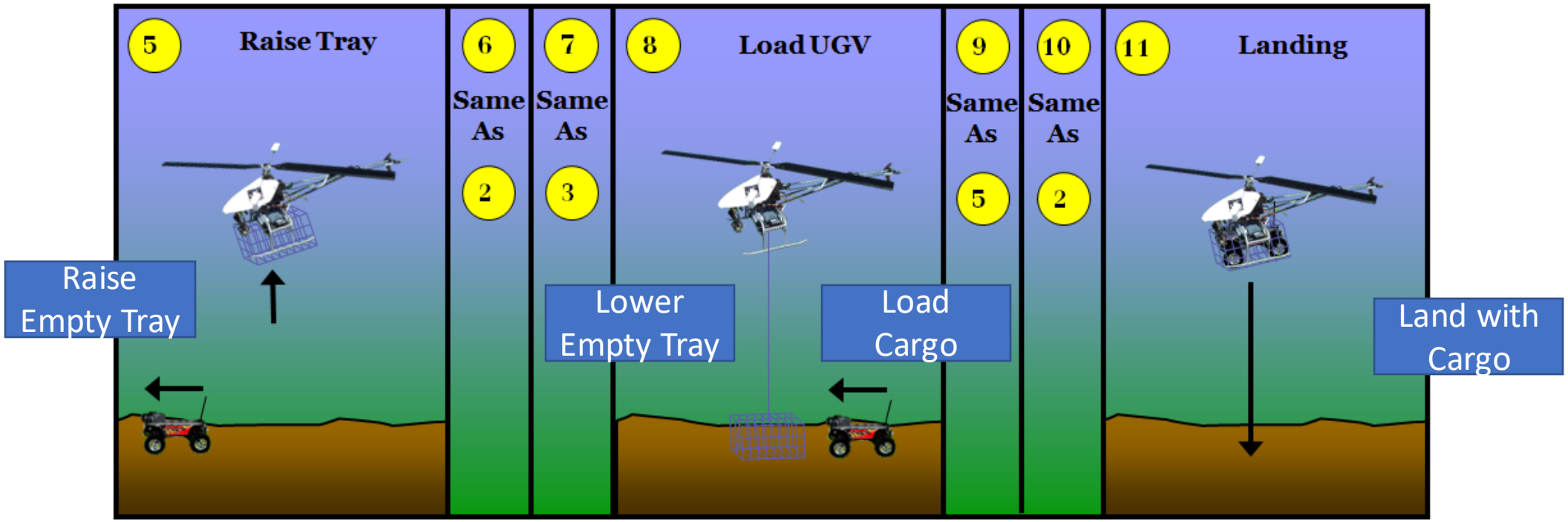
Autonomous takeoff is integral to Rotomotion's control package, but requires proper calibration of sensor gains.

Autonomous hovering and GPS waypoint navigation is integral to Rotomotion's control package, but requires a collision avoidance system for true safety of flight.

The cargo must be lowered at sufficient speed to avoid possible buildup of oscillations.

Stable hovering is especially important while unloading the UGV.

From CONOPs (2/2)



The cargo must be raised at sufficient speed to avoid possible buildup of oscillations, but less weight during raising means less risk.

(6) GPS Waypoint Navigation & Surveillance
(7) Lower Tray

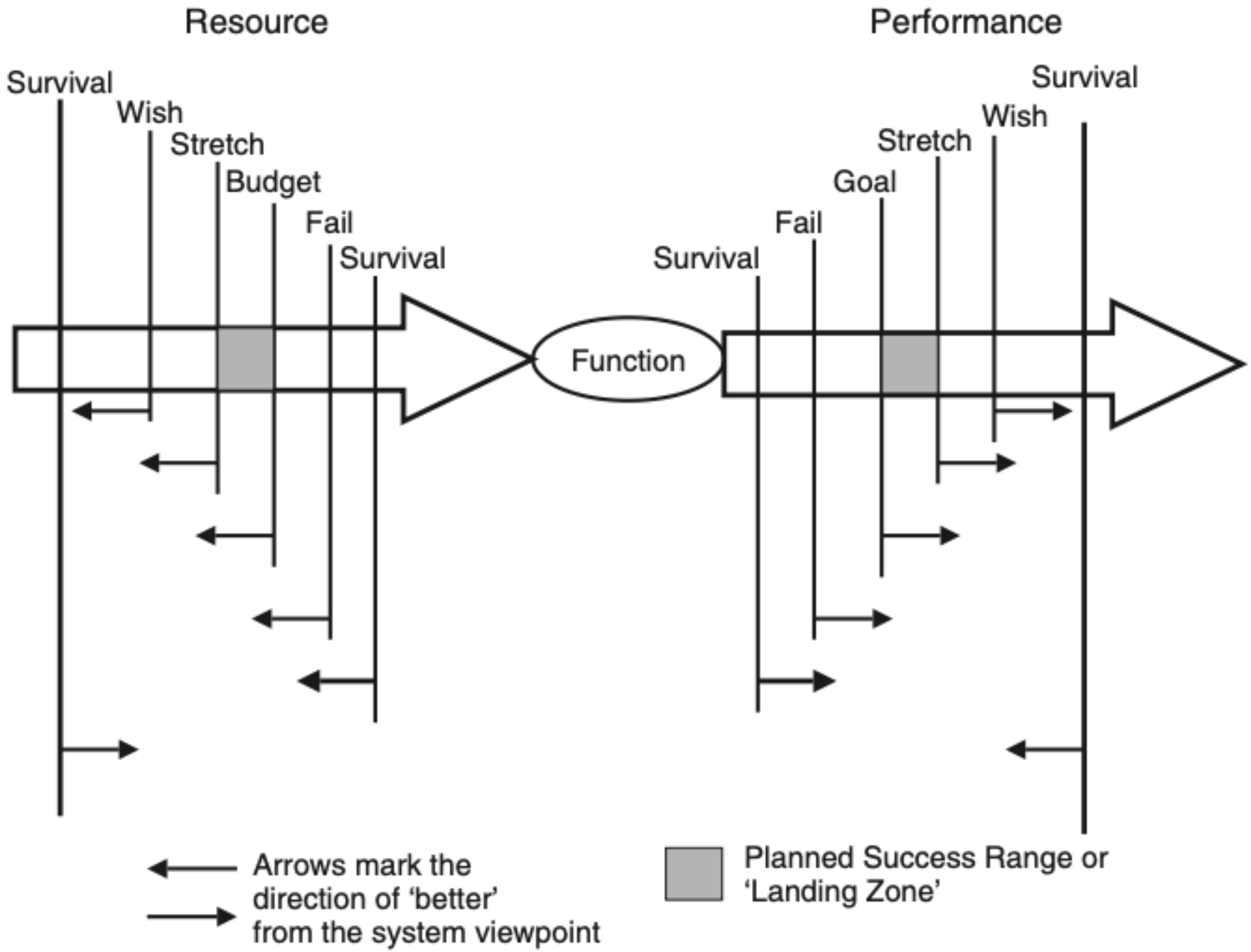
Stable hovering is especially important while loading the UGV. Servoing of UGV into the tray requires accurate sensors and failsafe tray design.

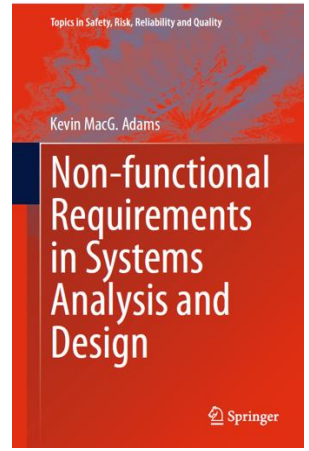
(9) Raise Tray
(10) GPS Waypoint Navigation & Surveillance

Autonomous landing is integral to Rotomotion's control package, but requires identification of proper landing zone through sensor systems for truly reliable landings.

Non-Functional Requirements (NFR)

- The non-functional requirements describe the qualities that your system must have or, to put that another way, how well it does the things it does.
- Such requirements make the product attractive, or usable, or fast, or reliable, or safe.
- You use non-functional requirements to specify response times, or accuracy limits on calculations.
- You write non-functional requirements when you need your product to have a particular appearance, or be used by people in particular circumstances, or adhere to laws applicable to your kind of business.


Resource
Survival
Wish
Stretch
Budget
Fail
Survival
Function
Performance
Survival
Wish
Stretch
Goal
Fail
Survival
Arrows mark the direction of 'better' from the system viewpoint
Planned Success Range or 'Landing Zone'



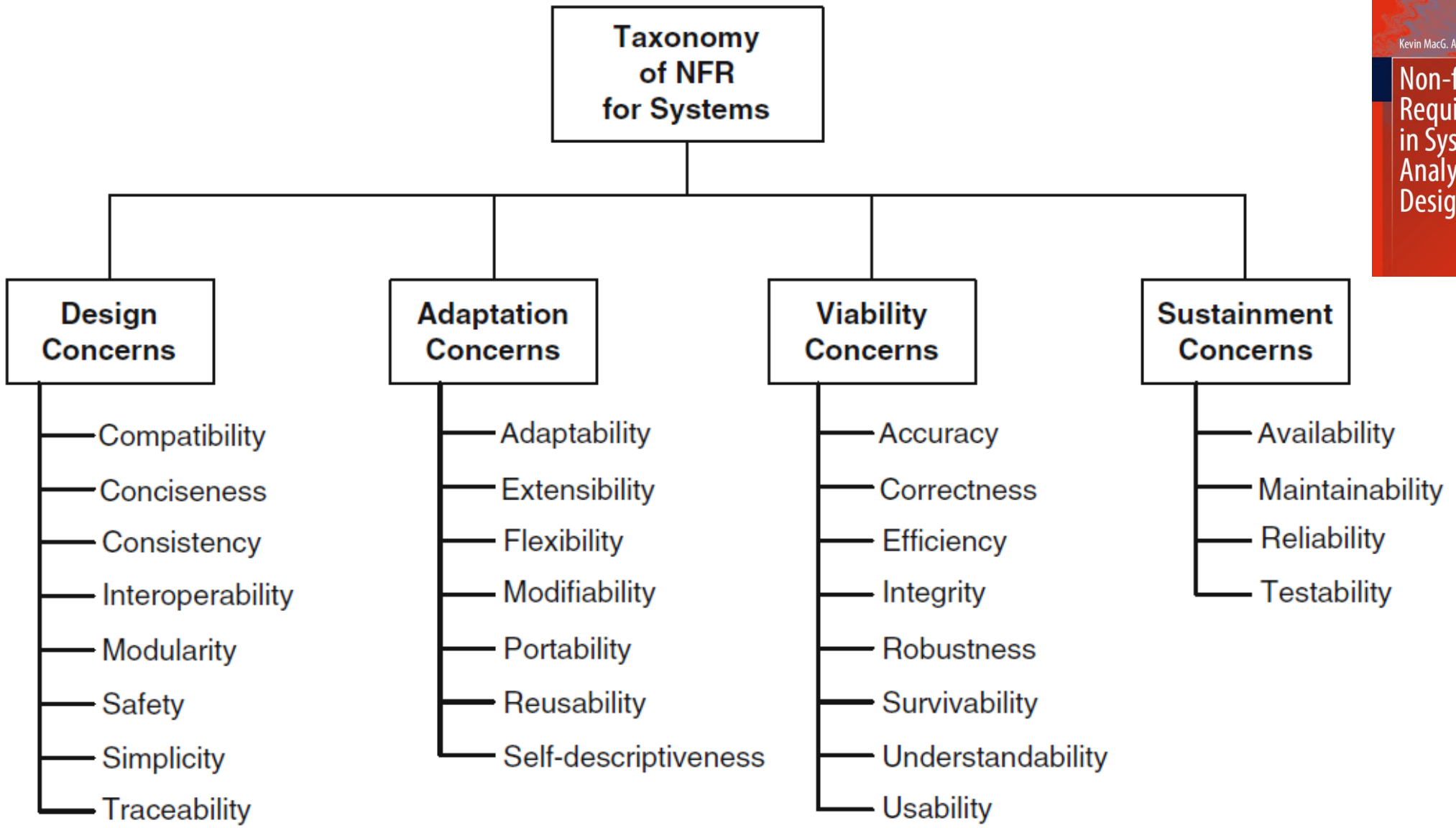


Fig. 3.5 Taxonomy of non-functional requirements for systems

Some Nonfunctional requirement types

Accuracy (5)	"1. A qualitative assessment of correctness, or freedom from error. 2. A quantitative measure of the magnitude of error" IEEE and ISO/IEC (2010, p. 6)
Adaptability (3)	"Degree to which a product or system can effectively and efficiently be adapted for different or evolving hardware, software or other operational or usage environments" [SE VOCAB]
Availability (2)*	"1. The degree to which a system or component is operational and accessible when required for use. 2. Ability of a component or service to perform its required function at a stated instant or over a stated period of time" IEEE and ISO/IEC (2010, p. 29)
Compatibility (3)	"1. The ability of two or more systems or components to perform their required functions while sharing the same hardware or software environment. 2. The ability of two or more systems or components to exchange information" IEEE and ISO/IEC (2010, p. 62)

Conciseness (3)	"Software attributes that provide implementation of a function with a minimum amount of code" IEEE and ISO/IEC (2010, p. 69)
Consistency (5)	"1. The degree of uniformity, standardization, and freedom from contradiction among the documents or parts of a system or component. 2. Software attributes that provide uniform design and implementation techniques and notations" IEEE and ISO/IEC (2010, p. 73)
Correctness (5)	"The degree to which a system or component is free from faults in its specification, design, and implementation" IEEE and ISO/IEC (2010, p. 81)
Efficiency (6)	"1. The degree to which a system or component performs its designated functions with minimum consumption of resources. 2. Producing a result with a minimum of extraneous or redundant effort" IEEE and ISO/IEC (2010, p. 120)
Extensibility (1)*	"The ease with which a system or component can be modified to increase its storage or functional capacity. Syn: expandability, extensibility" IEEE and ISO/IEC (2010, p. 136)
Flexibility (3)	"The ease with which a system or component can be modified for use in applications or environments other than those for which it was specifically designed. Syn: adaptability cf. extendability, maintainability" IEEE and ISO/IEC (2010, p. 144)

Integrity (5)	"The degree to which a system or component prevents unauthorized access to, or modification of, computer programs or data" IEEE and ISO/IEC (2010, p. 181)
Interoperability (6)	"The ability of two or more systems or components to exchange information and to use the information that has been exchanged" IEEE and ISO/IEC (2010, p. 186)
Maintainability (7)	"The ease with which a hardware system or component can be retained in, or restored to, a state in which it can perform its required functions" IEEE and ISO/IEC (2010, p. 204)
Modifiability (3)	"The ease with which a system can be changed without introducing defects cf. maintainability" IEEE and ISO/IEC (2010, p. 222)
Modularity (4)	"1. The degree to which a system or computer program is composed of discrete components such that a change to one component has minimal impact on other components. 2. Software attributes that provide a structure of highly independent components" IEEE and ISO/IEC (2010, p. 223)
Operability (4)	"The state of being able to perform the intended function" IEEE and ISO/IEC (2010, p. 240)

Portability (6)	"The ease with which a system or component can be transferred from one hardware or software environment to another" IEEE and ISO/IEC (2010, p. 261)
Reliability (8)	"The ability of a system or component to perform its required functions under stated conditions for a specified period of time" IEEE and ISO/IEC (2010, p. 297)
Reusability (4)	"The degree to which an asset can be used in more than one software system, or in building other assets" IEEE and ISO/IEC (2010, p. 307)
Robustness (2)*	"The degree to which a system or component can function correctly in the presence of invalid inputs or stressful environmental conditions cf. error tolerance, fault tolerance" IEEE and ISO/IEC (2010, p. 313)
Safety (1)*	"The expectation that a system does not, under defined conditions, lead to a state in which human life, health, property, or the environment is endangered" IEEE and ISO/IEC (2010, p. 315)
Self-descriptiveness (3)	"1. The degree to which a system or component contains enough information to explain its objectives and properties. 2. Software attributes that explain a function's implementation. cf. maintainability, testability, usability" IEEE and ISO/IEC (2010, p. 322)
Simplicity (3)	"1. The degree to which a system or component has a design and implementation that is straightforward and easy to understand. 2. Software attributes that provide implementation of functions in the most understandable manner cf. complexity" IEEE and ISO/IEC (2010, p. 327)

Survivability (1)*	"1. The degree to which a product or system continues to fulfill its mission by providing essential services in a timely manner in spite of the presence of attacks. cf. recoverability" [SE VOCAB]
Testability (4)	"1. The extent to which an objective and feasible test can be designed to determine whether a requirement is met. 2. The degree to which a requirement is stated in terms that permit establishment of test criteria and performance of tests to determine whether those criteria have been met" IEEE and ISO/IEC (2010, p. 371)
Traceability (3)	"1. The degree to which a relationship can be established between two or more products of the development process, especially products having a predecessor-successor or master-subordinate relationship to one another. 2. The identification and documentation of derivation paths (upward) and allocation or flowdown paths (downward) of work products in the work product hierarchy" IEEE and ISO/IEC (2010, p. 378)
Understandability (2)*	"The ease with which a system can be comprehended at both the system-organizational and detailed-statement levels. NOTE Understandability has to do with the system's coherence at a more general level than readability does" IEEE and ISO/IEC (2010, p. 385)
Usability (7)	"The ease with which a user can learn to operate, prepare inputs for, and interpret outputs of a system or component" IEEE and ISO/IEC (2010, p. 388)

Other taxonomies

Table 4.1 Requirements Taxonomy in Various Industries

REQUIREMENT GRANULARITY	INDUSTRIES: INFORMATION TECHNOLOGY OR SOFTWARE DEVELOPMENT	INDUSTRIES: ENGINEERING, ARCHITECTURE, OR PRODUCT DEVELOPMENT	INDUSTRIES: MULTIDISCIPLINARY PROJECTS
Very High Level	• Business Requirements • Problems • Objectives	• Objectives • Problems • Business Requirements	• Business Requirements • Problems • Objectives
High Level	• Features • High-Level Requirements	• Features • Technical Requirements	• Features • Components

Detailed Level	• Functional Requirements
	• Nonfunctional Requirements
		• Functions
		  ○ Performance
		  ○ Value
		  ○ Size
		  ○ Safety
		  ○ Special
		• Attributes
		• Behaviors
			• Requirements

Other	• Use Cases • User Stories • Constraints • Business Rules • Data Dictionaries	• Constraints • Business Rules	• Constraints • Business Rules

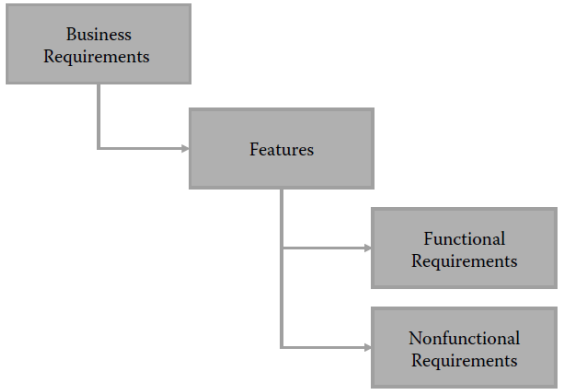


Figure 4.1 Software requirements taxonomy.

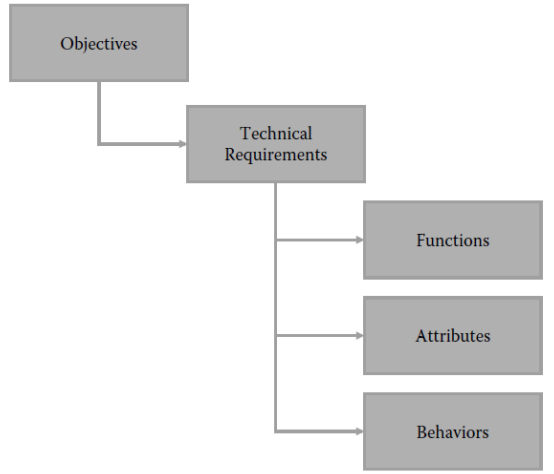


Figure 4.2 Engineering requirements taxonomy.

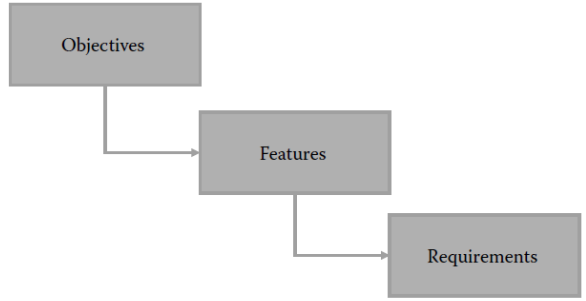
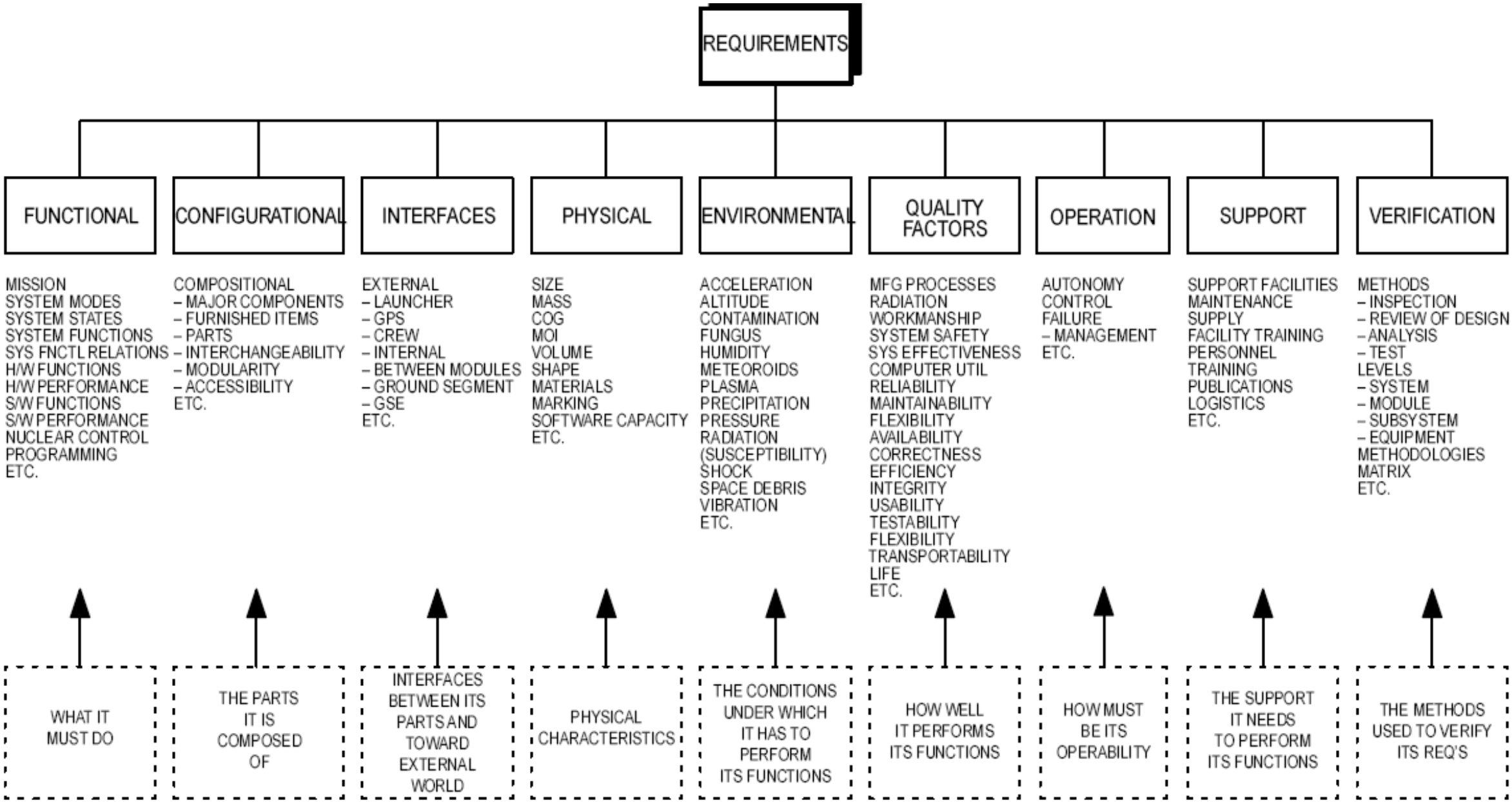
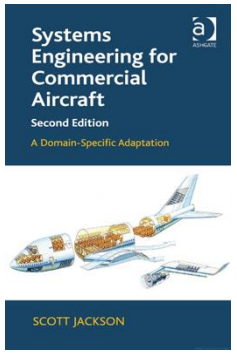


Figure 4.4 Multidisciplinary projects requirements taxonomy.


REQUIREMENTS
FUNCTIONAL
CONFIGURATIONAL
INTERFACES
PHYSICAL
ENVIRONMENTAL
QUALITY FACTORS
OPERATION
SUPPORT
VERIFICATION
MISSION
SYSTEM MODES
SYSTEM STATES
SYSTEM FUNCTIONS
SYS FNCTL RELATIONS
H/W FUNCTIONS
H/W PERFORMANCE
S/W FUNCTIONS
S/W PERFORMANCE
NUCLEAR CONTROL
PROGRAMMING
ETC.
COMPOSITIONAL
– MAJOR COMPONENTS
– FURNISHED ITEMS
– PARTS
– INTERCHANGEABILITY
– MODULARITY
– ACCESSIBILITY
ETC.
EXTERNAL
– LAUNCHER
– GPS
– CREW
– INTERNAL
– BETWEEN MODULES
– GROUND SEGMENT
– GSE
ETC.
SIZE
MASS
COG
MOI
VOLUME
SHAPE
MATERIALS
MARKING
SOFTWARE CAPACITY
ETC.
ACCELERATION
ALTITUDE
CONTAMINATION
FUNGUS
HUMIDITY
METEOROIDS
PLASMA
PRECIPITATION
PRESSURE
RADIATION
(SUSCEPTIBILITY)
SHOCK
SPACE DEBRIS
VIBRATION
ETC.
MFG PROCESSES
RADIATION
WORKMANSHIP
SYSTEM SAFETY
SYS EFFECTIVENESS
COMPUTER UTIL
RELIABILITY
MAINTAINABILITY
FLEXIBILITY
AVAILABILITY
CORRECTNESS
EFFICIENCY
INTEGRITY
USABILITY
TESTABILITY
FLEXIBILITY
TRANSPORTABILITY
LIFE
ETC.
AUTONOMY
CONTROL
FAILURE
– MANAGEMENT
ETC.
SUPPORT FACILITIES
MAINTENANCE
SUPPLY
FACILITY TRAINING
PERSONNEL
TRAINING
PUBLICATIONS
LOGISTICS
ETC.
METHODS
– INSPECTION
– REVIEW OF DESIGN
– ANALYSIS
– TEST
LEVELS
– SYSTEM
– MODULE
– SUBSYSTEM
– EQUIPMENT
METHODOLOGIES
MATRIX
ETC.
WHAT IT MUST DO
THE PARTS IT IS COMPOSED OF
INTERFACES BETWEEN ITS PARTS AND TOWARD EXTERNAL WORLD
PHYSICAL CHARACTERISTICS
THE CONDITIONS UNDER WHICH IT HAS TO PERFORM ITS FUNCTIONS
HOW WELL IT PERFORMS ITS FUNCTIONS
HOW MUST BE ITS OPERABILITY
THE SUPPORT IT NEEDS TO PERFORM ITS FUNCTIONS
THE METHODS USED TO VERIFY ITS REQ'S

- Functional requirements are simply the function without any quantitative value for the function.
  - *The system **shall** illuminate in the passenger seat trays.*
  - *The system shall provide illumination in the passenger seat trays.*
- Performance requirements is a measure of the extent to which a system performs a function.
  - *The minimum illumination level at passenger seat trays during night operations **shall** be of 300 lux.*
  - *The system shall provide a minimum illumination level of 300 lux at passenger seat trays during night operations*



Characteristics of Needs / Requirements

INCOSE

C1 - Necessity

- The need or requirement statement defines an essential capability, characteristic, constraint, or quality factor needed to satisfy a concept, need or parent requirement. If it is not included in the set of needs and requirements, a deficiency in capability or characteristic will exist which cannot be fulfilled by implementing other needs or requirements in the set.
- Each requirement, individually or in combination with other requirements in the set, must be sufficient to satisfy a specific stakeholder need or parent requirement, including applicable conditions.

C2 - Appropriate

- The specific intent and amount of detail of the need or requirement statement is appropriate to the level (the level of abstraction) of the entity to which it refers.

- Needs are not requirements. They are stated from the perspective of the stakeholder's expectations for the SOI. Requirements are stated from the perspective of what the system needs to do to implement the needs.

- Requirements may be imposed at any level; however, as a rule, a requirement should be expressed at the level of the entity to which it refers.

C3 - Ubiquitous

- Need statements must be written such that the stakeholder intent is clear. Requirement statements must be stated such that the requirement can be interpreted in only one way by all the intended readers.

- An ambiguous need is not correct nor able to be validated. An ambiguous requirement is not correct nor verifiable.

- Due to the limitations of language, it may prove difficult to completely remove all ambiguity. In this case the use of the rationale attribute, may provide additional insight of the intent to remove the ambiguity. The inclusion of contextual information with the requirement statement may help to reduce ambiguity.

C4 - Complete

- The requirement statement sufficiently describes the necessary capability, characteristic, constraint, or quality factor to meet the need without needing other information to understand the requirement.

- An agreement is not useful unless the obligation is complete and does not need further explanation. A well-formed requirement needs no further amplification to implement its intent.

- To be complete, functional/performance requirements must have observable functions (“what”), measurable performance (“how well”) and a statement of conditions (“under what conditions”, for example, triggering events, environments, states and modes).

C5 - Singular

- The stakeholder need or requirement statement should state a single capability, characteristic, constraint, or quality factor.

- Keep the need or requirement statement limited to one quality, characteristic, capability, function, or constraint. Understand how the statements fit into the allocation and traceability philosophy for the project.

- Avoid the use of the word “and” when it ties together multiple thoughts (phrases in the sentence), each of which may be allocated and verified differently. The presence of the conjunction “and” in a need or requirement statement should always prompt the writer to consider whether or not the statement is singular.

C6 - Feasible

- The need or requirement can be realized within entity constraints (for example: cost, schedule, technical, legal, ethical, safety) with acceptable risk.
- There is little point in agreeing to an obligation for a need or requirement that is not feasible. Agreeing to a need or requirement that cannot be realized with acceptable risk within constraints often results in project cost overruns and schedule slips. Inherently unachievable needs and requirements, such as 100% reliability, are at best a waste of time, and at worst lead to needlessly expensive solutions.
- A good tool for assessing risk is the use of Technology Readiness Levels (TRLs) to determine and compute the maturity of a critical technology—a lower TRL represents more risk to the project than a higher TRL.

C7 - Verifiable

- The requirement is structured and worded such that its realization can be proven (verified) to the customer's satisfaction at the level the requirement exists.

- A requirement is considered to be verifiable if:
  - 1) a verification case can be determined from the requirement statement that allows for complete examination of all aspects of the requirement and precise determination of all values, including tolerances, such that success or failure can be determined, and
  - 2) the requirement content is adequate to completely define the expected behavior, characteristics, conditions, and success criteria for the actual verification activity.

C8 - Correct

- The need must be an accurate representation of the concept from which it was transformed. A requirement must be an accurate representation of the need from which it was transformed.
- Needs must be traceable to a source: concept, stakeholder expectation, drivers and constraints, goals, objectives, risks, etc. defined as part of the scope definition activities during the pre-concept and concept life cycle stages. The resulting needs transformed from these sources must be evaluated to determine the accuracy of the transformation.

C9 - Conforming

- The individual needs and requirements should conform to an approved standard pattern and style guide or standard for writing and managing needs and requirements.
- Conforming to standards or templates for needs and requirements will help to identify missing information resulting in the characteristics of complete, unambiguous, and verifiable.
- Organizations must have well-defined standards and processes for needs and requirements development and management. In many instances, there are applicable government, industry, and product standards, specifications, and interfaces with which compliance is required.

Final Considerations

Strongly recommend to

- Read w/ care the INCOSE Needs/Requirements Guide – Chapter 02 (2.3.1 and 2.3.2)
- Fast reading of the Mastering the Requirement Process – Chapter 01 / 10 / 11
- Read extra papers added on the Canvas