IEA-P – DEPARTAMENTO DE PROJETOS

(PROJECT DEPARTMENT)

Requirements Engineering Review

[2025]

Prof. Dr. Christopher S. Cerqueira

Fevereiro

	S	T	Q	Q	S	S	D
						1	2
	3	4	5	6	7	8	9
	10	11	12	13	14	15	16
	17	18	19	20	21	22	23
~~1~~	~~24~~	~~25~~	~~26~~	~~27~~	~~28~~		

Março

	S	T	Q	Q	S	S	D
1						1	2
2	**3**	**4**	**5**	6	7	8	9
3	10	11	12	13	14	15	16
4	17	18	**19**	20	21	22	23
5	**24**	**25**	26	27	28	29	30
6	31						

Abril

	S	T	Q	Q	S	S	D
6		1	2	3	4	5	6
7	7	8	9	10	11	12	13
8	14	15	16	17	**18**	☹	**20**
s	**21**	22	23	24	25	26	27
1	28	29	30				

Maio

	S	T	Q	Q	S	S	D
1				**1**	**2**	☹	4
2	5	6	7	8	9	10	11
3	12	13	14	15	16	17	18
4	19	20	21	22	23	24	25
5	26	27	28	29	30	31	

Junho

	S	T	Q	Q	S	S	D
5							1
6	2	3	4	5	6	7	8
7	9	10	11	12	13	14	15
8	16	17	18	**19**	20	☹	22
Ex1	23	24	25	26	27	28	29
Ex2	30						

Julho

	S	T	Q	Q	S	S	D
Ex2		1	2	3	4	5	6
	7	8	**9**	10	11	12	13
	14	15	16	17	18	19	**20**
	21	22	23	24	25	26	**27**
1	28	29	30	31			

Theory

WEEK	CLASS ACTIVITY	INDIVIDUAL	W	GROUP	W
1 27-Feb 28-Feb	Course Structure and Initial Definitions Systems Engineering Review	IA-01 - Reading and Conceptual Questions (10)	5%	GA-01 - Define Groups	0%
2 06-Mar 07-Mar	Requirements Engineering Review	IA-02 - Reading and Conceptual Questions (10)	5%	GA-02 - Requirement writing and requirement validation (cross-check inter-groups).	10%
3 13-Mar 14-Mar	Risk - Hazard Analysis Review	IA-03 - Reading and Conceptual Questions (10)	5%	GA-03 - Raise overall mission risks and specific to the subsystem	0%
4 20-Mar 21-Mar	AIVV Introduction	IA-04 - Reading and Conceptual Questions (10)	5%	GA-04 -	20%
5 27-Mar 28-Mar	Means of Compliance Success Criteria	IA-05 - Reading and Conceptual Questions (10)	5%	GA-05 - Write the means of verification and the success criteria.	20%
6 03-Apr 04-Apr	Structuring the Life Cycle Technical Reviews	IA-06 - Reading and Conceptual Questions (10)	5%	GA-06 - Structure a lifecycle and the content of the reviews.	20%
7 10-Apr 11-Apr	Model Philosophy and Environmental Testings Questions?	IA-07 - Reading and Conceptual Questions (10)	10%	GA-07 - Structure the model philosophy	30%
8 17-Apr 18-Apr	P1 - Conceptual Questions and Case	IA-08 - Select 20 questions and a mini-case to build a coherent DVM (RTM)	60%	GA-08 -	
			100%		100%

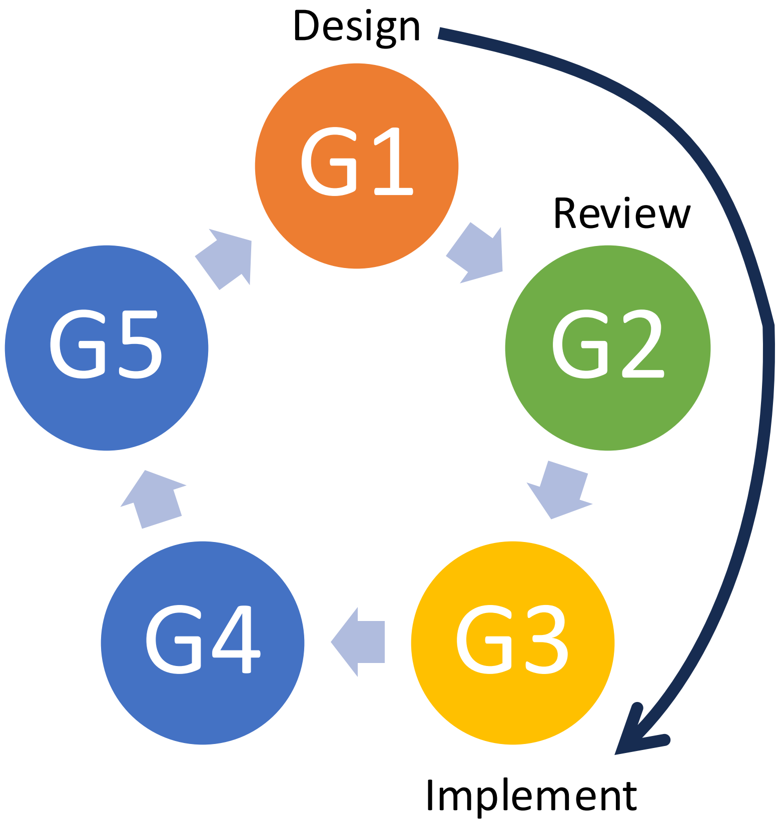
Async

☹

Requirements perception experiment

Split in groups and use the first half of the first time to todays activity

- Today:
  - Create a system with max of **20 pieces**. (take Picture and do not show to the other groups – return the pieces)
  - Describe it with **10 requirements**, save one copy and give it to other group.
- Homework:
  - Correct the requirements of the group with the NASA checklist that I’ll show on the last section.
  - Search for EARS boilerplate/use the example on the slide to correct the formats.
  - You can use AI, but.... I want the whole prompt session
- Tomorrow:
  - Give the corrected requirements to other group and ask them to recreate the system. (take pictures to compare)
- Next week:
  - GA-02 - Group report with the experience



REQUIREMENT IMPORTANCE

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.

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).

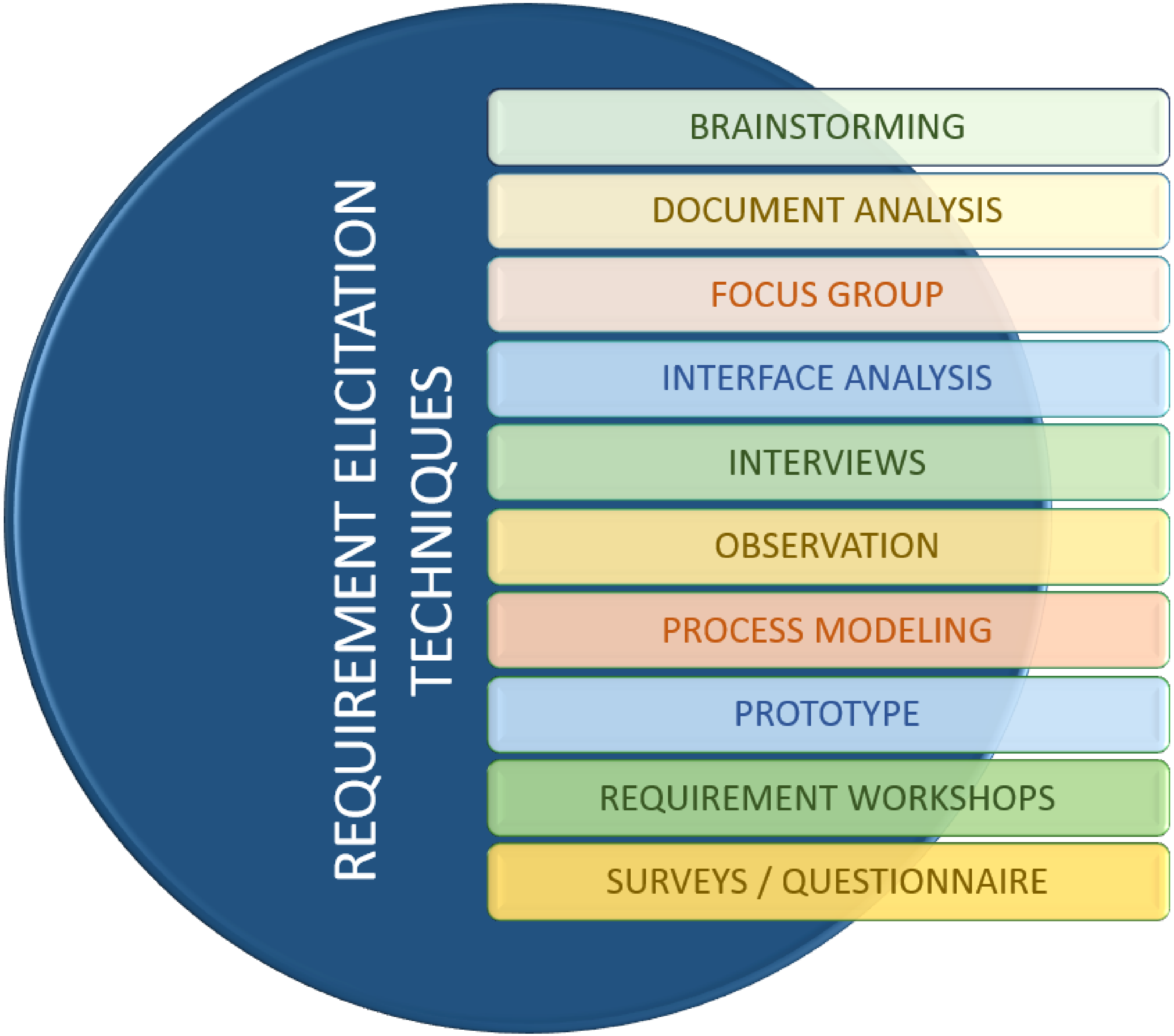
I DON'T UNDERSTAND ANYTHING YOU DO, SO I ASSUME IT'S ALL USELESS.

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.

REQUIREMENTS ARE A HUMAN ISSUE

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



REQUIREMENT ELICITATION TECHNIQUES

- BRAINSTORMING
- DOCUMENT ANALYSIS
- FOCUS GROUP
- INTERFACE ANALYSIS
- INTERVIEWS
- OBSERVATION
- PROCESS MODELING
- PROTOTYPE
- REQUIREMENT WORKSHOPS
- SURVEYS / QUESTIONNAIRE

TRANSFORM STAKEHOLDER NEEDS/GOALS INTO REQUIREMENTS

- Identify Stakeholders and their **expectations/needs** (“what’s an ‘NGO?”).
  - A Stakeholder is defined as anyone **affected by or accountable** for the system
  - Stakeholders' **inputs may be very informal** (e.g. an interview or e-mail) or may be documented and provided to you.

- Use **tools to drive out** needed requirements.
  - ConOps, Design Reference Mission, Reference Mission Scenario.
  - Conceptual Design

- Expect to **iterate**.
- Identify **Constraints and challenge them** (“Congress is designing my rocket!!!”).
  - Even if you can’t change/eliminate a constraint you can at least understand the ‘cost’ and risk of having to live with it (which may give you the ‘ammo’ you need to challenge it).

CLASS EXAMPLE: 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

REQUIREMENT CRITERIAS

NECESSARY

- The stated requirement is an essential capability, physical characteristic, or quality factor of the product or process. If it is removed or deleted, a deficiency will exist, which cannot be fulfilled by other capabilities of the product or process.

- An example of a necessary requirement for a combat vehicle could be "The vehicle's combat loaded weight shall not exceed 35 Tons".

CONCISE (MINIMAL, UNDERSTANDABLE)

- The requirement statement includes only one requirement stating what must be done and only what must be done, stated simply and clearly. It is easy to read and understand.

- An example of a requirement can be "The element shall provide a visual and audible alarm under all conditions listed in Table 3-10." "The alarm shall be activated no longer than 1 second after the conditions exists."

IMPLEMENTATION FREE

- The requirement states what is required, not how the requirement should be met. A requirement statement should not reflect a design or implementation, nor should it describe an operation. However, the treatment of interface requirements is generally an exception.

- An example of a requirement for a shipboard system at the system level is: "The system shall be capable of engaging sea skimming anti ship cruise missile (ASCM) targets.“
- The allocated requirement for the radar element can be: "In a clear environment, the radar element shall be capable of detecting 0.1 (meter)2 ASCM targets at ranges of up to 20 KM with a probability of detection of no less than 0.9 and probability of false alarm of no greater than 10-6.”

ATTAINABLE (ACHIEVABLE OR FEASIBLE)

- The stated requirement can be achieved by one or more developed system concepts at a definable cost. This implies that at least a high-level conceptual design has been completed, and cost tradeoff studies have been conducted.

- Consider the following requirement for a radar element: "In a clear environment, the radar element shall be capable of detecting 0.1 (meter)2 ASCM targets at ranges of up to 20 KM with a probability of detection of no less than 0.9 and probability of false alarm of no greater than 10-6." It is well known that radars operating in the microwave frequencies are more or less horizon limited due to straight line propagation of electromagnetic waves at these frequencies

COMPLETE (STANDALONE)

- The stated requirement is complete and does not need further amplification. The stated requirement will provide sufficient capability.

- Let us examine a basic statement: "The vehicle shall permit growth" as an initial idea in trying to specify growth. The question is how much growth and in which areas? A requirement statement, for example, in the area of weight growth, may then be stated as follows: "The vehicle shall permit 12% weight growth."

CONSISTENT

- The stated requirement does not contradict other requirements. It is not a duplicate of another requirement. The same term is used for the same item in all requirements.

- Connectivity refers to the property whereby all of the terms within the requirement are adequately linked to other requirements and to word and term definitions, so causing the individual requirement to properly relate to the other requirements as a set.

UNAMBIGUOUS

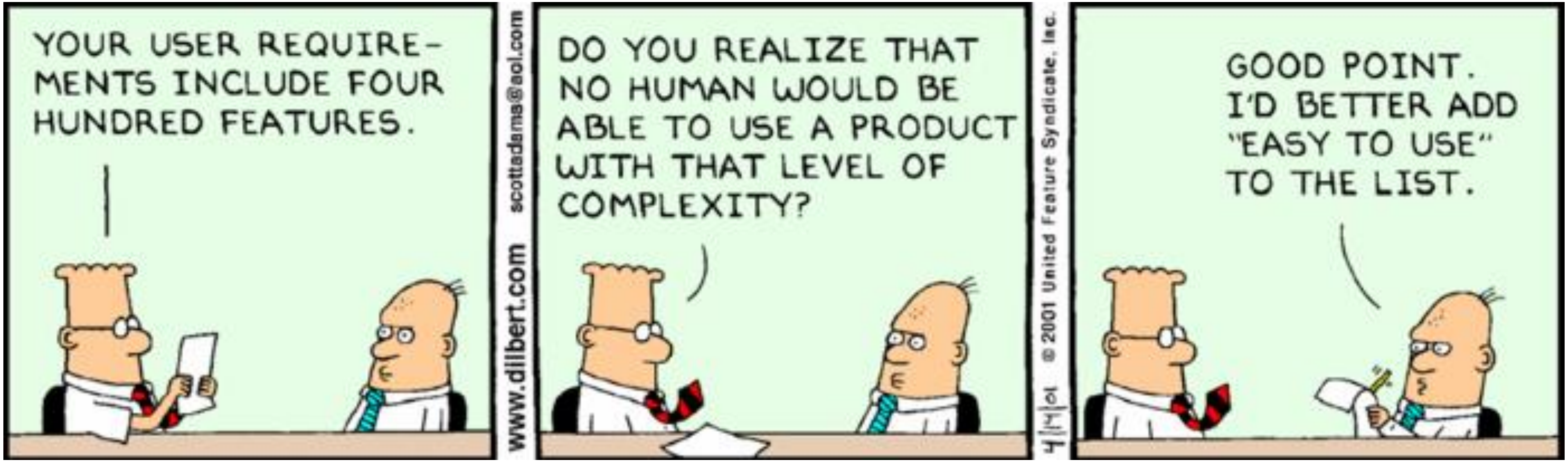
- Each requirement must have one and only one interpretation. Language used in the statement must not leave a doubt in the reader's mind as to the intended descriptive or numeric value.

VERIFIABLE

- The stated requirement is not vague or general but is quantified in a manner that can be verified by one of these 4 alternative methods: inspection, analysis, demonstration or test.

- In order to be verifiable, the requirement should be stated in measurable terms such as: "The overall length of the system shall be 105+/-0.5 inches" (verifiable by inspection) or "Gun alignment error in elevation shall be no greater than 1.5 milliradians"(verifiable by test).

REQUIREMENT ORGANIZATION



YOUR USER REQUIREMENTS INCLUDE FOUR HUNDRED FEATURES.
scottadams@aol.com
www.dilbert.com
DO YOU REALIZE THAT NO HUMAN WOULD BE ABLE TO USE A PRODUCT WITH THAT LEVEL OF COMPLEXITY?
© 2001 United Feature Syndicate, Inc.
4/14/01
GOOD POINT. I'D BETTER ADD "EASY TO USE" TO THE LIST.

REQUIREMENT TREE

- Understand the Requirements Structure of your Project
  - Identify Parent Requirement Documents and Requirements Sources
  - Identify Peer and Interface Requirements
  - Identify Children Requirement Documents

- Note that at the ‘Document Level’, the above will start to identify ‘Stakeholders’

DECOMPOSITION AND DERIVATION OF REQUIREMENTS

- **As requirements are developed at each level of the requirements hierarchy**, you start with the requirements allocated to you (your Parent requirements) but that is not where you stop!
  - The same process, identification of Stakeholders, a ConOps, etc., **must be applied at every level** of the requirements hierarchy.
  - The **size & depth** of those activities may diminish, and some products may be combined or informal, but you must go through the same thought process to ensure **each level's requirement set is complete and correct**.

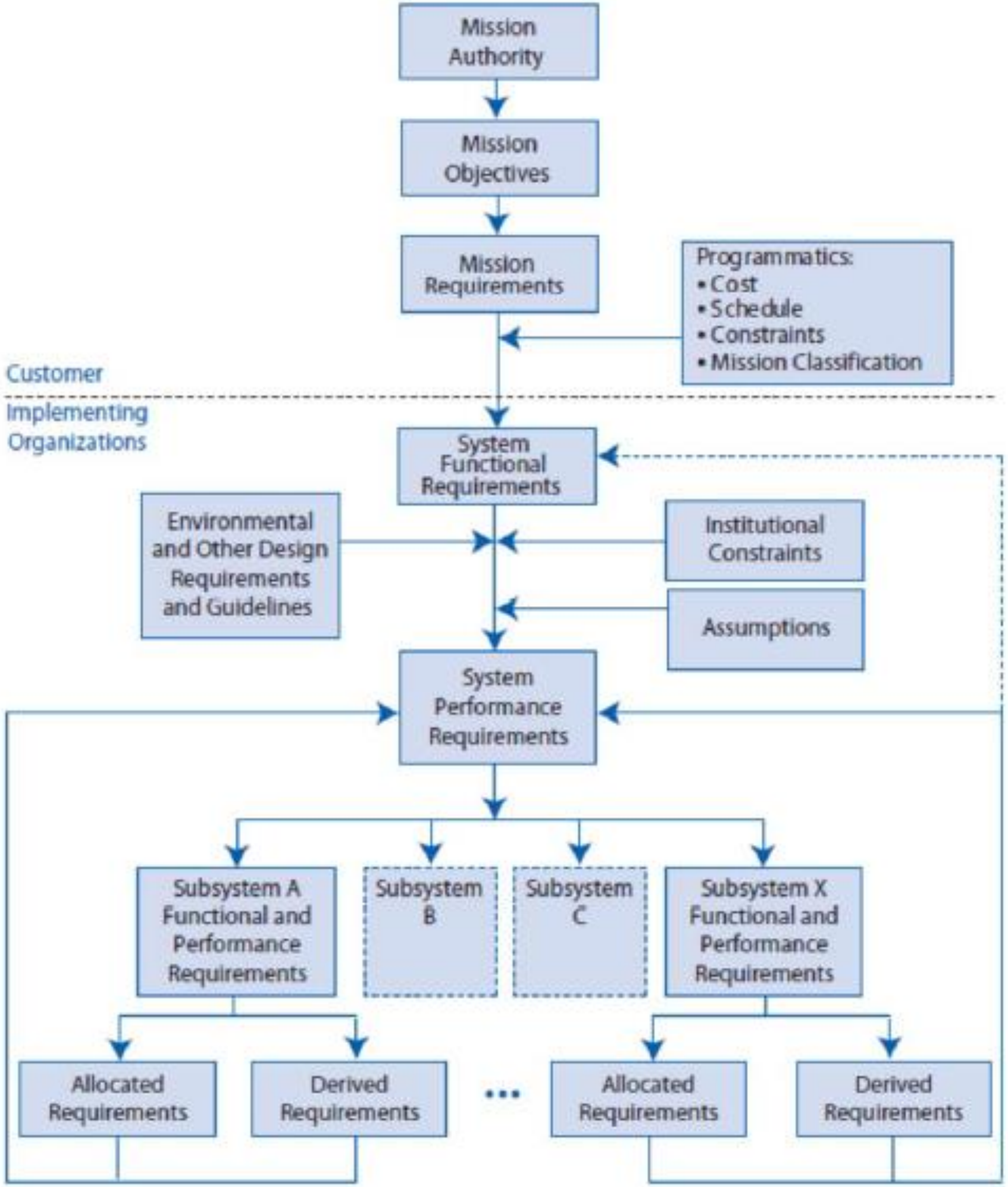
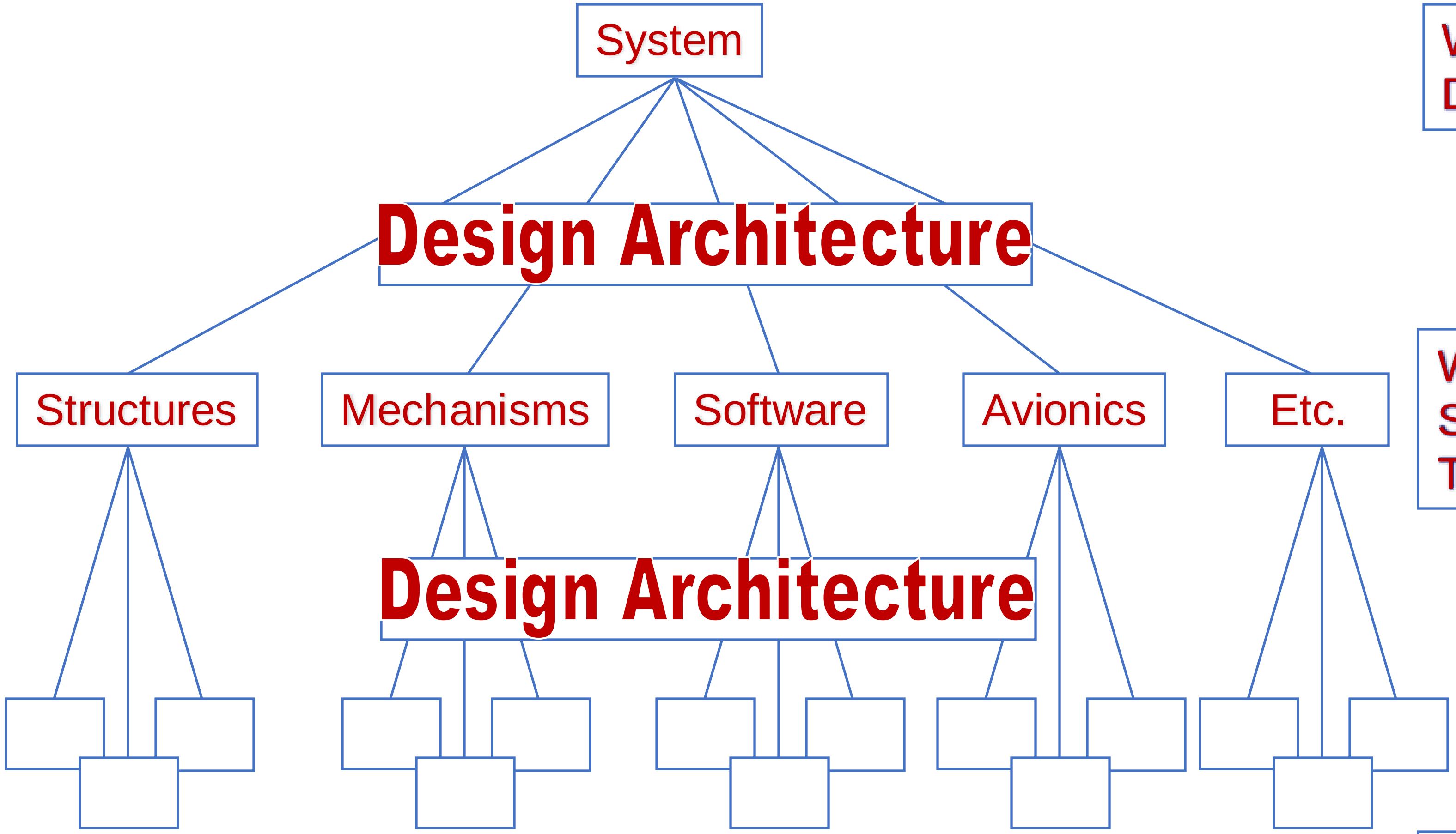

Mission Authority
Mission Objectives
Mission Requirements
Programmatics:
• Cost
• Schedule
• Constraints
• Mission Classification
Customer
Implementing Organizations
System Functional Requirements
Environmental and Other Design Requirements and Guidelines
Institutional Constraints
Assumptions
System Performance Requirements
Subsystem A Functional and Performance Requirements
Subsystem B
Subsystem C
Subsystem X Functional and Performance Requirements
Allocated Requirements
Derived Requirements
...
Allocated Requirements
Derived Requirements

Figure 4.2-4 The Flowdown of Requirements

REQUIREMENTS FLOWDOWN: ONE TO MANY



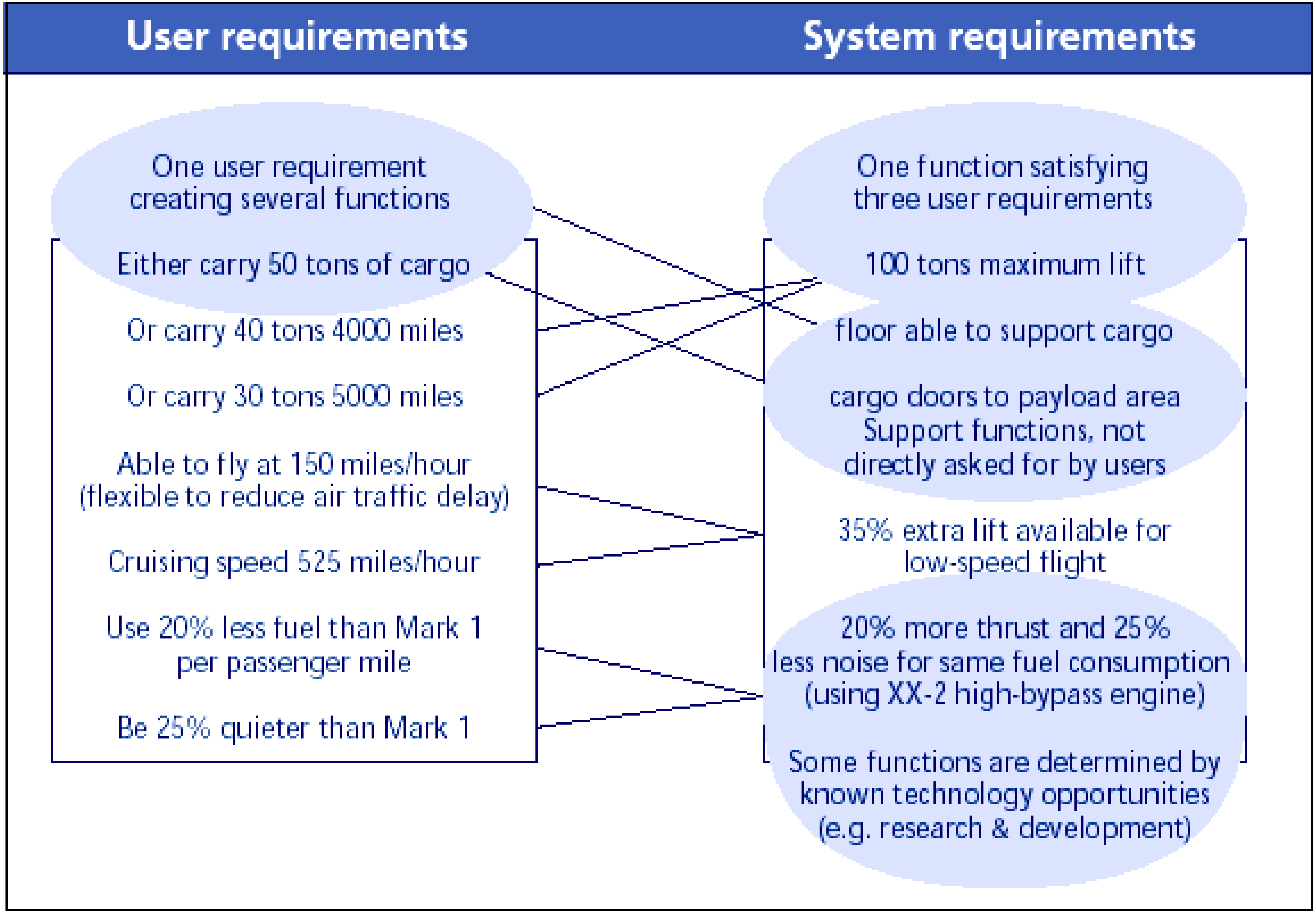
What The System Shall Do

What The Subsystems Shall Do For The The System

What The Components Shall Do For The The Subsystem

TRACEABILITY

- ‘Traceability is the relationship of requirements up & down the requirements hierarchy.
    - Typically this is Parent to Child (and the reverse) but can also be Peer to Peer.
    - Helps to confirm complete and accurate allocation of requirements
        - **Do all Parent Requirements have Children?**
        - **Do all Children Requirements have Parents (or are they Orphans)?**
    - Helps identify ‘gold plating’, the addition of un-needed requirements.
    - Vitale to **assessing impact of changes** in requirements.
    - *Obviously, a Requirements Management Tool (database) is very helpful.*
        - *But we’ve used Traceability Tables in Word documents as well.*



User requirements

One user requirement creating several functions

Either carry 50 tons of cargo

Or carry 40 tons 4000 miles

Or carry 30 tons 5000 miles

Able to fly at 150 miles/hour
(flexible to reduce air traffic delay)

Cruising speed 525 miles/hour

Use 20% less fuel than Mark 1
per passenger mile

Be 25% quieter than Mark 1

System requirements

One function satisfying three user requirements

100 tons maximum lift

floor able to support cargo

cargo doors to payload area

Support functions, not directly asked for by users

35% extra lift available for
low-speed flight

20% more thrust and 25%
less noise for same fuel consumption
(using XX-2 high-bypass engine)

Some functions are determined by known technology opportunities (e.g. research & development)

RATIONALE

- Provides key information on how/why/when a requirement was developed or derived.
  - Can explain **why a requirement is needed** (including linking back to a User Need statement).
  - **What assumptions** the requirement may be based on.
  - **What trade study or conceptual design or architecture drove** the requirement.
  - Documenting this information in a rationale **statement helps maintain ‘corporate knowledge’ over time and personnel changes**.
    - MSRR Example
  - Can be used to make the requirement statement **easier to understand**… **but don’t use this as an excuse to write a poor Shall statement!**

RATIONALE INFORMATION

- **Reason for the Requirement**: **Often the reason for the requirement is not obvious**, and it **may be lost** if not recorded as the requirement is being documented. The reason may point to a constraint or concept of operations. If there is a clear parent requirement or trade study that explains the reason, then it should be referenced.
- **Document Assumptions**: If a requirement was written assuming the **completion of a technology development program or a successful technology mission**, the assumption should be documented.
- **Document Relationships**: The relationships with the **product's expected operations** (e.g., expectations about how stakeholders will use a product) should be documented. This may be done with a link to the ConOps.
- **Document Design Constraints**: Constraints imposed by the results from **decisions made as the design evolves** should be documented. If the requirement states a method of implementation, the rationale should state why the decision was made to limit the solution to this one method of implementation.

REQUIREMENT CONSTRUCTS

STANDARD CONSTRUCTS - SHALL

- It is a statement of imperative need and indicates that the requirement **must be verified**.
- One important word that most requirements use is the word "shall“. A standard construct for a requirement is therefore: *"The system(or element or equipment) shall ...".*
- The goal requirements are not imperatives, and sentences containing them do not contain the word "shall". An example of a goal requirement is: *"The goal is to provide a maximum range capability of 45 Kilometers."*

STANDARD CONSTRUCTS - WILL

- The word "will" is not (usually) used to signify a requirement.
- It may be used in a sentence containing a statement of fact such as *"vehicle tests will be conducted at government test facilities".*
- They may also be used to indicate that some events will take place in the future. *"Test instrumentation will be provided by the government in 1998."*

STANDARD CONSTRUCTS – IN ACCORDANCE / AS SPECIFIED

- When a requirement is defined by referencing another standard, use a construct such as "in accordance with" or "as specified in".
- An example requirement is *"System parts and equipment requiring identification shall be marked in accordance with MIL STD 130."*

STANDARD CONSTRUCTS - LIMITS

- Do not use "Minimum" and "Maximum" to state limits. Use "No less than" or "No greater than".
- This standard construct avoids the ambiguity associated with the limiting values. The requirement *"Gun alignment error in elevation shall be no greater than 1.5 milliradians"* is not vague because it is clear that a 1.5 milliradian error is permissible.

STANDARD CONSTRUCTS – WORDS TO AVOID

- Specific words that should be avoided because they are vague and general are
  - "flexible", "fault tolerant", "high fidelity", "adaptable", "rapid or fast", "adequate", "user friendly", "support", "maximize" and "minimize" (*"The system design shall be flexible and fault tolerant"* is an example).
  - Other words that should be avoided are "and/or", "etc." and "may".

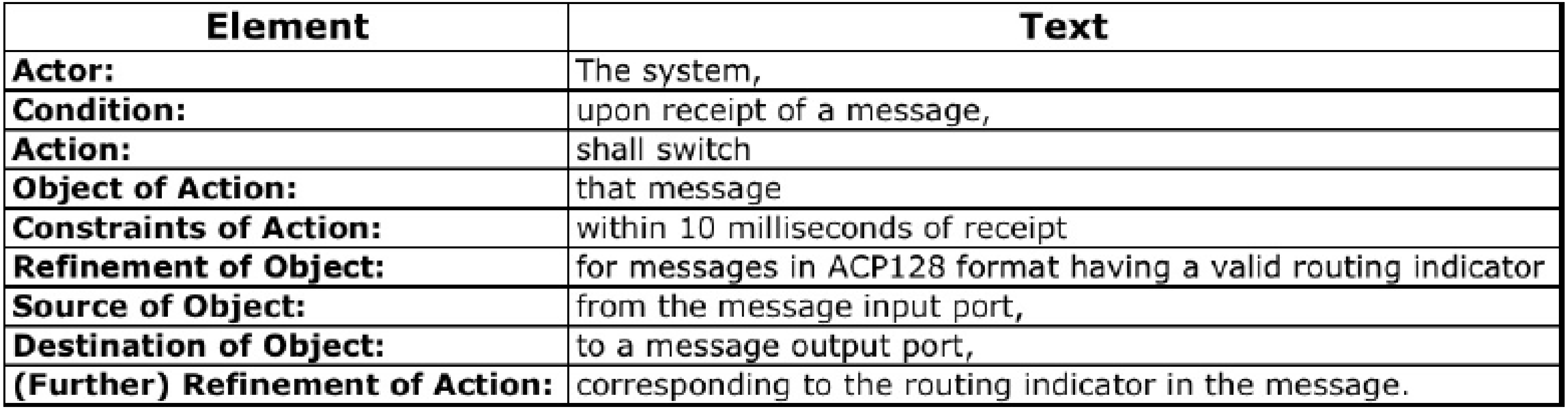
STRUCTURE MODEL EXAMPLE

- Requirements are most commonly expressed as natural language statements, although graphical and formal mathematical requirements languages are also used.
- For the natural language type of expression, requirements quality metrics may be developed through the parsing of each requirement statement into the elements of a structural model of a sound requirement, a template.

ELEMENTS

- **Actor**. This is the subject of the sentence - the thing being specified. Examples (good and bad!) are: “the system”, “the interface”, “the function”, ...
- **Conditions of Action**. This defines the conditions during which the action is to take place, for example “in Replay Mode”, and/or the initiating conditions: “upon receipt of a message”, “power having been applied”, ...
- **Action**. This is a verb - the action to be taken by the actor (subject). Examples are “shall calculate”, “shall display”, “shall fly”, “shall be displayed”.......
- **Constraints of Action**. These qualify the action, for example “at a resolution of 400 x 1000 pixels”, “within limits imposed by vehicle speed”, ..., and include performance
- **Object of Action**. This is a noun, and is the thing acted upon in taking the action. Examples are: “the message”, “the input signal”, ...
- **Refinement/Source of Object**. These qualify the object, for example (refinement): “of flash priority”, for example (source): “from DISCON”.
- **Refinement/Destination of Action**. These further qualify the action and may be additional to Constraints of Action. Examples are “in accordance with IEEE 802.11g”, “to DISCON”.

REQUIREMENT TEXT TEMPLATE



Element	Text
Actor:	The system,
Condition:	upon receipt of a message,
Action:	shall switch
Object of Action:	that message
Constraints of Action:	within 10 milliseconds of receipt
Refinement of Object:	for messages in ACP128 format having a valid routing indicator
Source of Object:	from the message input port,
Destination of Object:	to a message output port,
(Further) Refinement of Action:	corresponding to the routing indicator in the message.

METADATA

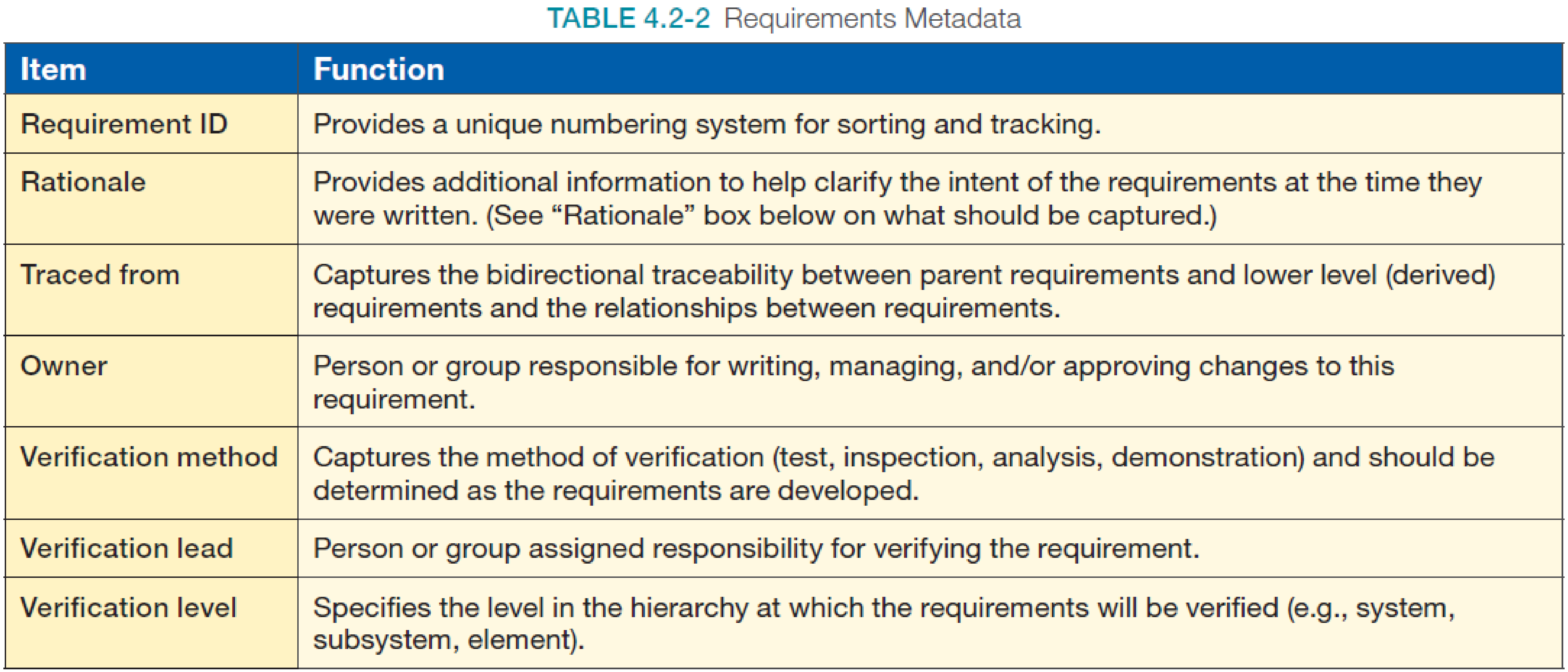
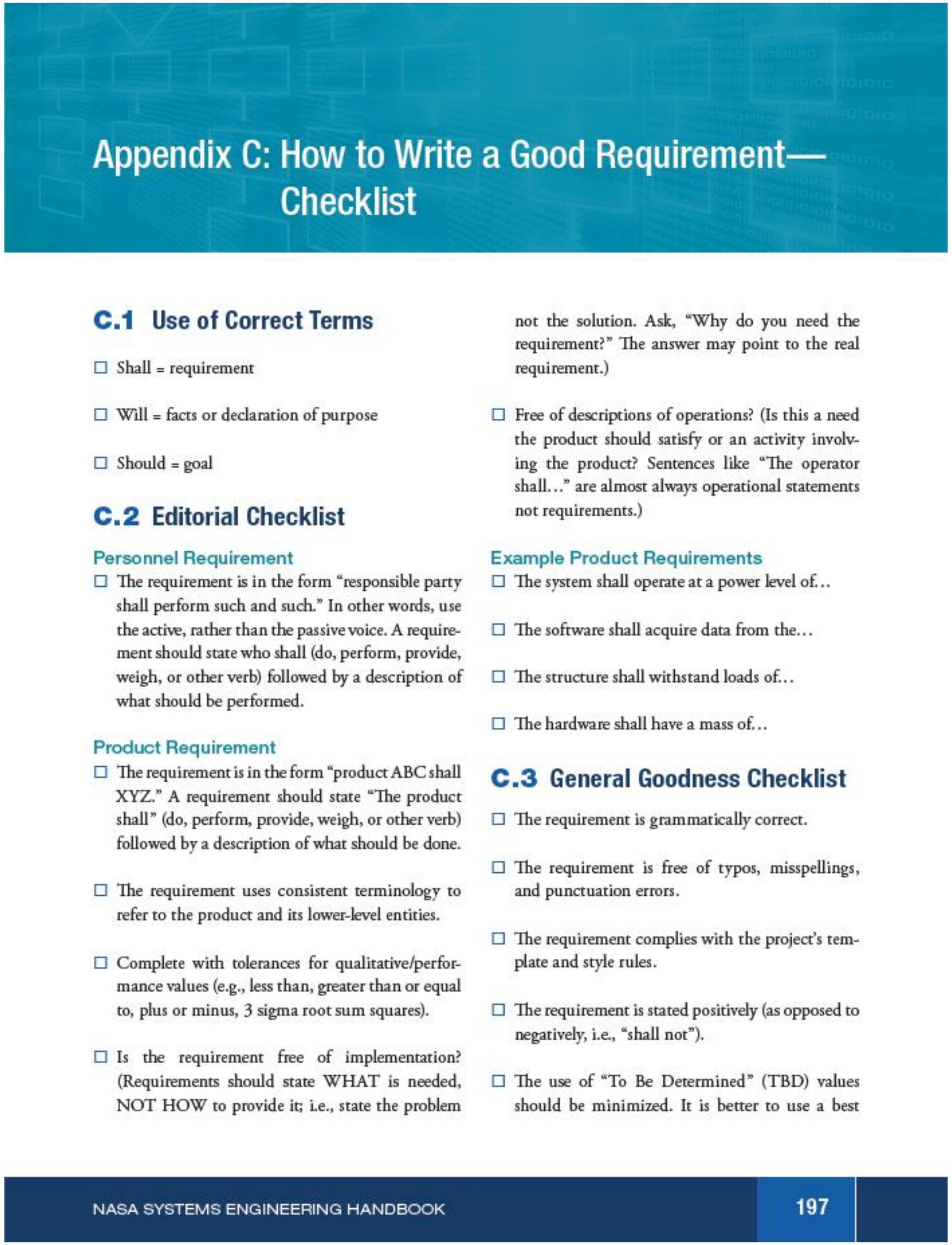


TABLE 4.2-2 Requirements Metadata

Item	Function
Requirement ID	Provides a unique numbering system for sorting and tracking.
Rationale	Provides additional information to help clarify the intent of the requirements at the time they were written. (See “Rationale” box below on what should be captured.)
Traced from	Captures the bidirectional traceability between parent requirements and lower level (derived) requirements and the relationships between requirements.
Owner	Person or group responsible for writing, managing, and/or approving changes to this requirement.
Verification method	Captures the method of verification (test, inspection, analysis, demonstration) and should be determined as the requirements are developed.
Verification lead	Person or group assigned responsibility for verifying the requirement.
Verification level	Specifies the level in the hierarchy at which the requirements will be verified (e.g., system, subsystem, element).

TOPIC ENDING

NASA SE HB



Appendix C: How to Write a Good Requirement—Checklist

C.1 Use of Correct Terms

- □ Shall = requirement
- □ Will = facts or declaration of purpose
- □ Should = goal

C.2 Editorial Checklist

Personnel Requirement

- □ The requirement is in the form "responsible party shall perform such and such." In other words, use the active, rather than the passive voice. A requirement should state who shall (do, perform, provide, weigh, or other verb) followed by a description of what should be performed.

Product Requirement

- □ The requirement is in the form "product ABC shall XYZ." A requirement should state "The product shall" (do, perform, provide, weigh, or other verb) followed by a description of what should be done.
- □ The requirement uses consistent terminology to refer to the product and its lower-level entities.
- □ Complete with tolerances for qualitative/performance values (e.g., less than, greater than or equal to, plus or minus, 3 sigma root sum squares).
- □ Is the requirement free of implementation? (Requirements should state WHAT is needed, NOT HOW to provide it; i.e., state the problem not the solution. Ask, "Why do you need the requirement?" The answer may point to the real requirement.)
- □ Free of descriptions of operations? (Is this a need the product should satisfy or an activity involving the product? Sentences like "The operator shall…" are almost always operational statements not requirements.)

Example Product Requirements

- □ The system shall operate at a power level of…
- □ The software shall acquire data from the…
- □ The structure shall withstand loads of…
- □ The hardware shall have a mass of…

C.3 General Goodness Checklist

- □ The requirement is grammatically correct.
- □ The requirement is free of typos, misspellings, and punctuation errors.
- □ The requirement complies with the project's template and style rules.
- □ The requirement is stated positively (as opposed to negatively, i.e., "shall not").
- □ The use of "To Be Determined" (TBD) values should be minimized. It is better to use a best

FINAL THOUGHTS

- Requirements are the way to **formalize the problem domain**.
- The **elicitation of requirements requires several approaches** to collect the correct needs
- The requirements must be as **closed as possible in early phases**
- At each level of the System the **requirement flows down** to inner requirements