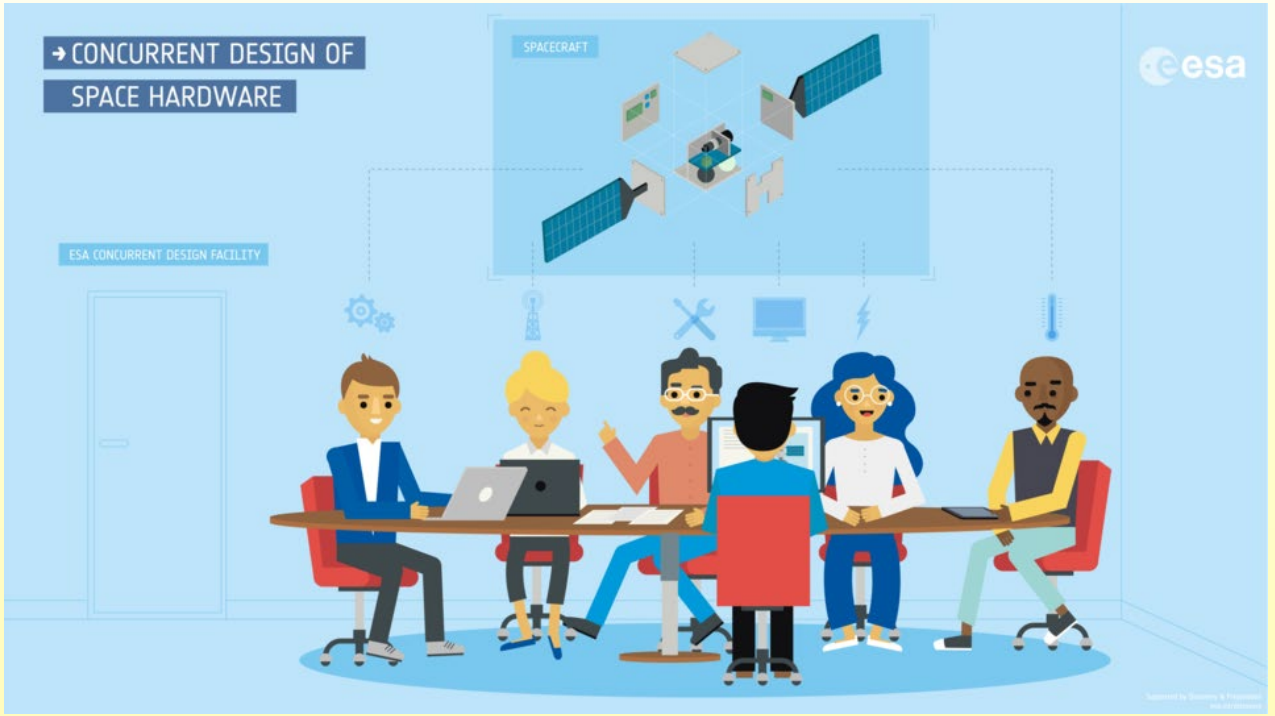
tsc|uniovi

EPI GIJÓN Universidad de Oviedo

# Requieremements

Germán León

gleon@uniovi.es



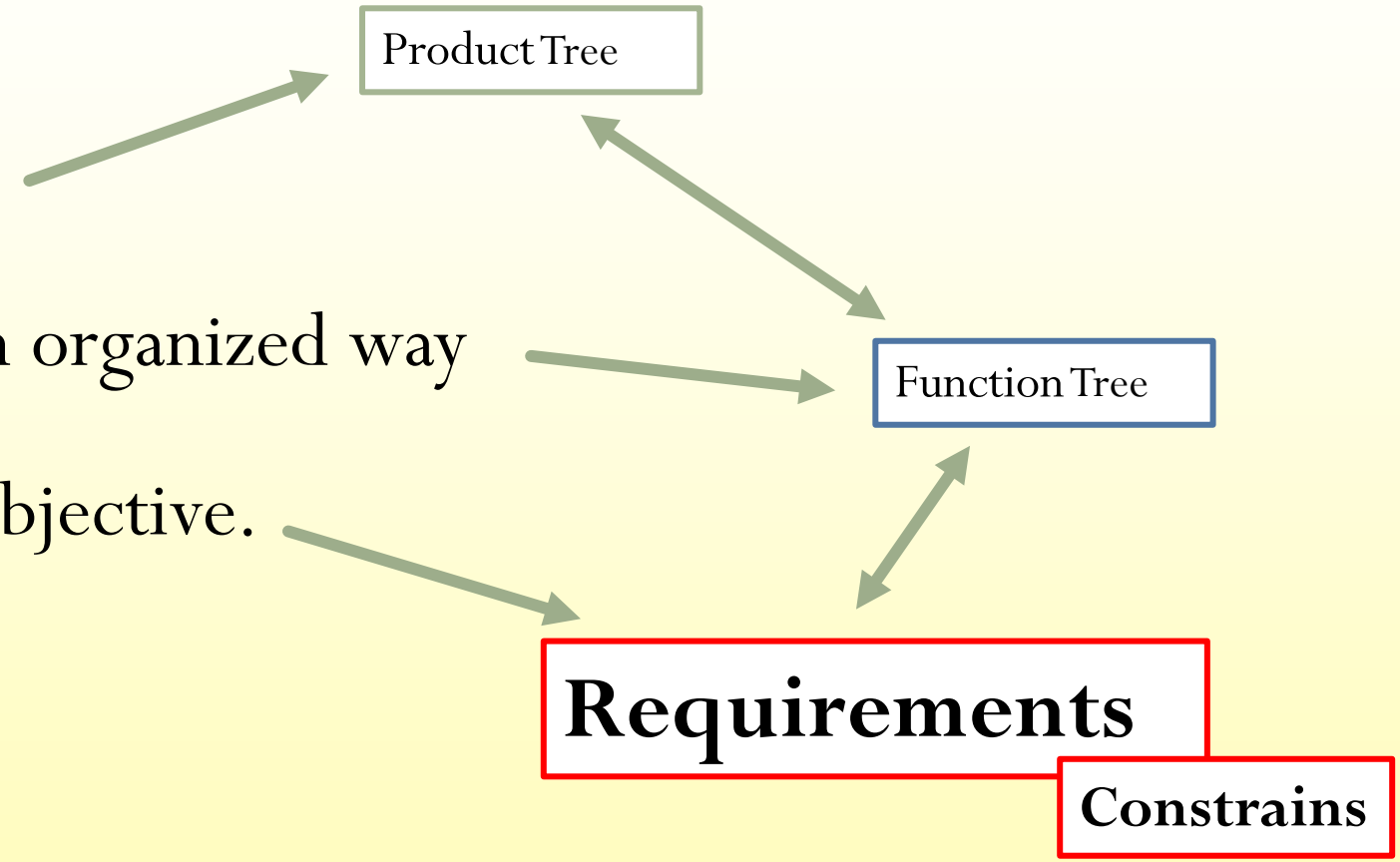

Grado en Ciencia y en Ingeniería de Datos

# A System is…

… a set of elements

which cooperate in an organized way

to achieve a defined objective.




**Exercise:**
https://www.linkedin.com/posts/fernando-fandi%C3%B1o-oliver_congratulations-engineers-robots-activity-6981942303103832064-2JbJ?utm_source=share&utm_medium=member_desktop

# Requirements definition is a means, not an end…

Dr. Dale Thomas (former ISS Systems Engineering and Integration Manager) described the approach as:
*"All too often, system engineering preoccupies itself with requirements definition for a product. Requirements definition is a means, not an end. … system engineering includes the* ***development of a valid*** *and cogent set of* ***requirements*** *and* ***the verification*** *of the as built* ***design against those requirements****.*

**Verification Process:** (all hardware and software meet)

1. Clearly identify all requirements.
2. Define the requirement's closure strategy—verify the requirements are met via inspection, analysis, demonstration or test.
3. Execute the necessary verification activities
4. Develop verification reports/analysis
5. Document closure

# RFLP **(Requirement, Functional, Logical, Physical) general methodology**

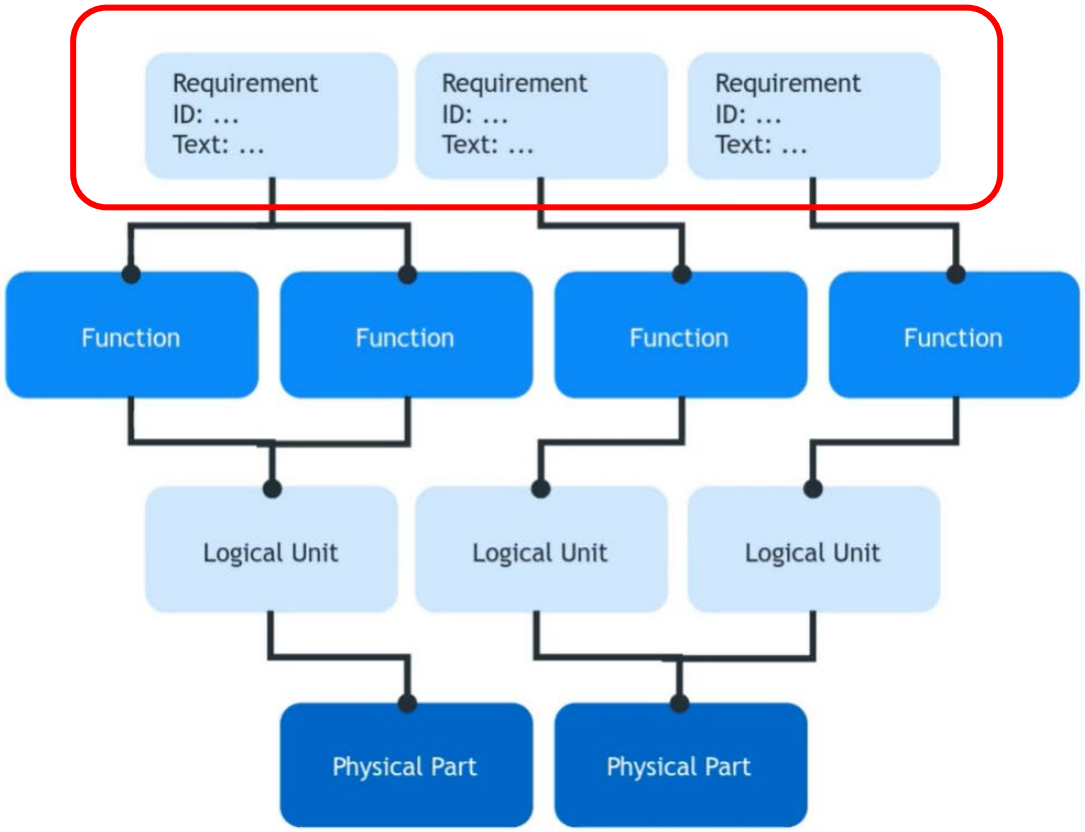
## MBSE Deep Dive: RFLP Approach

INTLAND SOFTWARE NTT DATA Trusted Global Innovator

**4 perspectives of the RFLP approach:**

- R: Requirements View
- F: Functional View
- L: Logical View
- P: Physical View

Different abstractions levels for system requirements



# Requirements

Phase A

1. finalises the expression of the needs identified in Phase 0.
2. proposes **system solutions** (including identification of critical items and risks) to meet the customer needs.
3. supports the **Preliminary Requirement Review (PRR)** and ensures implementation of PRR actions.
4. finalises the validation of **the requirements against the expressed needs** together with the customer.

**Description:** Transforming the baseline stakeholder expectations into unique, quantitative, and measurable technical requirements expressed as "shall" statements that can be used for defining the design solution. This includes analyzing the scope of the technical problems to be solved, defining constraints affecting the designs, defining the performance requirements, validating the resulting technical requirement statements, defining the Measures of Performance (MOPs).

Requirements need to be written down, validated and accessible to everyone on the team (document or database)

# Types of Requirements

Functional Requirements vs Non-functional requirements (constraints)

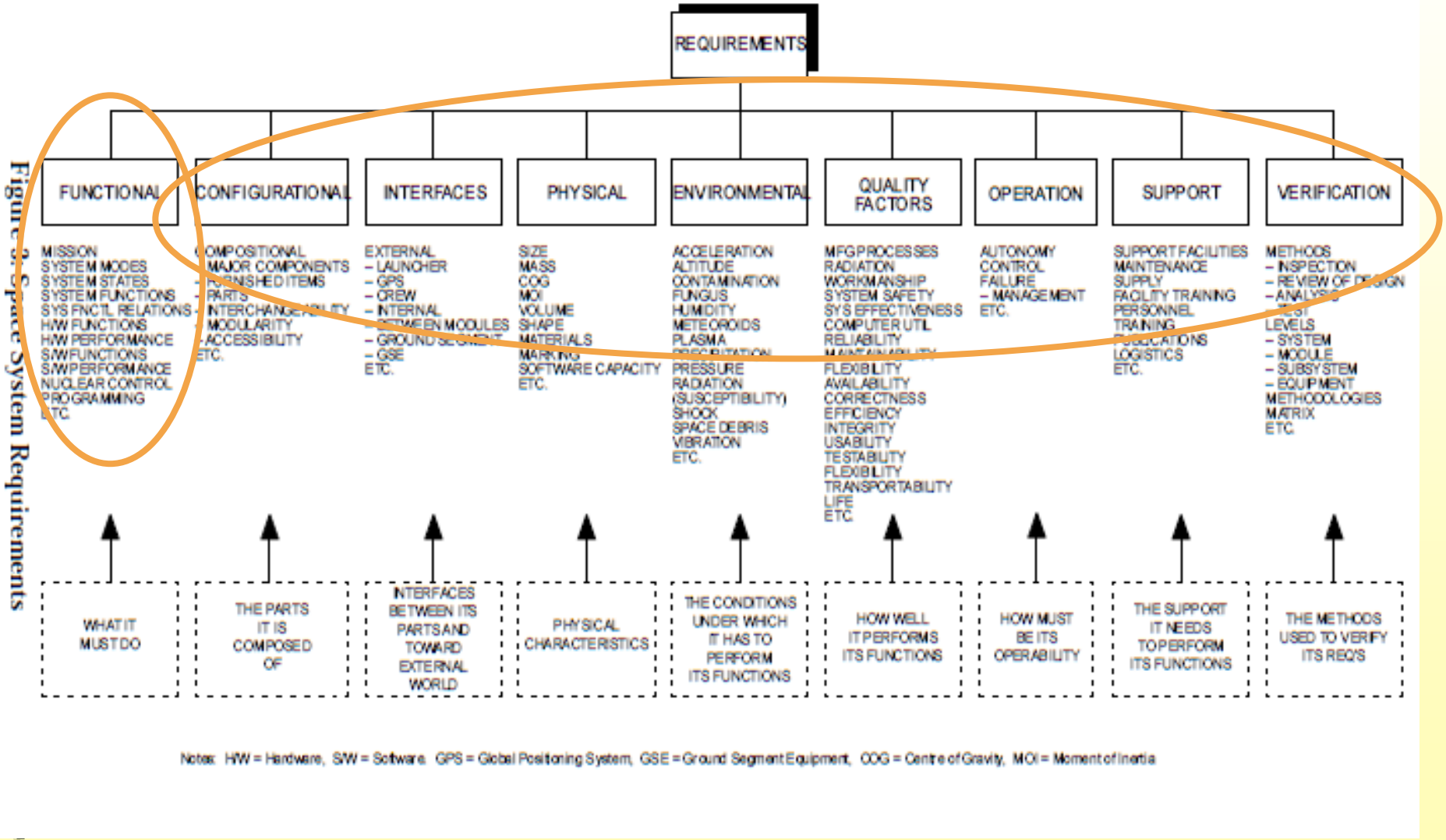
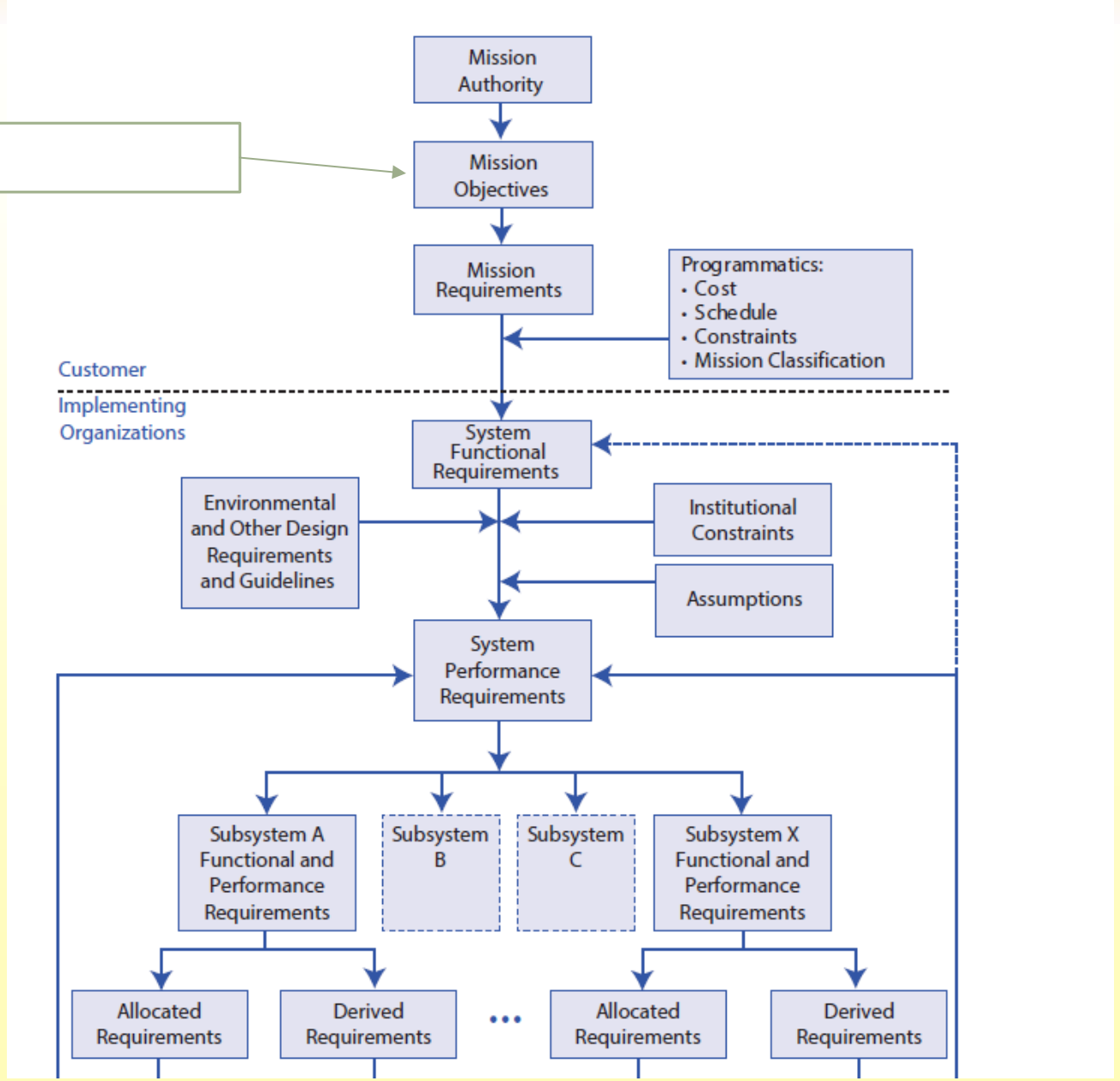



Figure 3: Space System Requirements

# Flowdown of Requirements

Diseñar e implementar un prototipo demostrador …



# Requirements vs Solutions

## Phase A: Requirement

| Req ID | Requirement Text | Rationale | Parent Req | Child Req | Owner | Test |
|---|---|---|---|---|---|---|
| Rob-Mis-10 | bla, bla | | N/A | | | |
| Rob-Fun-20 | The robot shall know its position outdoor | Linked to objetive X | N/A | Sys-Fun-30 | Localization Team | Phase C: SS outdoor func. test / Phase D: System (rover) outdoor test |

## Phase B: trade-off

Possible solutions analysis

Others Requirements / driven design

| | Pros | Cons |
|---|---|---|
| GNSS (GPS, Galileo…) | Low cost. A lot of experience | Accuracy 0.5 m |
| 4G/5G Cellular Networks | High accurary | New development. Need SIM card |

## Phase C: Design

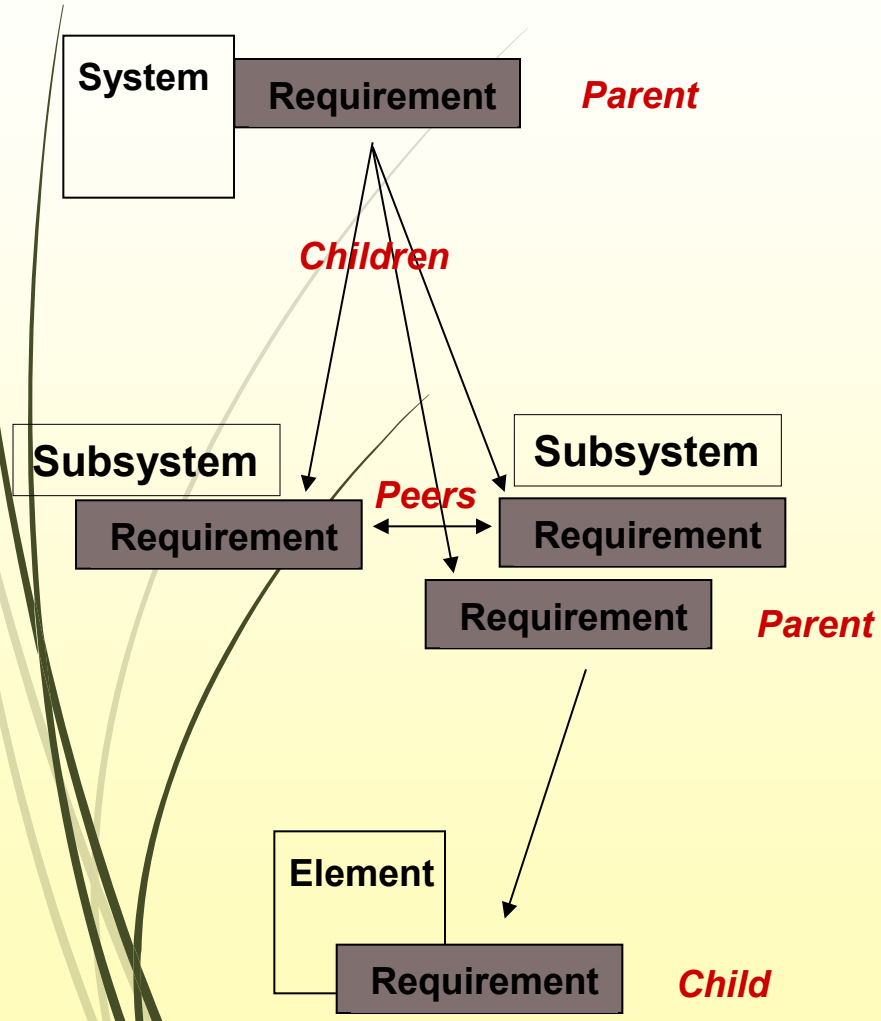
Component Selection

# 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). |

| Req ID | Requirement Text | Rationale | Parent Req | Child Req | Owner | Test |
|---|---|---|---|---|---|---|
| Rob-Mis-10 | bla, bla | | N/A | | | |
| Rob-Fun-20 | The robot shall know its position outdoor | Linked to objetive X | N/A | Sys-Fun-30 | Localization Team | Phase C: SS outdoor func. test / Phase D: System (rover) outdoor test |

# Requirements Families




| Req ID | Requirement Text | Rationale | Parent Req | Child Req | Owner | Test |
|---|---|---|---|---|---|---|
| Rob-Mis-10 | bla, bla | | N/A | | | |
| Rob-Fun-20 | The robot shall know its position outdoor | Linked to objetive X | N/A | Sys-Fun-30 | Localization Team | Phase C: SS outdoor func. test / Phase D: System (rover) outdoor test |

| Req ID | Requirement Text | Rationale | Parent Req | Child Req | Owner | Test |
|---|---|---|---|---|---|---|
| Sys-Fun-30 | The robot shall use GNSS to know its position | GNSS is easy to integrate and lower cost than others solutions | Rob-Fun-20 | Loc-Fun-10 | Localization Team | |
| Loc-Fun-10 | The robot shall measure its position each **TBD** seconds | | Sys-Fun-30 | | Localization Team | |

# SMART Requirements (tips)

Requirements need to be written down, verified, validated and accessible to everyone on the team (document or database)

**Good requirements are:**

– Short/Synthetic

– Definite/Unambiguous

– Verifiable

– Traceable

–Formulated using terms that have been properly defined earlier

-Prioriazable: must / desirable / nice to have

**Good set of requirements is:**

– Complete

– Coherent

– Structured

– Non-redundant

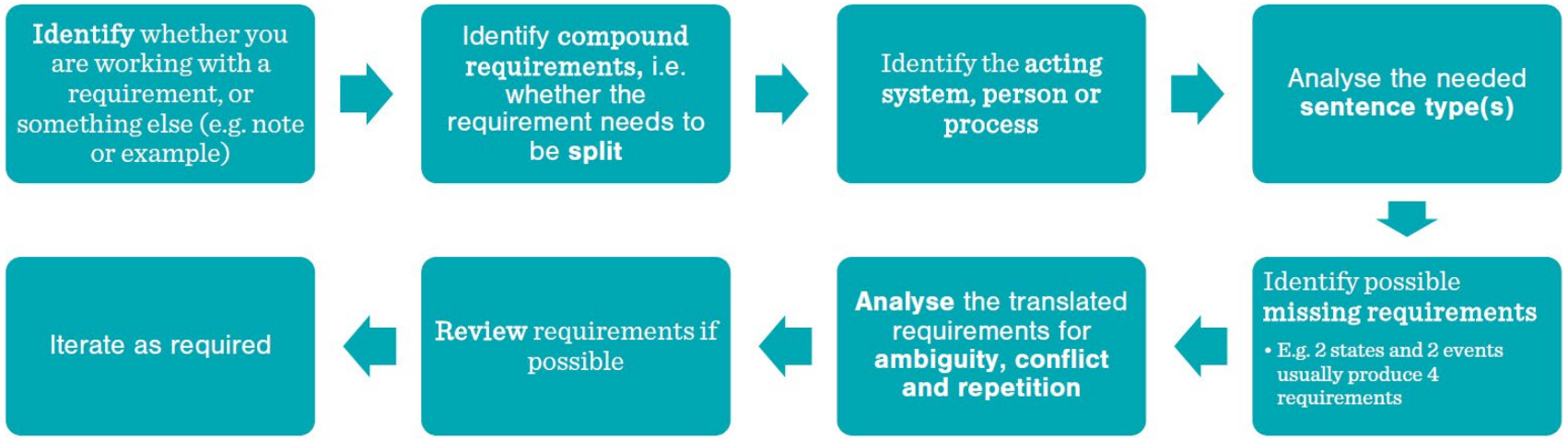
Aalto University

# EARS quick reference sheet

*Easy Approach to Requirements Syntax*

## Sentence types

| | |
|---|---|
| Ubiquitous | • The <system name> shall <system response><br>• *The kitchen system shall have an input hatch.* |
| Event-driven | • **When** <optional preconditions> <trigger>, the <system> shall <system response><br>• *When the chef inserts a potato to the input hatch, the kitchen system shall peel the potato.* |
| State-driven | • **While** <in a state>, the <system> shall <system response><br>• *While the kitchen system is in maintenance mode, the kitchen system shall reject all input.* |
| Unwanted behavior | • **If** *<optional preconditions> <trigger>*, ***then*** the <system> shall <system response><br>• *If a spoon is inserted to the input hatch, then the kitchen system shall eject the spoon.* |
| Optional | • **Where** <feature>, the <system> shall <system response><br>• *Where the kitchen system has a food freshness sensor, the kitchen system shall detect rotten foodstuffs.* |




## Some characteristics of a good requirement


**Avoid:**

- "support", "but not limited to", "etc", "and/or"
- Unverificable requirements (minimize, maximize, rapid, user-friendly, easy, sufficient, adequate, quick …)

# Hands-on Requirements (your project)

Return to your group and put your heads together and begin to (TBD min)…

- Develop System Requirements

See More…

Get Started with Requirements Toolbox - MathWorks España