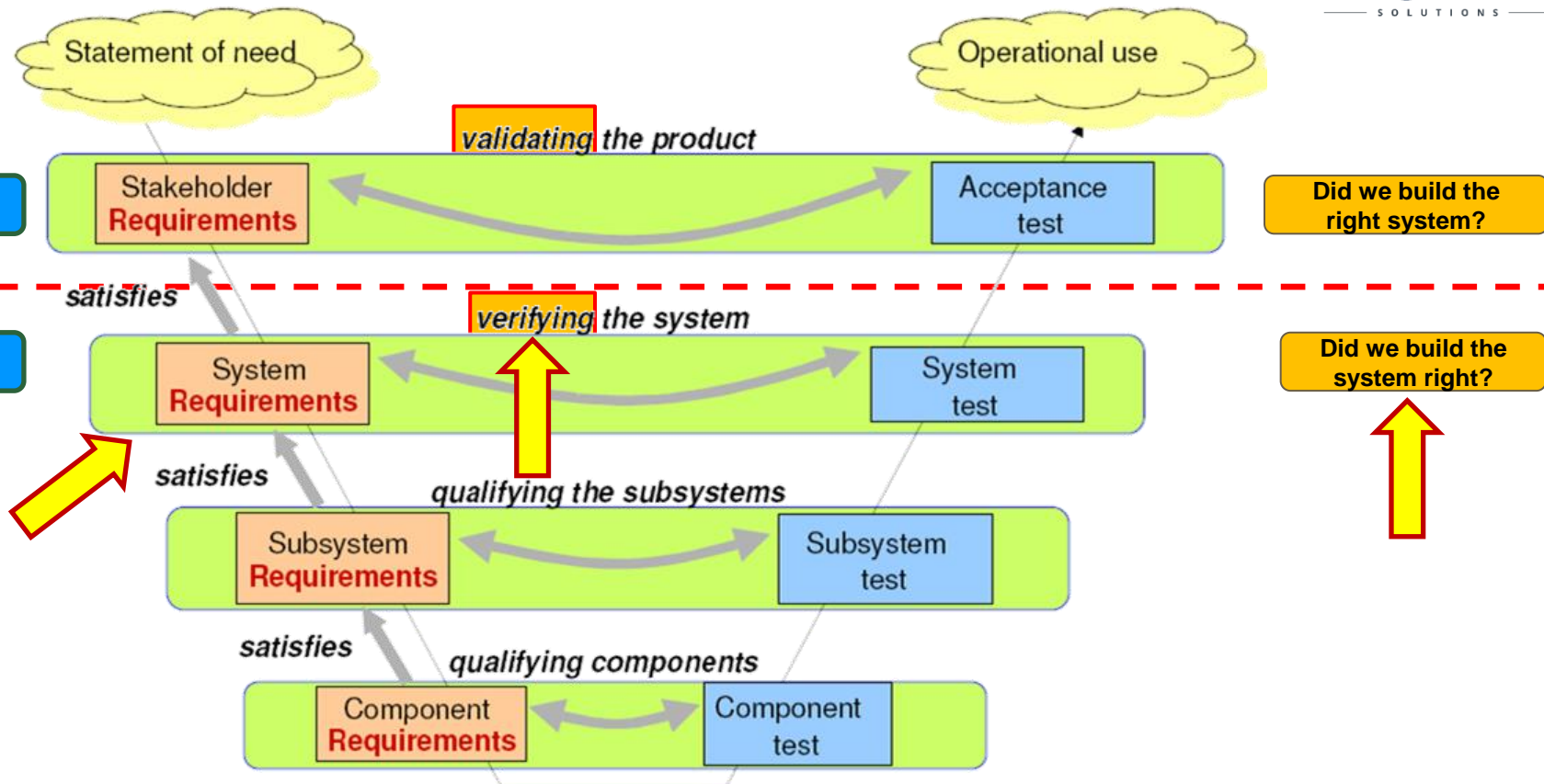# System Requirements Definition

Outline

- **Inputs-Activities-Outputs of Process**

- **Characteristics of Good Requirements**

- **Rules for Writing Requirements Statements**

- **Stakeholder Requirements to System Requirements**

- **System Requirements to Sub System Requirements**

  - SW Requirements

  - HW Requirements

System Requirements Definition Process

- The purpose of the System Requirements Definition process is to **transform** the stakeholder, **user-oriented view of desired capabilities** into a **technical view of a solution** that meets the operational needs of the user.

(*INCOSE Systems Engineering Handbook, v4*)
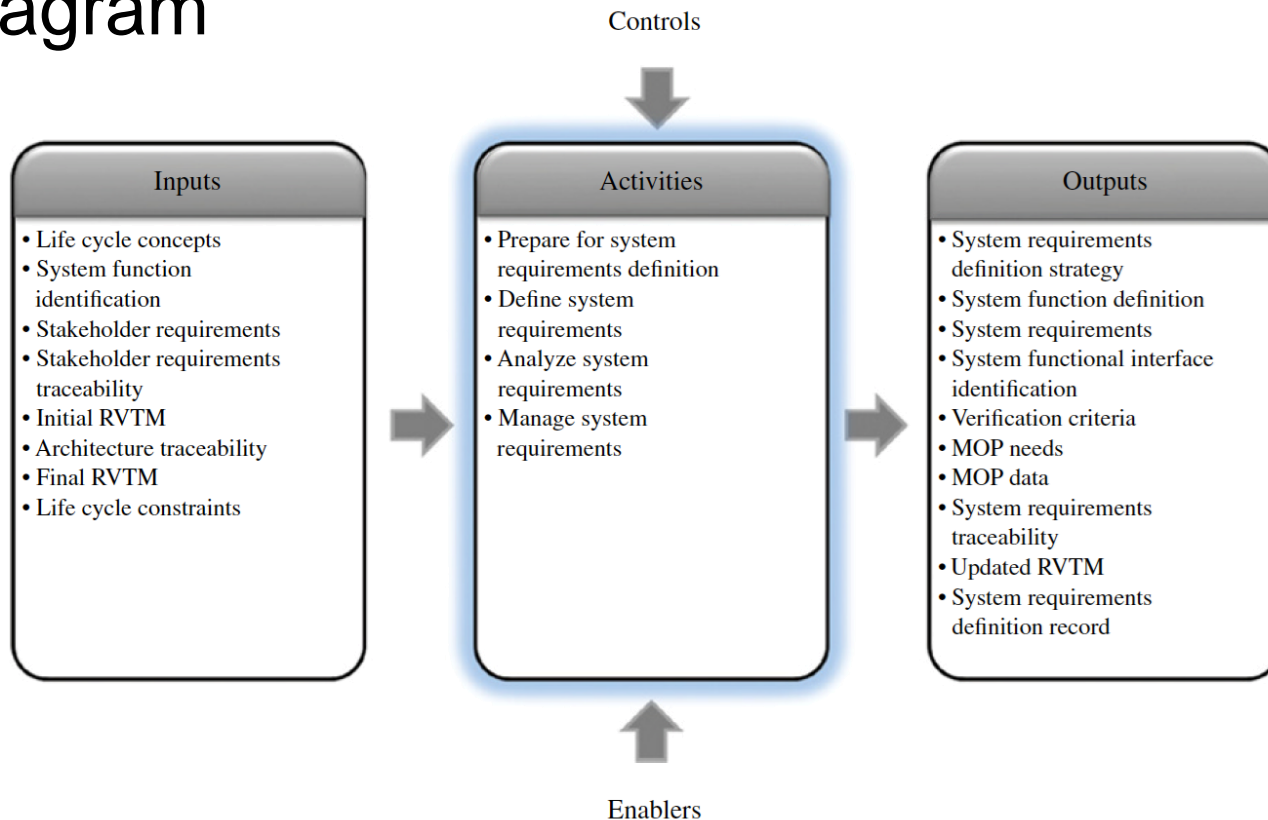
System Requirements

System requirements are all of the <u>requirements</u> at the system level <u>that describe the functions which the system as a whole should fulfill to satisfy the stakeholder needs and requirements</u>, and are expressed in an appropriate combination of textual statements, views, and <u>non-functional requirements</u>; the latter expressing the levels of safety, security, reliability, etc., that will be necessary.

System requirements play major roles in systems engineering:

- Basis of system architecture and design activities
- Basis of system integration and verification activities
- Means of communication between the various technical staff that interact throughout the project
- Every requirement carries a cost!

Inputs-Activities-Outputs of Process

IPO Diagram

Controls

Inputs	Activities	Outputs
• Life cycle concepts	• Prepare for system	• System requirements
• System function	requirements definition	definition strategy
identification	• Define system	• System function definition
• Stakeholder requirements	requirements	• System requirements
• Stakeholder requirements	• Analyze system	• System functional interface
traceability	requirements	identification
• Initial RVTM	• Manage system	• Verification criteria
• Architecture traceability	requirements	• MOP needs
• Final RVTM		• MOP data
• Life cycle constraints		• System requirements
		traceability
		• Updated RVTM
		• System requirements
		definition record

Enablers

Major Activities

- Define System Requirements

- Analyze Integrity of System Requirements

- Define Verification Criteria

Major Artifacts

- **System Requirements Document**

    - System External Interface Requirements Document

- **System Requirements Justification Documents**

- **System Requirements Traceability**

Characteristics of Good Requirements

Characteristics of Good Requirements (Individual)

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

Source: INCOSE Guide to Writing Requirements

C1 – Necessary

- If the requirement is not included, a deficiency in the capability or characteristic will exist, which cannot be fulfilled by implementing other requirements

- A requirement is NOT necessary if:
  - The requirement can be removed and the remaining set will still result in the needs being satisfied
  - The requirement cannot be traced back to a source, need, or parent requirement
  - The author cannot communicate a valid reason (rationale) for the requirement

C1 – Necessary Example

- Unacceptable:

  SR98: When ice is detected, car shall inform the user

  SR101: When ice is detected, car shall show an ice icon in less than 0.5 seconds from its detection

- Improved (Removed duplicate vague requirement)

  SR101: When ice is detected, car shall show an ice icon in less than 0.5 seconds from its detection

C2 – Appropriate

- The specific intent and amount of detail of the requirement is appropriate to the level of the entity to which it refers
  - Level: of abstraction, organization, or system architecture (system, subsystem, or system elements)

- A requirement stated at the wrong level for an entity is either not correct or may not be verifiable at that level

C2 – Appropriate Example

- Unacceptable:
  Example 1:

  SR110: The user shall be categorized either as trusted or Not_Trusted

  Example 2:

  SR201: Thermistor input ADC shall have resolution of 10 bit

- Improved
  SR110: The Security_System shall categorize each user as either Trusted or Not_Trusted

  SR201: Temperature_Measurement_System shall detect temperature with 0.1 degree C resolution

C3 – Unambiguous

- Requirement statements must be stated such that the requirement can be interpreted in only one way by all the intended stakeholders.

- Ambiguity leads to interpretations not intended by the author, leading to problems such as schedule slips, budget overruns, failure of the SOI to pass system validation, and not be accepted for its intended use.

C3 – Unambiguous Example

- **Unacceptable:**

  SR110: When the clutch is disengaged and/or the break is applied, the Engine_Management_System shall disengage the Speed_Control_System within xxx milliseconds

- **Improved** (Cleared ambiguity for consistent interpretation as per intent)

  SR110: When the [clutch is disengaged] AND the [break is applied] , the Engine_Management_System shall disengage the Speed_Control_System within xxx milliseconds

   OR

  SR110: When the clutch is disengaged, the Engine_Management_System shall disengage the Speed_Control_System within xxx milliseconds

  SR111: When the break is applied , the Engine_Management_System shall disengage the Speed_Control_System within xxx milliseconds

  OR

  SR110:When EITHER [the Clutch is disengaged] OR[the Brake is applied] the Engine_Management_System shall disengage the Speed_Control_System within xxx milliseconds

C4 – Complete

- The requirement statement sufficiently describes the necessary capability, characteristic, constraint, or quality factor to meet the need, source, or parent requirement from which it was transformed without needing other information to understand the requirement.

- Each requirement should be understood in its own right without having to understand a number of other requirements.

- Baselined requirement statements should not contain To Be Defined (TBD), To Be Specified (TBS), or To Be Resolved (TBR) clauses.

C4 – Complete Example

- **Unacceptable:**

  SR110: The Thermal_Control_System shall monitor system temperature

  Or

  SR110:The Thermal_Control_System shall update the display of System temperature

- **Improved**

  SR110: The Thermal_Control_System shall update the display of System temperature

  every 10 +/-1 second in degree Celsius with an accuracy of +/- 0.2 degree Celsius.

C4 – Complete Example 2

- **Unacceptable:**

  SR210: The Corporate_Website shall display Approved_Fonts

- **Improved** (Corrected as per intent)

  SR210: : The Corporate_Website shall display information using Approved_Fonts defined in Display_Standards_XXX

C5 – Singular

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

- A requirement with multiple thoughts is difficult to allocate and to trace to a parent or source.

- A nonsingular requirement is not verifiable

C5 – Singular Example

- Unacceptable:

    SR110: The control system will close the inlet_valve until the temperature has reduced to

    85 degree C, when it will then reopen it in less than 1 second

- Improved (Split in two requirements each with it's own time constraint)

    SR110: : If the Water_Temperature in the boiler increases to greater than 85 degree C, the Control_Subsystem shall close the Inlet_Valve in less than 1 second

    SR111: When the Water_Temperature in the boiler reduces to less than equal to 85 degree C, the Control_Subsystem shall open the Inlet_Valve in less than 1 second

Copyright © 2025 BlueKei Solutions Pvt Ltd

C6 – Feasible

- The requirement can be realized within entity constraints (e.g.: cost, schedule, technical, legal, ethical, safety) with acceptable risk.

- An infeasible need or requirement cannot be satisfied because
  - it breaks the laws of physics,
  - it violates laws or regulations in an applicable jurisdiction,
  - it conflicts with another requirement and cannot be concurrently satisfied, or
  - it leads to excessive program risk because of technical immaturity or inadequate cost/schedule margin

C6 – Feasible Example

- **Unacceptable:**
    SR210: The periodicity of the Engine RPM CAN Signal shall be 0.01 ms.

- **Improved** (clear the definition)
    SR210: The periodicity if the Engine RPM CAN Message shall be 5 ms.

C7 – Verifiable

- The requirement statement is structured and worded such that its realization can be verified to the approving authority's satisfaction.
    - Otherwise, there is no way to tell if it has been satisfied and that the obligation has been met.

- A requirement may not to be verifiable if there is:
    - no clear definition of the correct functional behavior, conditions, and states.
    - lack of accuracy or feasibility in the ranges of acceptable performance.
    - use of ambiguous terms.
    - no feasibility of the requirement.
    - Absolute

C7 – Verifiable Example

- **Unacceptable:**

  SR210: The Weapon_System shall not fail

- **Improved** (clear the definition)

  SR210: The Weapon_System shall have availability of greater than or equal to 95%

  or

  SR210:: The Weapon_System shall have a Mean_Time_Between_Failures of xxxx hours

C8 – Correct

- The requirement statement must be an accurate representation of the need, source, or parent requirement from which it was transformed.

- Correct implies "no errors" both from the perspective of the inclusion of incorrect information, the omission of required information, and avoidance of ambiguous wording.

- Incorrect information can mean having the wrong values, functions, conditions, or other characteristics identified in the requirement.

C9 – Conforming

- Requirements should conform to an approved standard pattern and style guide or standard for writing and managing needs and requirements.

- When requirements within the same organization have the same look and feel, each requirement statement is easier to write, understand, and review.

- The people responsible for writing the needs and requirements may need to conform to the customer's processes and standards.

Characteristics of Good Requirements (Set)

- C10 – Complete

- C11 – Consistent

- C12 – Feasible

- C13 – Comprehensible

- C14 – Able to be validated

Source: INCOSE Guide to Writing Requirements

C10 – Complete

- The requirement set for a given SOI should stand alone such that it sufficiently describes the necessary capabilities, characteristics, functionality, performance, drivers, constraints, interactions, standards, regulations, and/or quality factors without requiring other sets of requirements

- The goal is to clearly communicate the requirements for an SOI via a minimum set that are necessary and sufficient and no more

- Completeness of the set of requirements can be facilitated through the use of templates

C11 – Consistent

- The set of requirements contains individual requirements that are unique, do not conflict with or overlap with other requirements in the set

- The units and measurement systems they use are homogeneous

- The language used within the set of requirements is consistent

- Consistency in requirements wording is greatly assisted using a centralized domain ontology, glossary, and data dictionary that is shared among all stakeholders.

C12 – Feasible

- Set of requirements can be realized within entity constraints (e.g., cost, schedule, technical) with acceptable risk.

- The combination of feasible individual requirements does not necessarily imply a feasible set of those individual requirements

- If feasibility is not addressed early in the development process, it can lead to wasted effort and cost.

C13 – Comprehensible

- The set of requirements must be written such that it is clear as to what is expected of the entity and its relation to the macro system of which it is a part.

- This set must be written such that the relevant audience can understand what is being communicated
  - An agreement is difficult to enact unless both parties are clear on the exact obligation and the expected outcome(s) as a result of the realization of the entity the set of requirements represents.

C14 – Able to be Validated

- It must be able to be validated that the set of requirements will lead to the achievement of the integrated set of needs and higher-level requirements within the constraints (such as cost, schedule, technical, and regulatory compliance) with acceptable risk.

- Ask these questions:
  - "Will the entity developed by this set of requirements satisfy the needs?"
  - "Are we building the right thing?"

Rules for Writing Requirement Statements

Rules (1)

- **4.1 ACCURACY**
  - R1 - /Accuracy/SentenceStructure
  - R2 - /Accuracy/UseActiveVoice
  - R3 - /Accuracy/SubjectVerb
  - R4 - /Accuracy/UseDefinedTerms
  - R5 - /Accuracy/UseDefiniteArticles
  - R6 - /Accuracy/Units
  - R7 - /Accuracy/AvoidVagueTerms
  - R8 - /Accuracy/NoEscapeClauses
  - R9 - /Accuracy/NoOpenEnded

- **4.2 CONCISION**
  - R10 - /Concision/SuperfluousInfinitives
  - R11 - /Concision/SeparateClauses

- **4.3 NON-AMBIGUITY**
  - R12 - /NonAmbiguity/CorrectGrammar
  - R13 - /NonAmbiguity/CorrectSpelling
  - R14 - /NonAmbiguity/CorrectPunctuation
  - R15 - /NonAmbiguity/LogicalCondition
  - R16 - /NonAmbiguity/AvoidNot
  - R17 - /NonAmbiguity/Oblique

- **4.4 SINGULARITY**
  - R18 - /Singularity/SingleSentence
  - R19 - /Singularity/AvoidCombinators
  - R20 - /Singularity/AvoidPurpose
  - R21 - /Singularity/AvoidParentheses
  - R22 - /Singularity/Enumeration
  - R23 - /Singularity/Context

Rules (2)

- 4.5 COMPLETENESS
  - R24 - /Completeness/AvoidPronouns
  - R25 - /Completeness/UseOfHeadings
- 4.6 REALISM
  - R26 - /Realism/AvoidAbsolutes
- 4.7 CONDITIONS
  - R27 - /Conditions/Explicit
  - R28 - /Conditions/ExplicitLists
- 4.8 UNIQUENESS
  - R29 - /Uniqueness/Classify
  - R30 - /Uniqueness/ExpressOnce
- 4.9 ABSTRACTION
  - R31 - /Abstraction/SolutionFree
- 4.10 QUANTIFIERS
  - R32 - /Quantifiers/Universals

- 4.11 TOLERANCE
  - R33 - /Tolerance/ValueRange
- 4.12 QUANTIFICATION
  - R34 - /Quantification/Measurable
  - R35 - /Quantification/TemporalIndefinite
- 4.13 UNIFORMITY OF LANGUAGE
  - R36 - /UniformLanguage/UseConsistentTerms
  - R37 - /UniformLanguage/DefineAcronyms
  - R38 - /UniformLanguage/AvoidAbbreviations
  - R39 - /UniformLanguage/StyleGuide
- 4.14 MODULARITY
  - R40 - /Modularity/RelatedRequirements
  - R41 - /Modularity/Structured

Rules to Characteristics Mapping

Category	Definition	ID	Necessary (C1)	Appropria (C2)	unambigu (C3)	Complete (C4)	Singular (C5)	Feasible (C6)	Verifiable (C7)	Correct (C8)	Conformi (C9)	Complete (C10)	Consisten (C11)	Feasible (C12)	Comprehe (C13)	Able to be (C14)
Accuracy	SentenceStructure	R1														
	UseActiveVoice	R2			X											
	SubjectVerb	R3		X	X							X	X		X	X
	UseDefinedTerms	R4			X								X		X	X
	UseDefiniteArticles	R5			X											
	Units	R6			X					X						
	AvoidVagueTerms	R7			X											
	NoEscapeClauses	R8			X		X									
	NoOpenEnded	R9			X		X									
Concision	SuperfluousInfinitives	R10			X											
	SeparateClauses	R11			X											
Non-ambiguity	CorrectGrammar	R12									X					
	CorrectSpelling	R13			X											
	CorrectPuntuaction	R14			X											
	LogicalCondition	R15							X							
	AvoidNot	R16			X											
	Oblique	R17			X					X						
Singularity	SingleSentence	R18			X				X		X				X	
	AvoidCombinators	R19				X			X							
	AvoidPurpose	R20				X										
	AvoidParentheses	R21			X											
	Enumeration	R22				X										
	Context	R23			X		X									
Completeness	AvoidPronouns	R24			X				X							
	UseOfHeadings	R25			X	X										
Realism	AvoidAbsolutes	R26						X	X					X		
Conditions	Explicit	R27							X							
	ExplicitLists	R28			X											
Uniquess	Classify	R29										X	X			
	ExpressOnce	R30	X									X				
Abstraction	SolutionFree	R31		X												
Quantifiers	Universals	R32			X				X							
Tolerance	ValueRange	R33			X									X		
Quantification	Measurable	R34			X				X							
	TemporalIndefinite	R35			X											
Uniformity of language	UseConsistentTerms	R36							X			X			X	
	DefineAcronyms	R37							X	X						
	AvoidAbbreviations	R38							X						X	
	StyleGuide	R39				X										
Modularity	RelatedRequirements	R40							X					X		

Source: Guide for writing good requirements - Helder Castro

Rules – Accuracy (1)

- ## R1 - /Accuracy/SentenceStructure
  - Basic structure:
    The <entity> shall <action verb> <object>
  - Example:
    The water bottle shall hold 1 litre of water

- ## R2 - /Accuracy/UseActiveVoice
  - Unacceptable:
    The audio shall by played by the system
  - Acceptable:
    The system shall play audio

Rules – Accuracy (2)

- R3 - /Accuracy/SubjectVerb
  - Unacceptable:

    The User shall

  - Acceptable:

    The <system> shall

  - Unacceptable

    The car shall have legroom of…

  - Acceptable

    The Rear_Passenger_Seat shall have Legroom of…

Rules – Accuracy (3)

- **R4 - /Accuracy/UseDefinedTerms**
  - <u>Unacceptable:</u>

    The system shall display the current date

  - <u>Acceptable:</u>

    The system shall display the Current_Date in DD/MM/YYYY format

- **R5 - /Accuracy/UseDefiniteArticles**
  - <u>Unacceptable:</u>

    The system shall provide a time display

  - <u>Acceptable:</u>

    The system shall display the Current_Time

Rules – Accuracy (4)

- R6 - /Accuracy/Units

- R7 - /Accuracy/AvoidVagueTerms
  - <u>Unacceptable</u>

    The system shall … at a maximum ambient temperature of approximately 60 degrees

  - <u>Acceptable</u>

    The system shall … in the temperature range of -10 degrees Celsius to +60 degrees Celsius

Rules – Accuracy (5)

- **R8 - /Accuracy/No Escape Clauses**
  - <u>Unacceptable:</u>

    The system shall, as far as possible, conform to Clause a) of Standard ABC

  - <u>Acceptable:</u>

    The system shall conform to Clause a) of Standard ABC

- **R9 - /Accuracy/NoOpenEnded**
  - Unacceptable:

    The display_unit shall display date, time, temperature, etc.

  - Acceptable:

    The display unit shall display Current_Date
    The display unit shall display Current_Time
    The display unit shall display Current_Temperature

Rules - Concision

- R10 - /Concision/Superfluous Infinitives
  - Unacceptable

    The system shall be designed to be able to…

  - Acceptable

    The system shall…

- R11 - /Concision/Separate Clauses

Rules – Singularity (1)

- R18 - /Singularity/SingleSentence

- R19 - /Singularity/AvoidCombinators

- R20 - /Singularity/AvoidPurpose
  - <u>Unacceptable:</u>
    The system shall… because
  - <u>Acceptable:</u>
    The system shall…

- R21 - /Singularity/AvoidParentheses

Rules – Singularity (2)

- **R22 - /Singularity/Enumeration**
  - Unacceptable:

    The system shall display performance information

  - Acceptable:

    The system shall display parameter #1
    The system shall display parameter #2

- **R23 - /Singularity/Context**
  - Unacceptable:

    The system shall… X while Y within Z when A

  - Acceptable:

    The system shall… as specified in Diagram #1

Rules - Completeness

- **R24 - /Completeness/AvoidPronouns**
    - Unacceptable:

        The system shall receive from the user, his inputs

    - Acceptable:

        The system shall receive inputs from User

- **R25 - /Completeness/UseOfHeadings**
    - Unacceptable:

        5 Airbag Requirements

        5.1 The car shall deploy them within X seconds of impact

    - Acceptable:

        The car shall deploy Airbags in less than X seconds after Impact

Rules – Realism, Conditions

- **R26 - /Realism/AvoidAbsolutes**
  - ○ <u>Unacceptable:</u>

    The system shall never…

  - ○ <u>Acceptable:</u>

    The system shall… greater than or equal to 95%

- **R27 - /Conditions/Explicit**
  - ○ <u>Unacceptable:</u>

    When X happens, the power shall be turned off

  - ○ <u>Acceptable:</u>

    In the event of X, the system shall turn off power

Rules - Conditions, Uniqueness

- R28 - /Conditions/ExplicitLists

- R29 - /Uniqueness/Classify

- R30 - /Uniqueness/ExpressOnce
  - <u>Unacceptable:</u>
    - The system shall generate a report of financial transactions
    - The system shall generate a Financial_Transaction_Report

Rules – Abstraction, Quantifiers

- R31 - /Abstraction/SolutionFree

- R32 - /Quantifiers/Universals
  - <u>Unacceptable:</u>

    The Operation_Logger shall record any (or all) warning messages

  - <u>Acceptable:</u>

    The Operation_Logger shall record each Warning_Message.

Rules – Tolerance, Quantification

- **R33 - /Tolerance/ValueRange**
  - Unacceptable:

    The system shall… at acceptable levels

  - Acceptable:

    The system shall… within the range X to Y

- **R34 - /Quantification/Measurable**
- **R35 - /Quantification/TemporalIndefinite**
  - Unacceptable:

    The system shall…. Immediately

  - Acceptable

    The system shall…. Within X seconds of Event_A

Rules – Uniformity of Language

- R36 - /UniformLanguage/UseConsistentTerms

- R37 - /UniformLanguage/DefineAcronyms

- R38 - /UniformLanguage/AvoidAbbreviations

- R39 - /UniformLanguage/StyleGuide

Rules - Modularity

- R40 - /Modularity/Related Requirements

- R41 - /Modularity/Structured

Stakeholder Requirements to System Requirements

Needs to Requirements

Stakeholder Needs

Stakeholder Needs

Stakeholder Needs

Problem Domain

Solution Domain

Stakeholder Requirements

System Requirements

What Stakeholders **EXPECT** from the System

What the System must **DO / FULFILL**
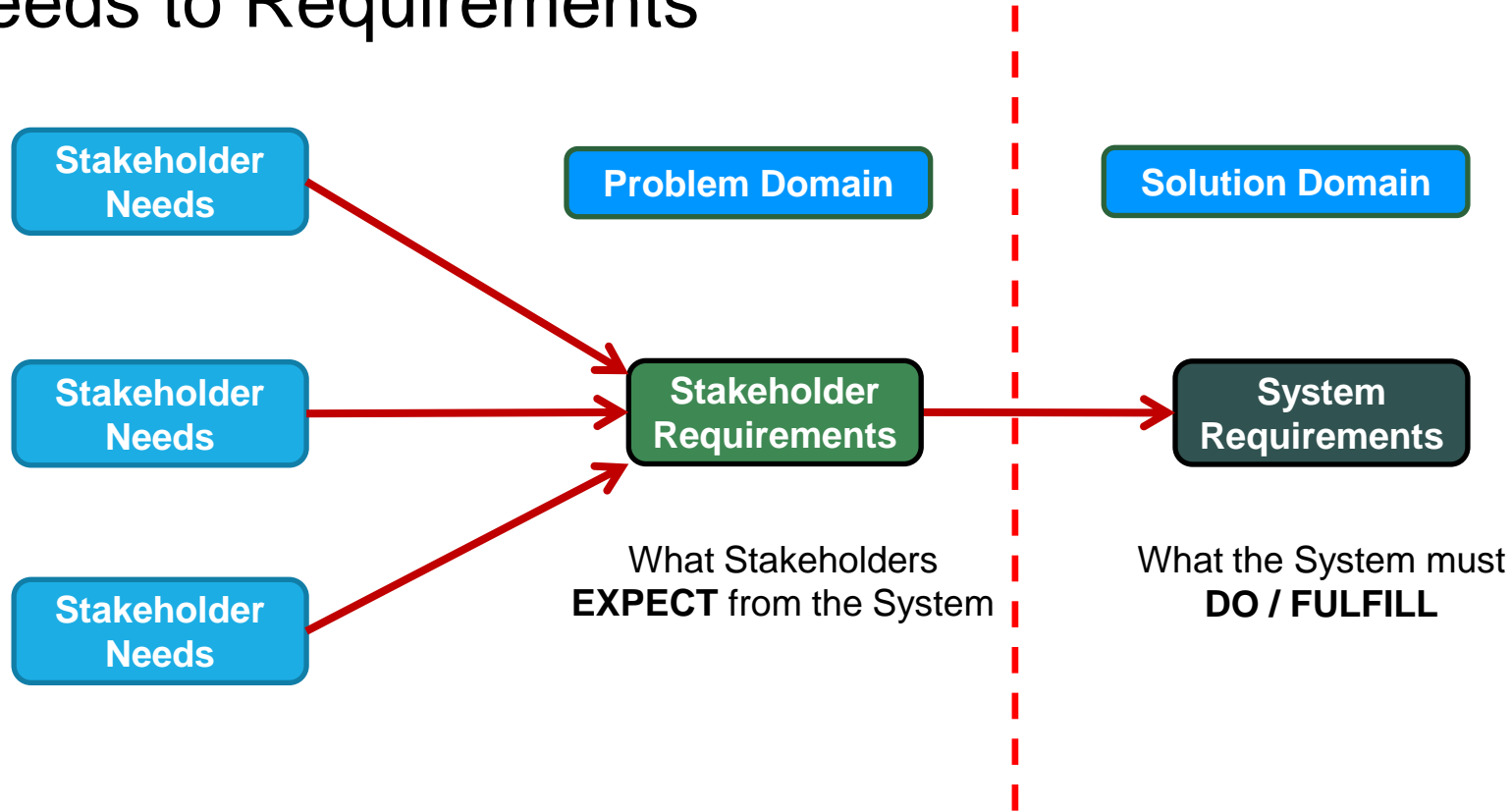
System Requirements Classification

1. Functional Requirements
2. Performance Requirements
3. Usability Requirements
4. Interface Requirements
5. Operational Requirements
6. Modes and/or States Requirements
7. Adaptability Requirements

8. Physical Constraints
9. Design Constraints
10. Environmental Conditions
11. Logistical Requirements
12. Policies and Regulations
13. Cost and Schedule Constraints
14. Regulatory Requirements

Source: SEBoK

System Requirements Classification

Types of System Requirement	Description
Usability Requirements	Define the quality of system use (e.g. measurable effectiveness, efficiency, and satisfaction criteria).
Interface Requirements	Define how the system is required to interact or to exchange material, energy, or information with external systems (external interface), or how system elements within the system, including human elements, interact with each other (internal interface). Interface requirements include physical connections (physical interfaces) with external systems or internal system elements supporting interactions or exchanges.
Operational Requirements	Define the operational conditions or properties that are required for the system to operate or exist. This type of requirement includes: human factors, ergonomics, availability, maintainability, reliability, and security.
Modes and/or States Requirements	Define the various operational modes of the system in use and events conducting to transitions of modes.
Adaptability Requirements	Define potential extension, growth, or scalability during the life of the system.
Physical Constraints	Define constraints on weight, volume, and dimension applicable to the system elements that compose the system.
Design Constraints	Define the limits on the options that are available to a designer of a solution by imposing immovable boundaries and limits (e.g., the system shall incorporate a legacy or provided system element, or certain data shall be maintained in an online repository).
Environmental Conditions	Define the environmental conditions to be encountered by the system in its different operational modes. This should address the natural environment (e.g. wind, rain, temperature, fauna, salt, dust, radiation, etc.), induced and/or self-induced environmental effects (e.g. motion, shock, noise, electromagnetism, thermal, etc.), and threats to societal environment (e.g. legal, political, economic, social, business, etc.).
Logistical Requirements	Define the logistical conditions needed by the continuous utilization of the system. These requirements include sustainment (provision of facilities, level support, support personnel, spare parts, training, technical documentation, etc.), packaging, handling, shipping, transportation.
Policies and Regulations	Define relevant and applicable organizational policies or regulatory requirements that could affect the operation or performance of the system (e.g. labor policies, reports to regulatory agency, health or safety criteria, etc.).
Cost and Schedule Constraints	Define, for example, the cost of a single exemplar of the system, the expected delivery date of the first exemplar, etc.

Source : SEBOK WIKI

Interface Requirements

1. The interaction could be direct (actual connection between two systems or
2. Indirect (no direct connection but there is some design feature of a system that can affect a design feature of another system, e.g., induced environments or competition for a common resource).
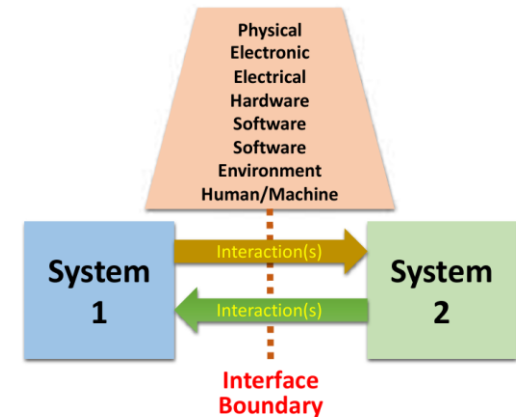

Figure 6-7: An Interface is a Boundary, not a Thing.

What an Interface is not!

- A general rule is that the word "interface" should not be used in a requirement statement either as a noun or a verb.
  - As a noun, it implies the interface is a thing, which it is not - it is a boundary across which, or at, two systems interact.
  - As a verb, it is ambiguous, in that often there are multiple interactions between systems across a single interface boundary.

- It is a best practice to focus on individual interactions when writing interface requirements. This is important from both a system verification perspective and an allocation perspective.

INCOSE-TP-2021-002-01 | VERS/REV: 1.1 | May 2022

Example of how not to write Interface Requirements

Incorrect Requirement	Explanation	System A Requirement	System B Requirement
The digital interface shall maintain full operational capability after two failures.	This requirement assumes the interface is a tangible thing with functionality - this is not true.	The digital system on the spacecraft side shall maintain operational capability by handling up to two failures in data exchange.	The digital system on the payload side shall maintain operational capability by handling up to two failures in data exchange.
The interfaces between the spacecraft and payload shall be designed to	This requirement is written in passive voice on the designers and also assumes the interfaces are things. The requirement should focus on accessibility of connectors, bolts, etc.	The spacecraft shall provide accessible connectors and bolts for attachment to the payload.	The payload shall provide accessible connectors and bolts for attachment to the spacecraft.
The interfaces between the spacecraft and payload shall have standard labels, controls, and displays.	This requirement is written in passive voice and again assumes the interface is a thing.	The spacecraft shall include standardized labels, controls, and displays at the interface for ease of use.	The payload shall include standardized labels, controls, and displays at the interface for ease of use.
The electrical interface between the spacecraft and payload shall have a reliability of 0.99999.	This requirement again assumes the interface is a thing. The requirement should address each of the systems and apply to any hardware or software involved in interactions on each side of the interface boundary.	The electrical system on the spacecraft side shall maintain a reliability of 0.99999 in operations involving the payload.	The electrical system on the payload side shall maintain a reliability of 0.99999 in operations involving the spacecraft.
The SOI shall interface with	This requirement is ambiguous because it does not focus on a specific interaction. There are often multiple interactions; to be specific, there should be a requirement for each interaction.	The SOI shall enable data exchange with the payload's telemetry system.	The payload shall establish a secure data connection with the SOI for telemetry exchange.

Steps to write Interface Requirements

Writing interface requirements is a three-step process:

Step 1: Identify the interface boundaries and interactions across those boundaries

Step 2: Define the interactions across the interface boundaries

Step 3: Write the Interface requirements.

INCOSE-TP-2021-002-01 | VERS/REV: 1.1 | May 2022

Examples of Requirements

The <system> shall not fail.

{This is unacceptable because verification of the requirement would require infinite time.}

Acceptable:

The <system> shall have an Availability of greater than or equal to 95%.

OR

The <system> shall have a Mean Time Between Failures (MTBF) of 6 months.

INCOSE - Guide for Writing Requirements

Examples of Requirements

The User_Management_System shall Open/Close the User_Account in less than 1 second.

{This is unacceptable because it is unclear as to what is meant: open, close, or both?}

Acceptable (Split into two requirements):

The User_Management_System shall Open the User_Account in less than 1 second.

The User_Management_System shall Close the User_Account in less than 1 second.
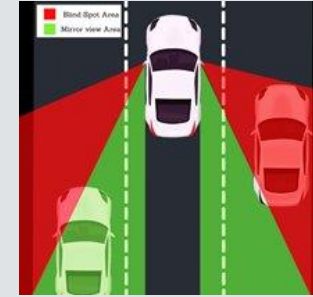
INCOSE - Guide for Writing Requirements

Good Requirements or Bad Requirements?

1. The smart watch shall be very nice looking and easy to use.

2. The smart watch shall work without charging for 10 hours.

3. The smart watch display shall be circular with 1 inch diameter.

Exercise 3.1

For your **BSD** ,

1. Transform some Stakeholder Requirements into **System Requirements** (*following characteristics of good system requirements, and using the format "The <system> shall…"*) while doing so please rectify the stakeholder requirements according to good requirement rules

    ○ Indicate one Stakeholder Requirement in one row

    ○ Some Stakeholder Requirements may translate into more than one System Requirement

SOFTWARE REQUIREMENTS AND HW REQUIREMENTS
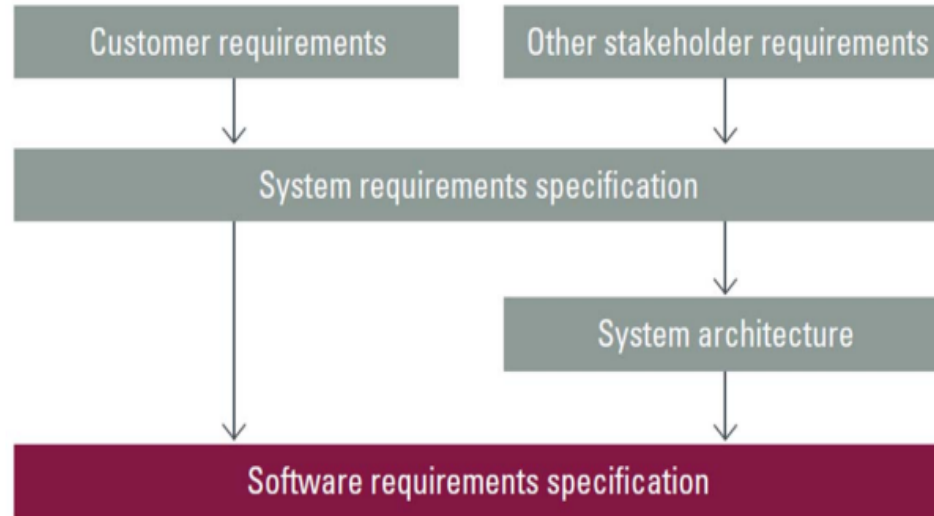
Sources for the determination of software requirements

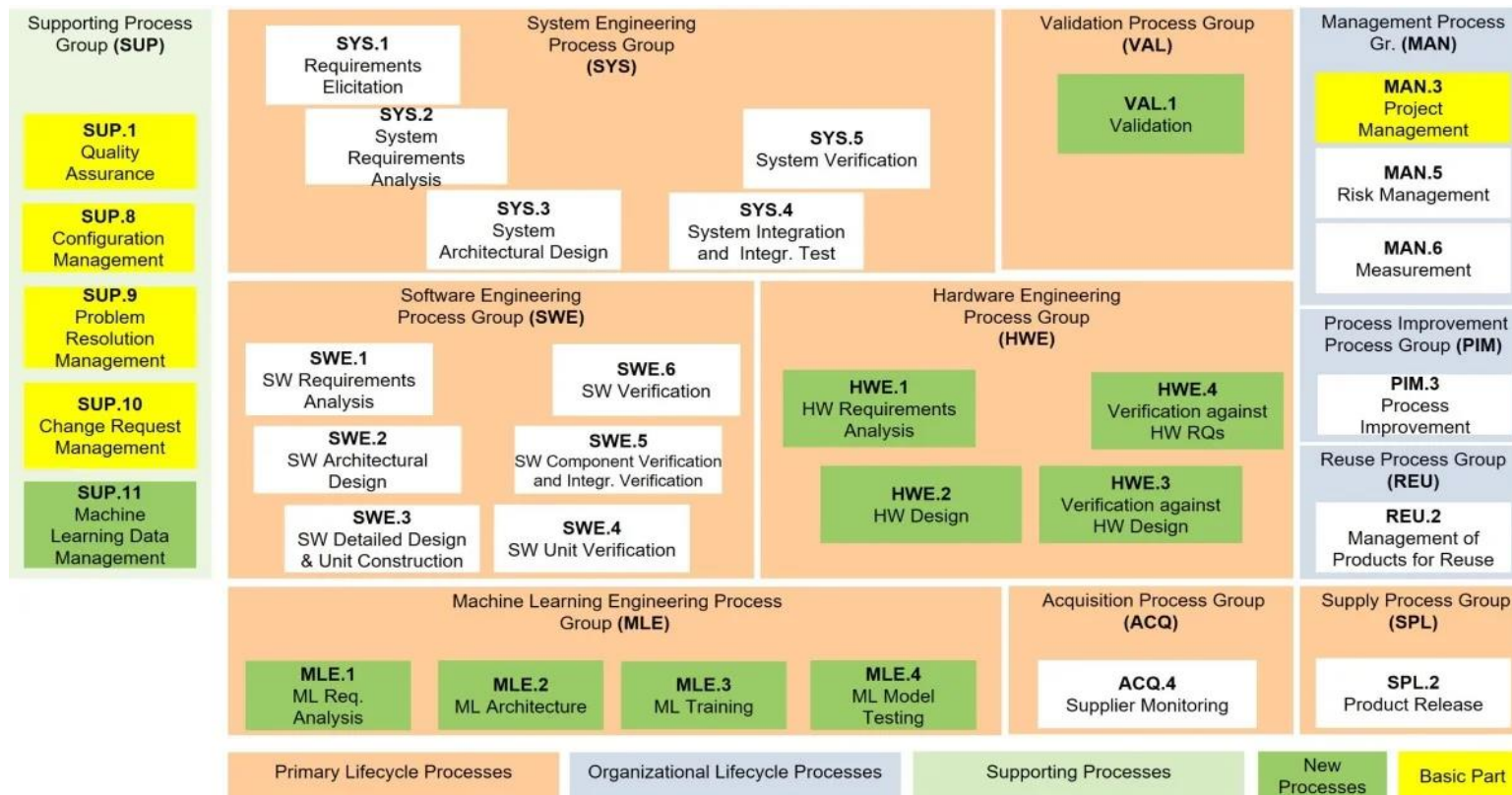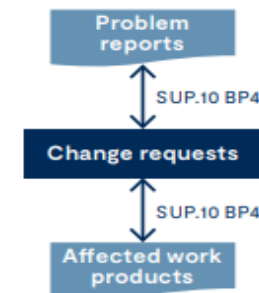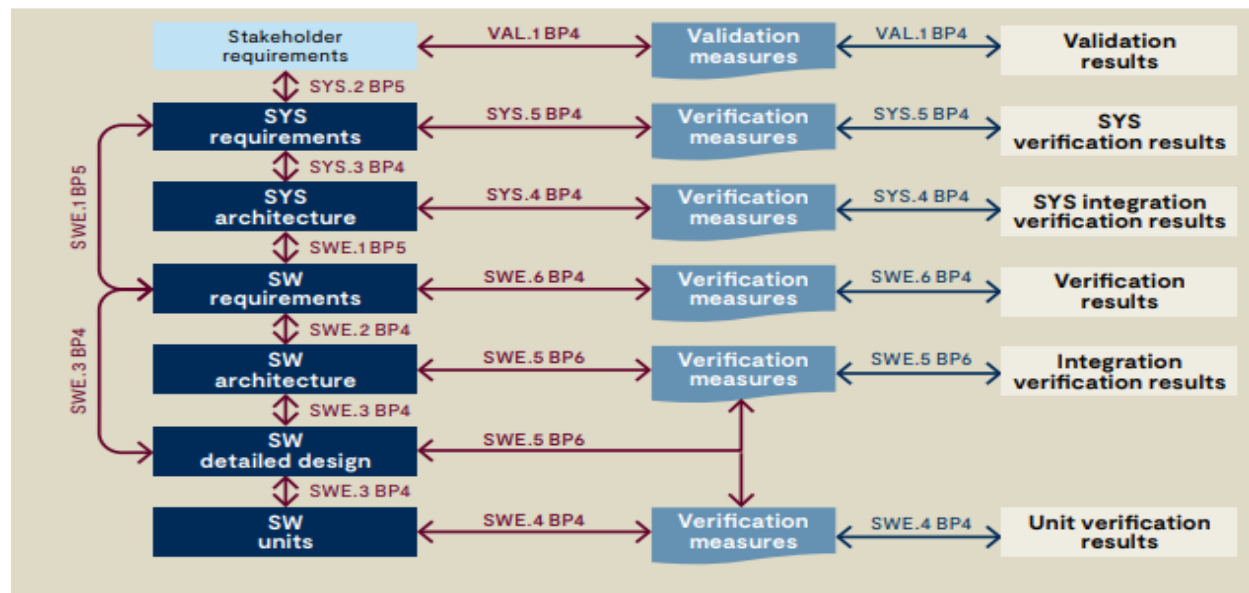# Automotive SPICE process reference model

image source: Invensity

REQUIREMENTS SYS-SW TRACEABILITY AND CONSISTENCY
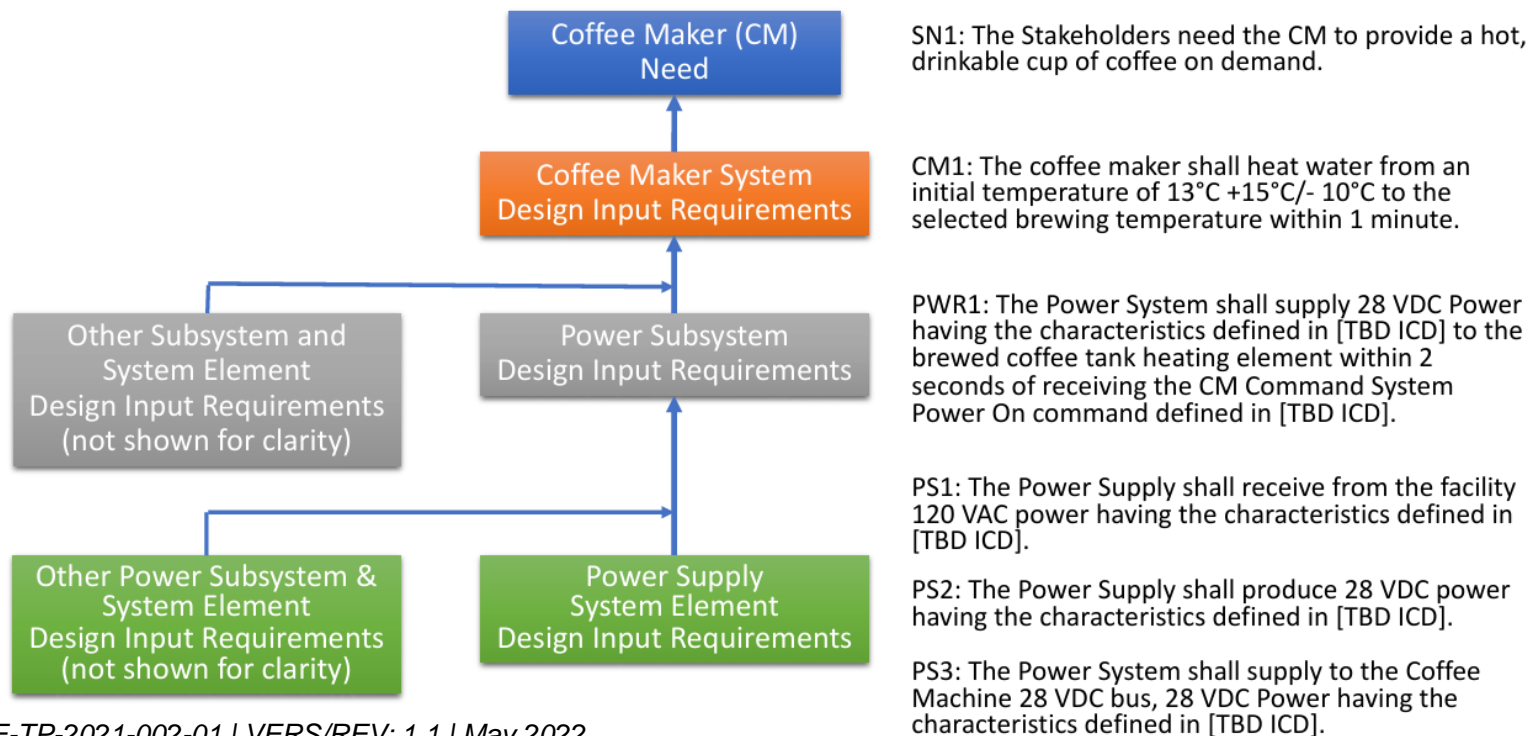
REQUIREMENTS SYS-HW TRACEBILITY AND CONSISTENCY
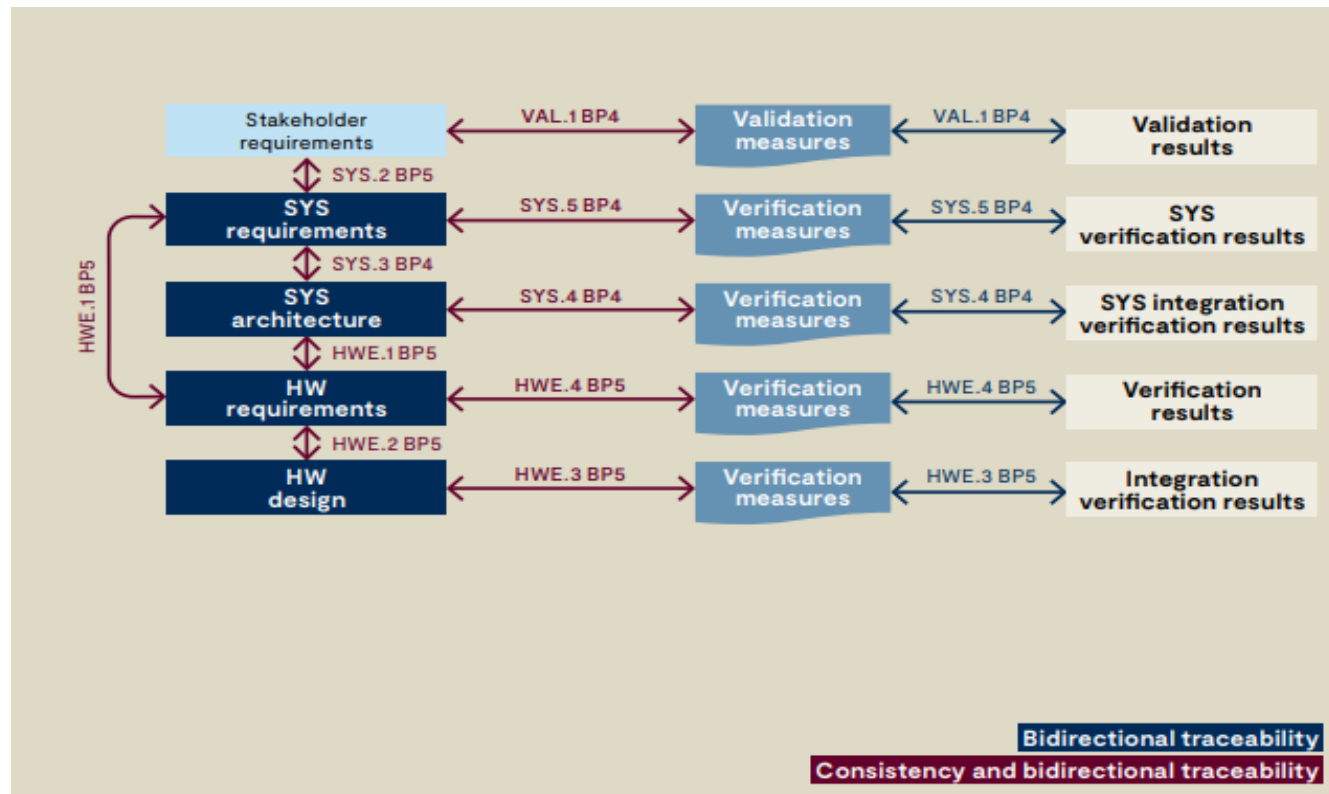
Consistency and traceability between system and hardware work products

Needs -> STK Req ->Sys Req -> Sub-Sys Req

Coffee Maker (CM) Need

SN1: The Stakeholders need the CM to provide a hot, drinkable cup of coffee on demand.

Coffee Maker System Design Input Requirements

CM1: The coffee maker shall heat water from an initial temperature of 13°C +15°C/- 10°C to the selected brewing temperature within 1 minute.

Other Subsystem and System Element Design Input Requirements (not shown for clarity)

Power Subsystem Design Input Requirements

PWR1: The Power System shall supply 28 VDC Power having the characteristics defined in [TBD ICD] to the brewed coffee tank heating element within 2 seconds of receiving the CM Command System Power On command defined in [TBD ICD].

Other Power Subsystem & System Element Design Input Requirements (not shown for clarity)

Power Supply System Element Design Input Requirements

PS1: The Power Supply shall receive from the facility 120 VAC power having the characteristics defined in [TBD ICD].

PS2: The Power Supply shall produce 28 VDC power having the characteristics defined in [TBD ICD].

PS3: The Power System shall supply to the Coffee Machine 28 VDC bus, 28 VDC Power having the characteristics defined in [TBD ICD].

INCOSE-TP-2021-002-01 | VERS/REV: 1.1 | May 2022

Architecture for Requirements Breakdown

Architecture

"The fundamental organization of a system, embodied in its components, their relationships to each other and the environment, and the principles governing its design and evolution."
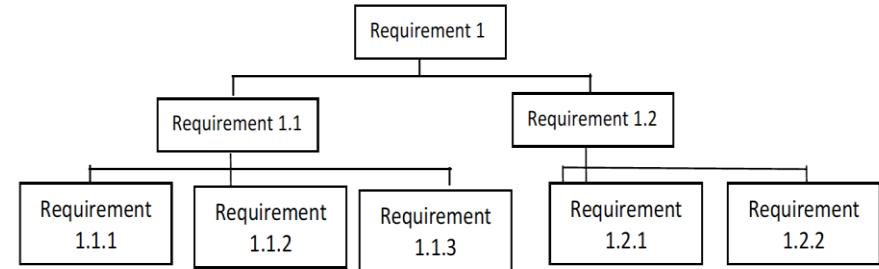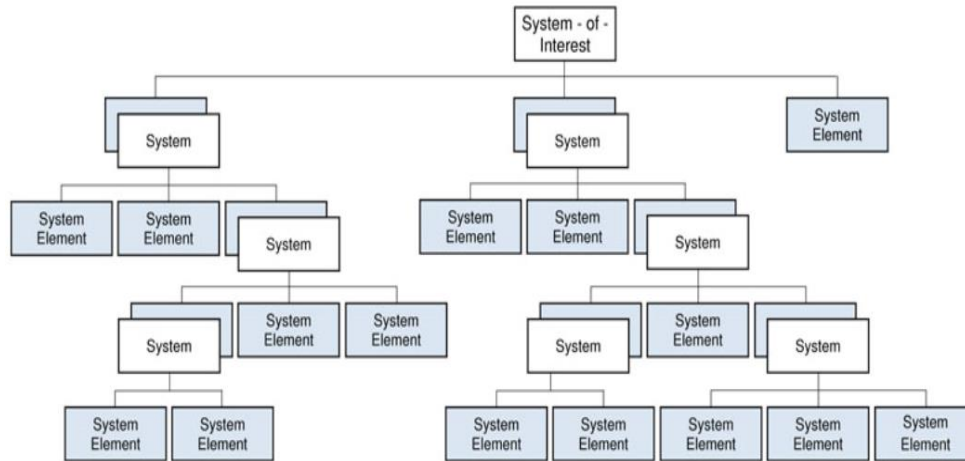
- ISO/IEC/IEEE 42010

Architecture

Defines…

- System functionality
- System boundaries relative to other systems
- External interfaces with other systems
- All system components
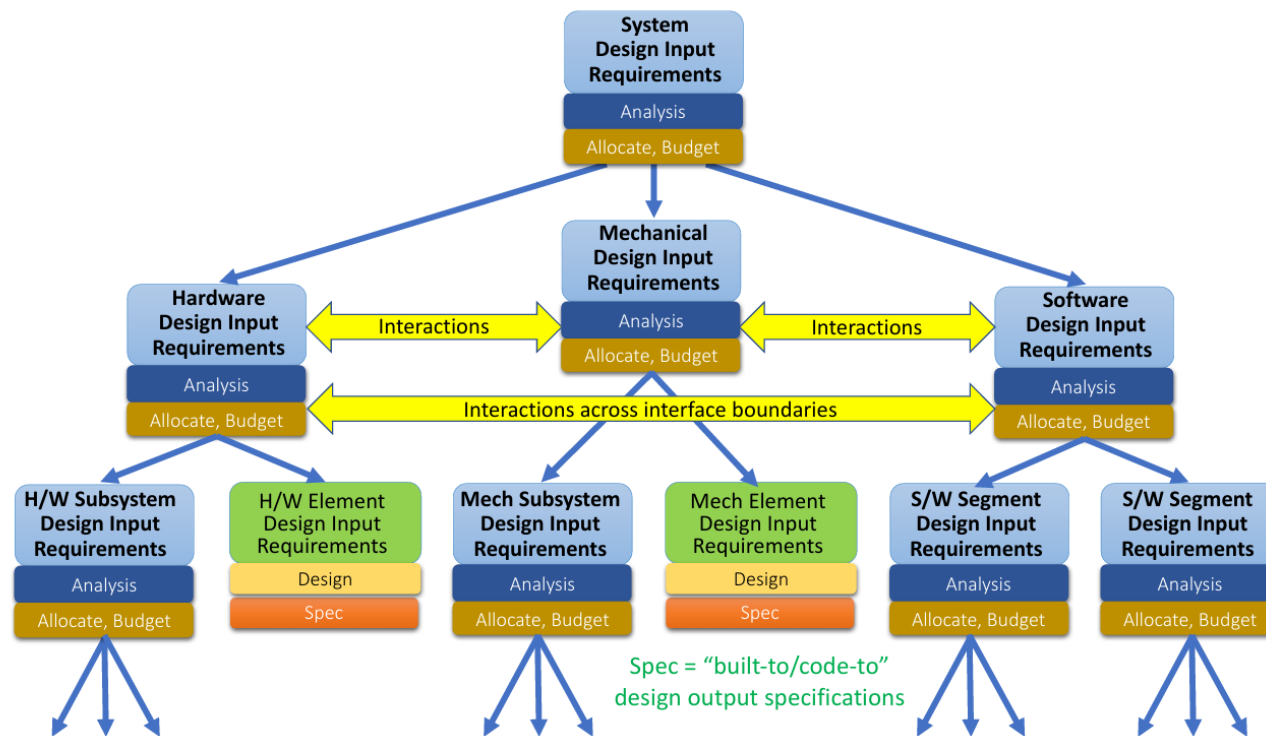- Internal interfaces between components

System Hierarchy

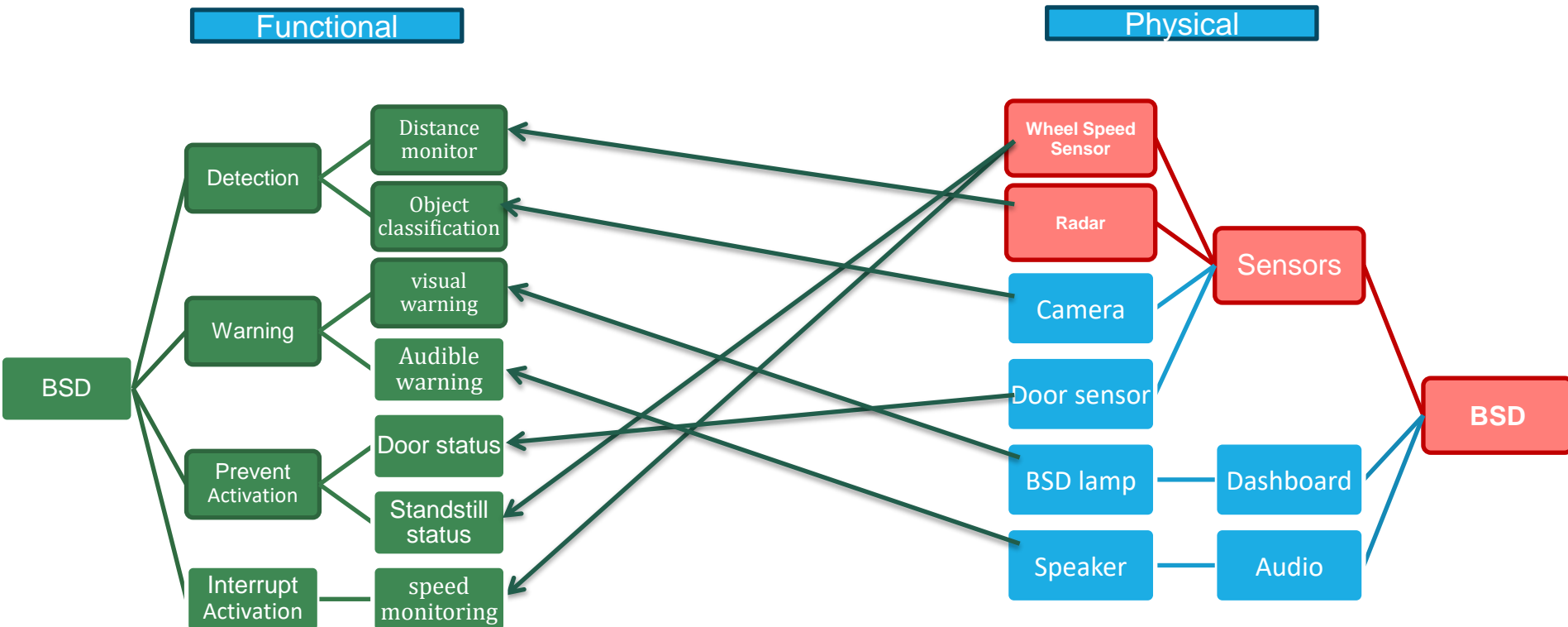Figure 2-3. The hierarchical elaboration of requirements.

Source : SEBOK

Relationship b/w Requirements for SW intensive Systems

INCOSE-TP-2021-002-01 | VERS/REV: 1.1 | May 2022
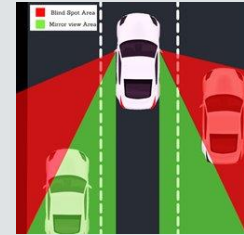
High Level Module Design: Physical Decomposition & Mapping

Discussion

1. Create a Ref Architecture of ADAS system

2. Use the same to derive next set of HW and SW requirements

Exercise 3.2

For your **<u>BSD</u>** ,

1. transform some Systems Requirements into **Software Requirements** (*following characteristics of good sw requirements, and using the format "The <SW> shall…"*)

    ○ Indicate one req in one row

    ○ Some System Requirements may translate into more than one Software Requirement

Exercise 3.3

For your **BSD** ,
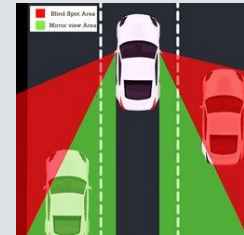
1. transform some Systems Requirements into **Hardware Requirements** (*following characteristics of good hw requirements, and using the format "The <SW> shall…"*)

   ○ Indicate one req in one row

   ○ Some System Requirements may translate into more than one HW Requirement

The End