



## › Webinar rules:

- › You'll be muted all along the Webinar
- › There's a *Questions* section to ask your questions whenever you want
- › All questions will be answered at the end of the webinar
- › If you have any technical issue please use the chatting box, or mail us at: [support@reusecompany.com](mailto:support@reusecompany.com)
- › The Webinar will be recorded. A link to the recording will be sent to you in few days

# Requirements Management: Managing data over entire life cycles



**Lars Anderberg**

Application Consultant  
The REUSE Company  
*[lars.anderberg@reusecompany.com](mailto:lars.anderberg@reusecompany.com)*



**Cecilia Karlsson**

Marketing & Communication  
The REUSE Company  
*[cecilia.karlsson@reusecompany.com](mailto:cecilia.karlsson@reusecompany.com)*



THE  
**REUSE**  
COMPANY



- Introduction to The REUSE Company and the presenter
- What is *Requirements Management*
- Life cycle
- Structure
- Traceability
- Attributes
- Draw the map
- Live demo
- Q&A





**The REUSE Company is a tool vendor specialized in the application of reuse methods, semantic technologies and artificial intelligence to improve the digitalization of the Systems Engineering lifecycle.**

**We promote lifecycle management methodologies guided by REUSE, based on a knowledge-centric approach, supporting the notion of authoritative source of truth, offering connectivity to everything, unlimited interoperability, and providing full support to technical management as in ISO 15288**



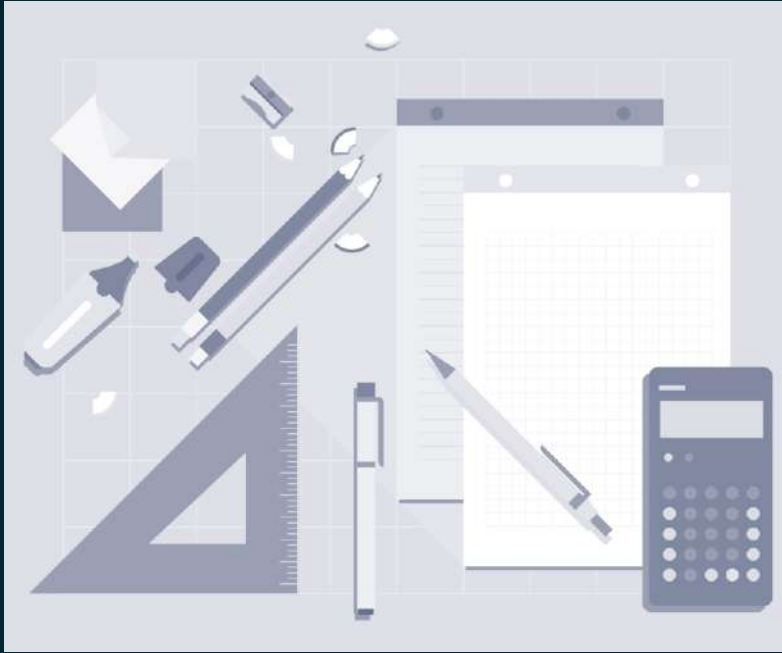
- RQA – QUALITY Studio
- RAT – AUTHORIZING Tool
- TRACEABILITY Studio
- V&V Studio
- KM – Knowledge Manager
- SES ENGINEERING Studio



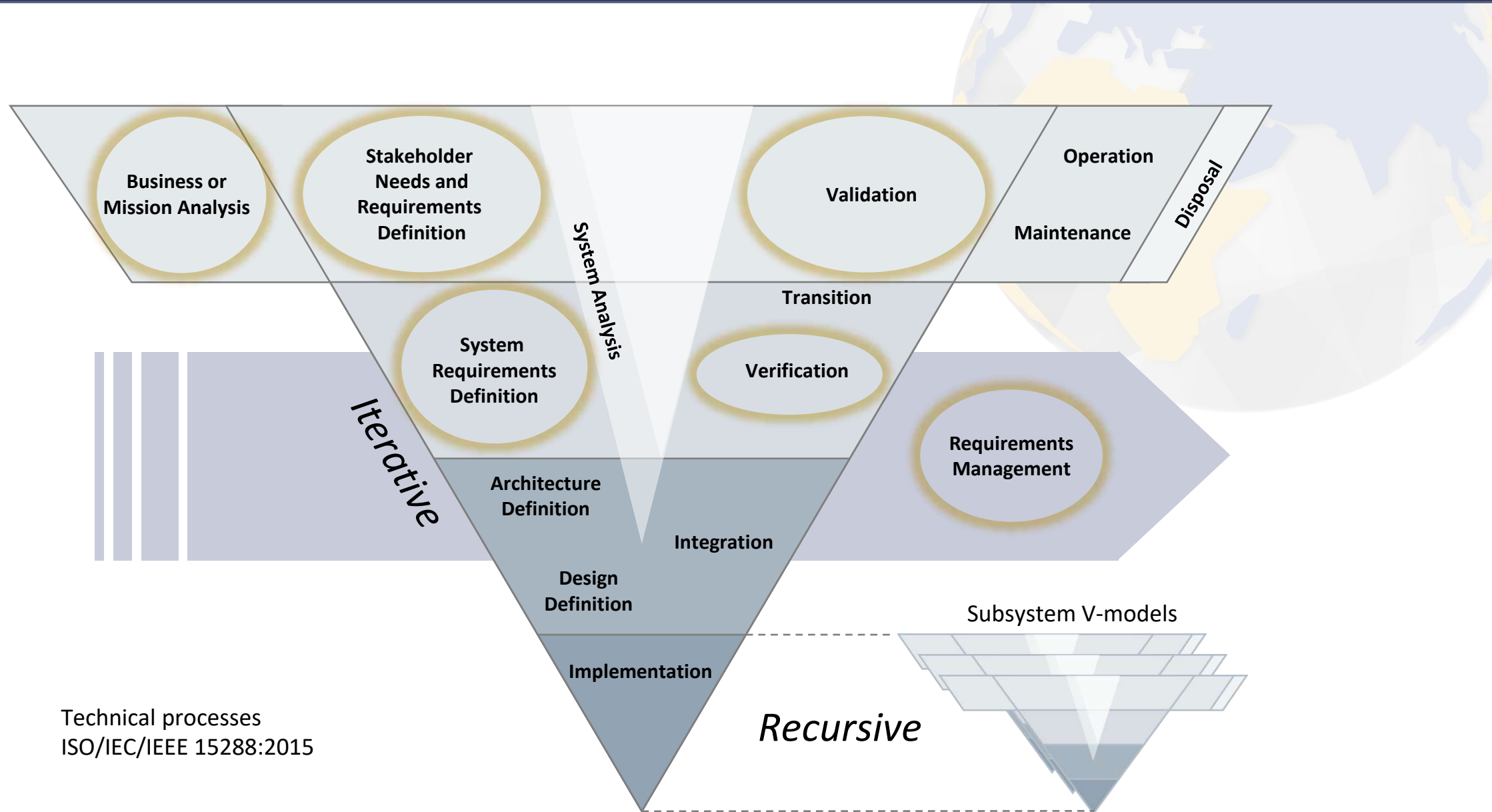
## Lars Anderberg



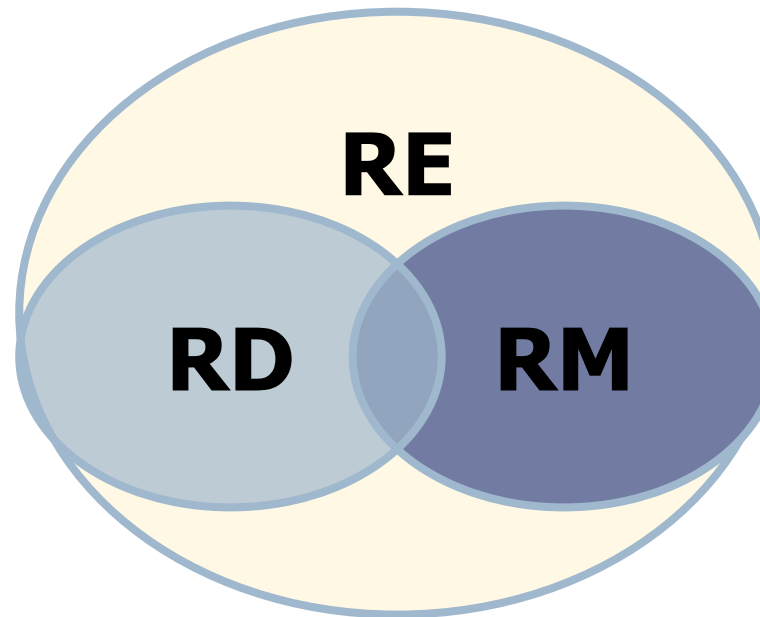
- **Current position:** Application consultant, The REUSE Company
- Requirements Engineering and Acquisition Support consultant for 20+ years
- INCOSE ESEP Certified



# Requirement's Management

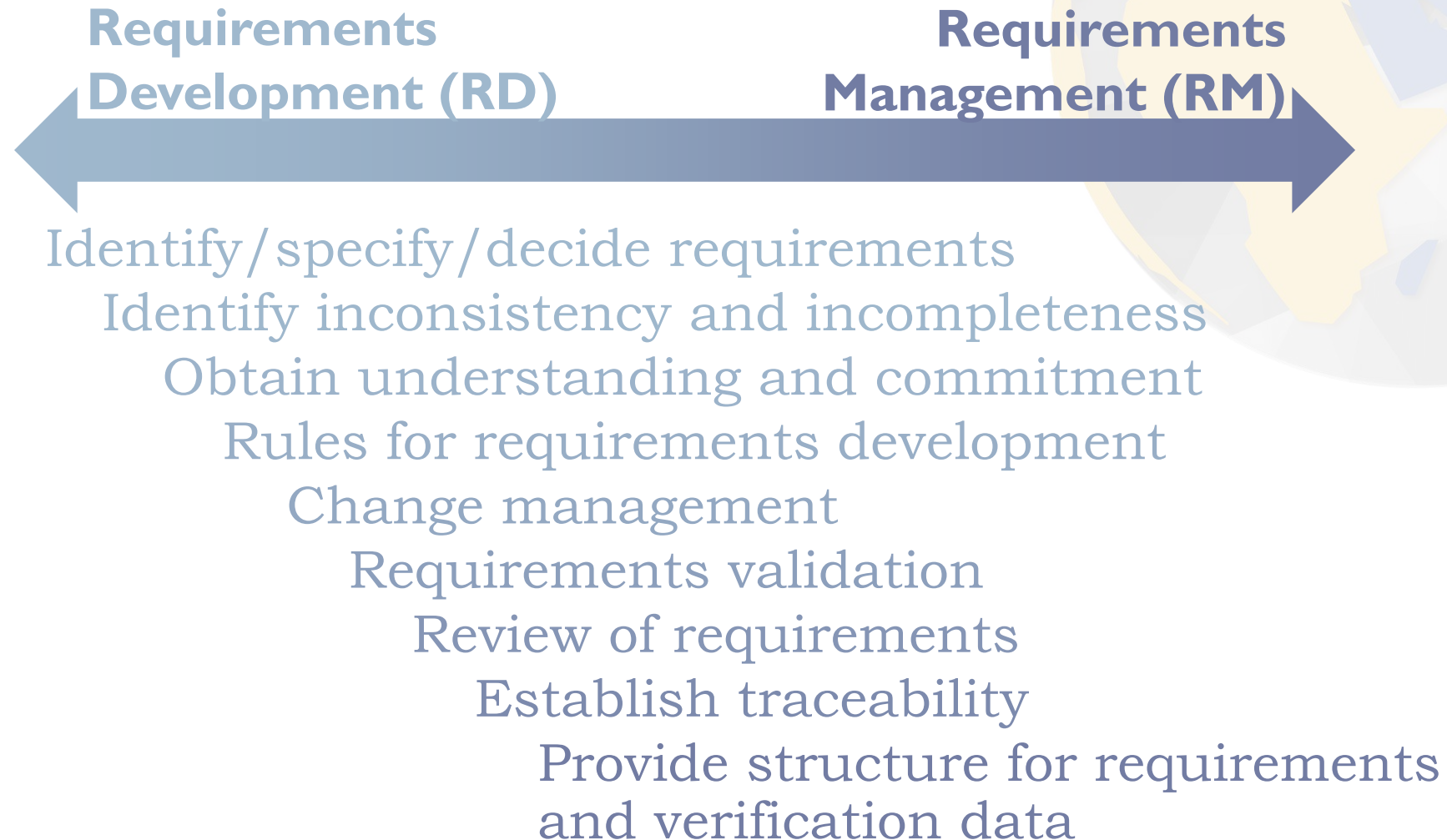






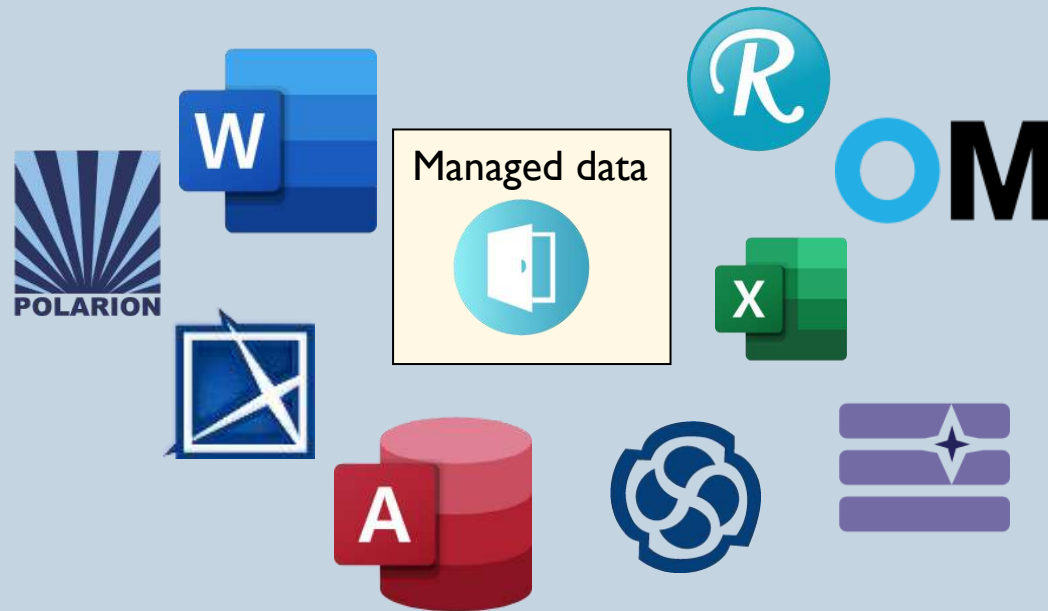
Requirements Development (RD)  
+ Requirements Management (RM)  
Requirements Engineering (RE)







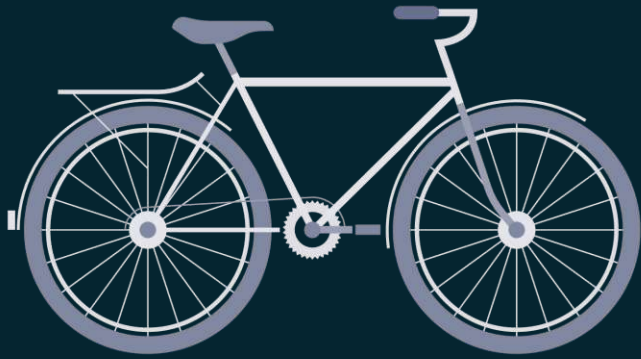
## All data in the project



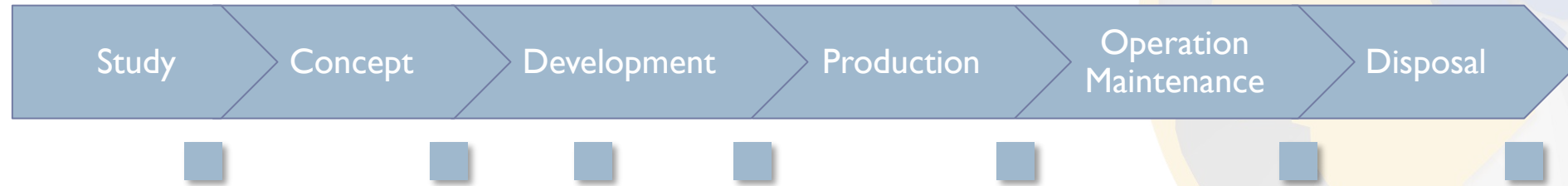
➤ ...common sense and hard work



- It's all about communication...
- ... between different people
- ... of different background
- ... over time



**Life cycles**

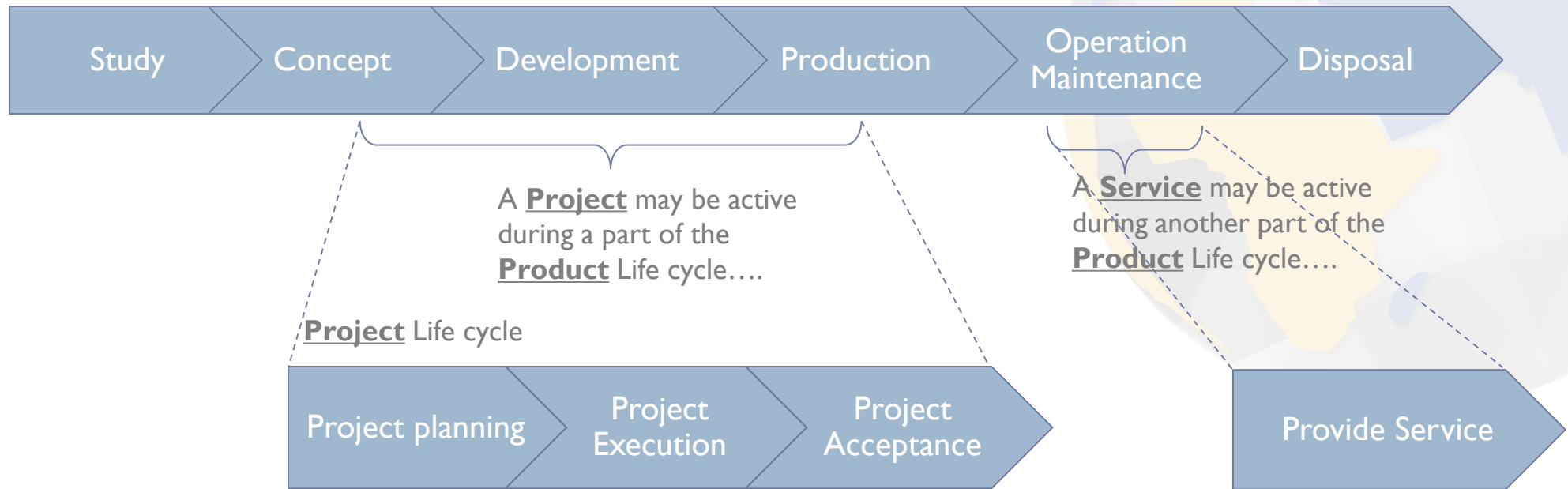


- A framework that helps us manage the system during its lifetime by defining
  - Phases
  - Milestones
  - Processes





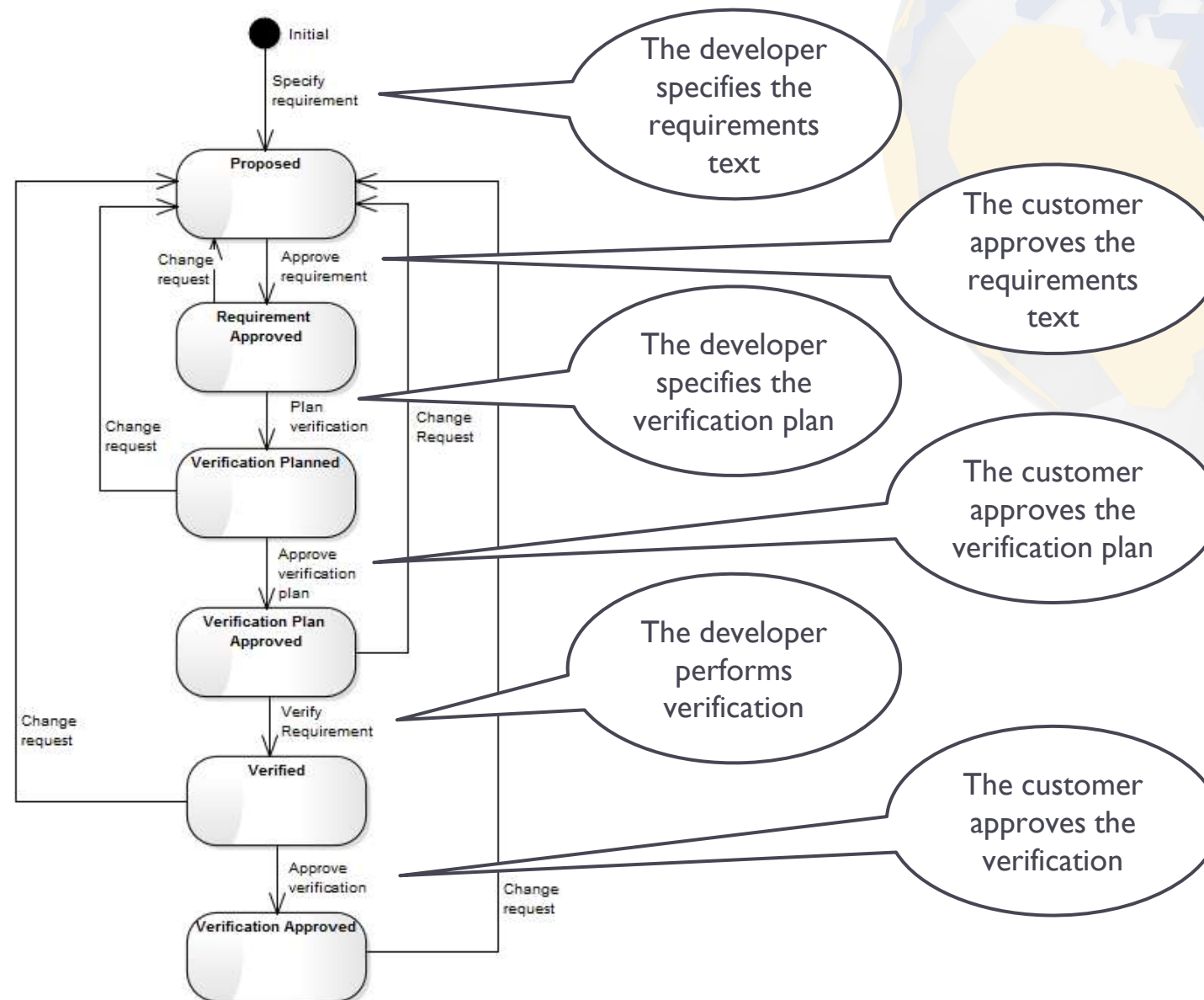
**Product** Life cycle



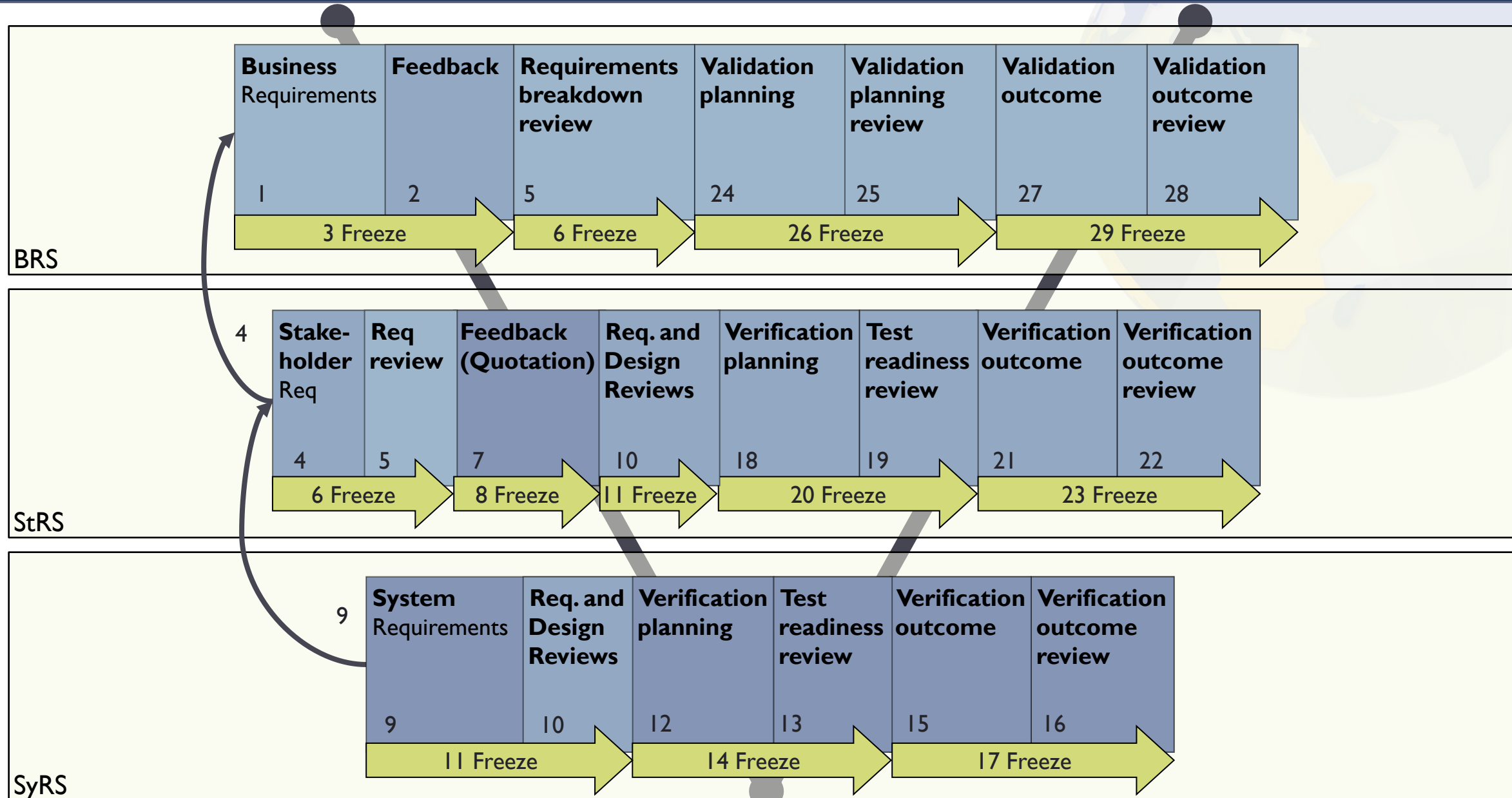
... i.e. a **Product requirement** ("It shall be blue") may have a completely different life cycle than a **Project requirement** ("The project shall produce a SEMP")  
Or a **Service requirement** ("The response time to a service request shall be less than 24 hours")

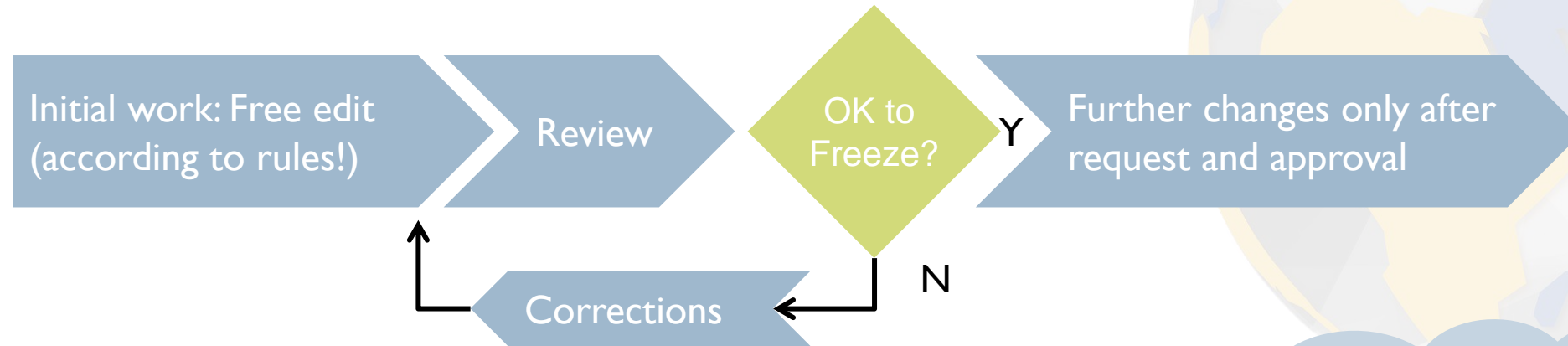
A Requirements Life Cycle identifies all data that is to be produced related to a (set of) Requirement(s) during the associated Product-, Project- or Service Life cycle along with rules for data use, ownership and freezing









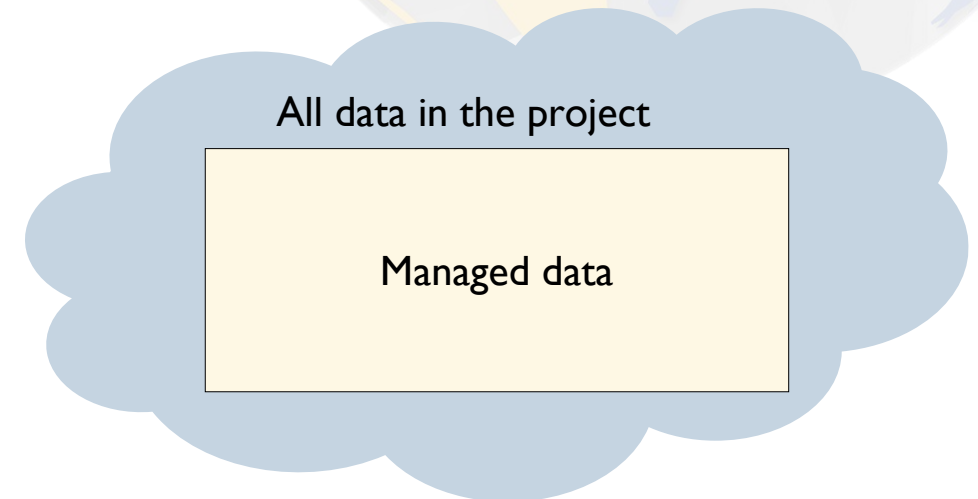


## › Configuration Management:

- › Decision to Freeze
- › Approval of Change Requests (CR) on frozen data

## › Requirements Management:

- › Enable Reviews
- › Keep track of frozen data (protect to avoid unapproved edit)
- › Enable Incremental Freeze
- › Impact analysis before change decisions (use links to track affected frozen data)
- › Implement approved changes



Remember:  
All managed data – not only requirements

## ➤ Work with data before vs after freeze

➤ Before: Free edit – but follow rules

➤ After: Issue change request

### ➤ Reasons for freeze

- Internal decisions and milestones
- External: Send data, receive data

### ➤ What shall be frozen

- Identified (sub-)set of Requirements, contextual information, definitions, ...
- Attributes, links

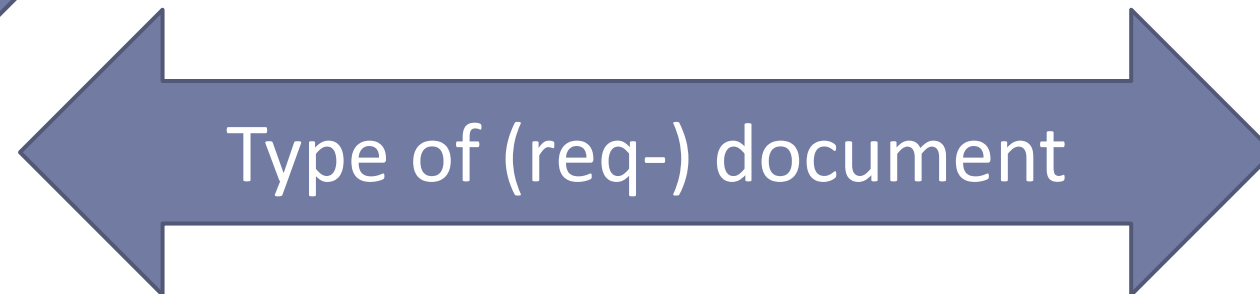
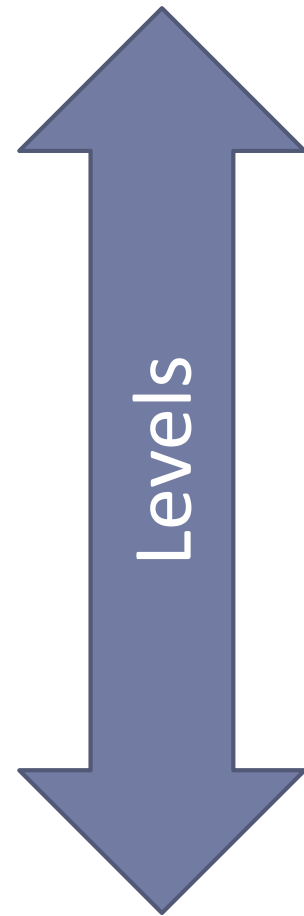
### ➤ What shall NOT be frozen

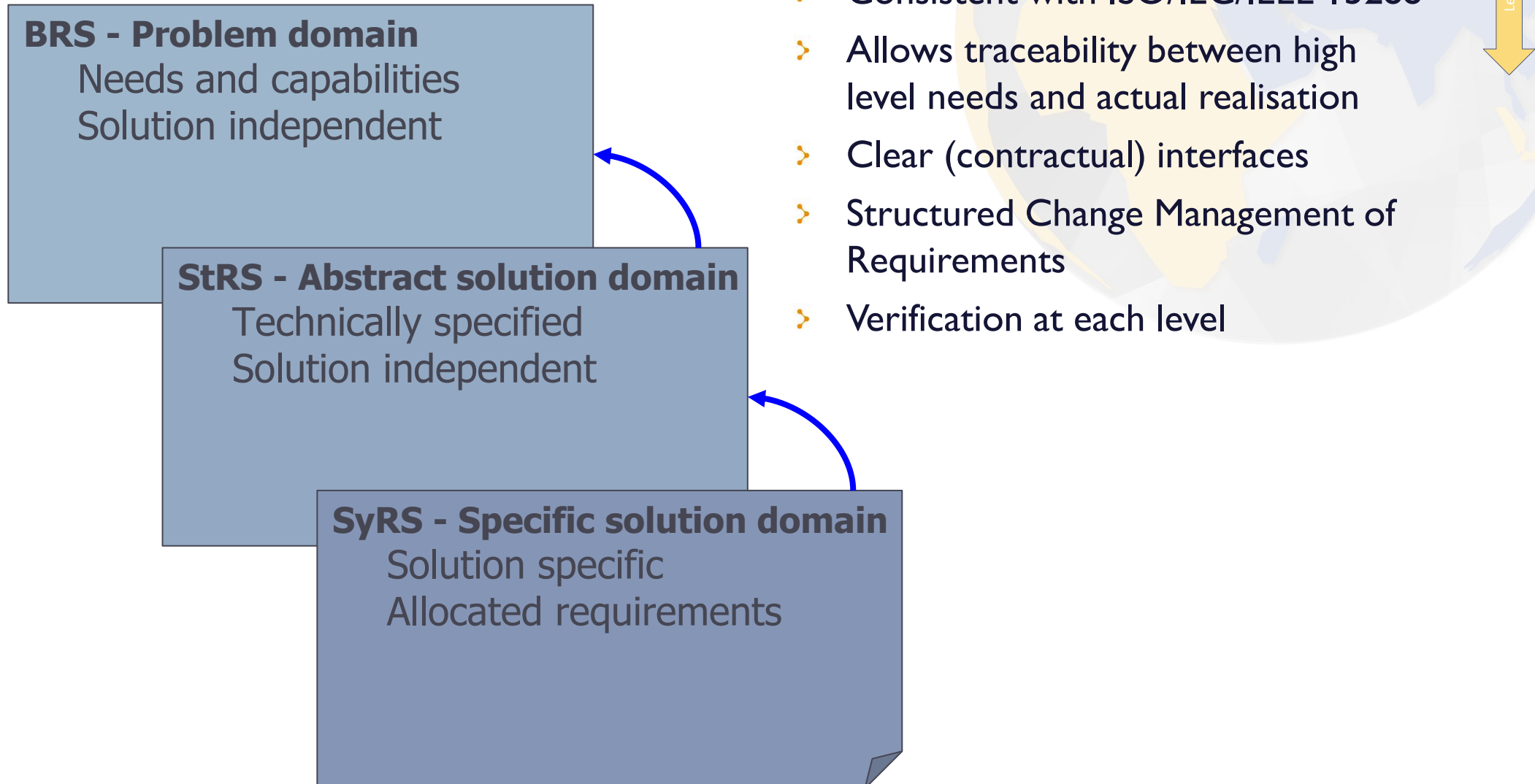
- Other identified (sub-)set of Requirements, contextual information, definitions, ...
- Notes, remarks, ownership (?)

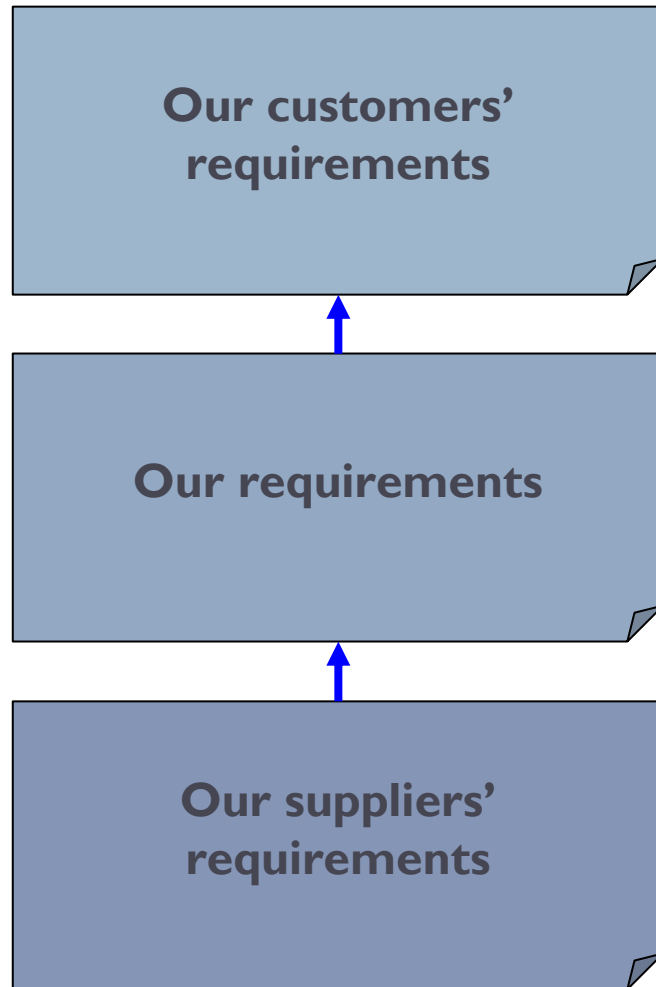




# Structure

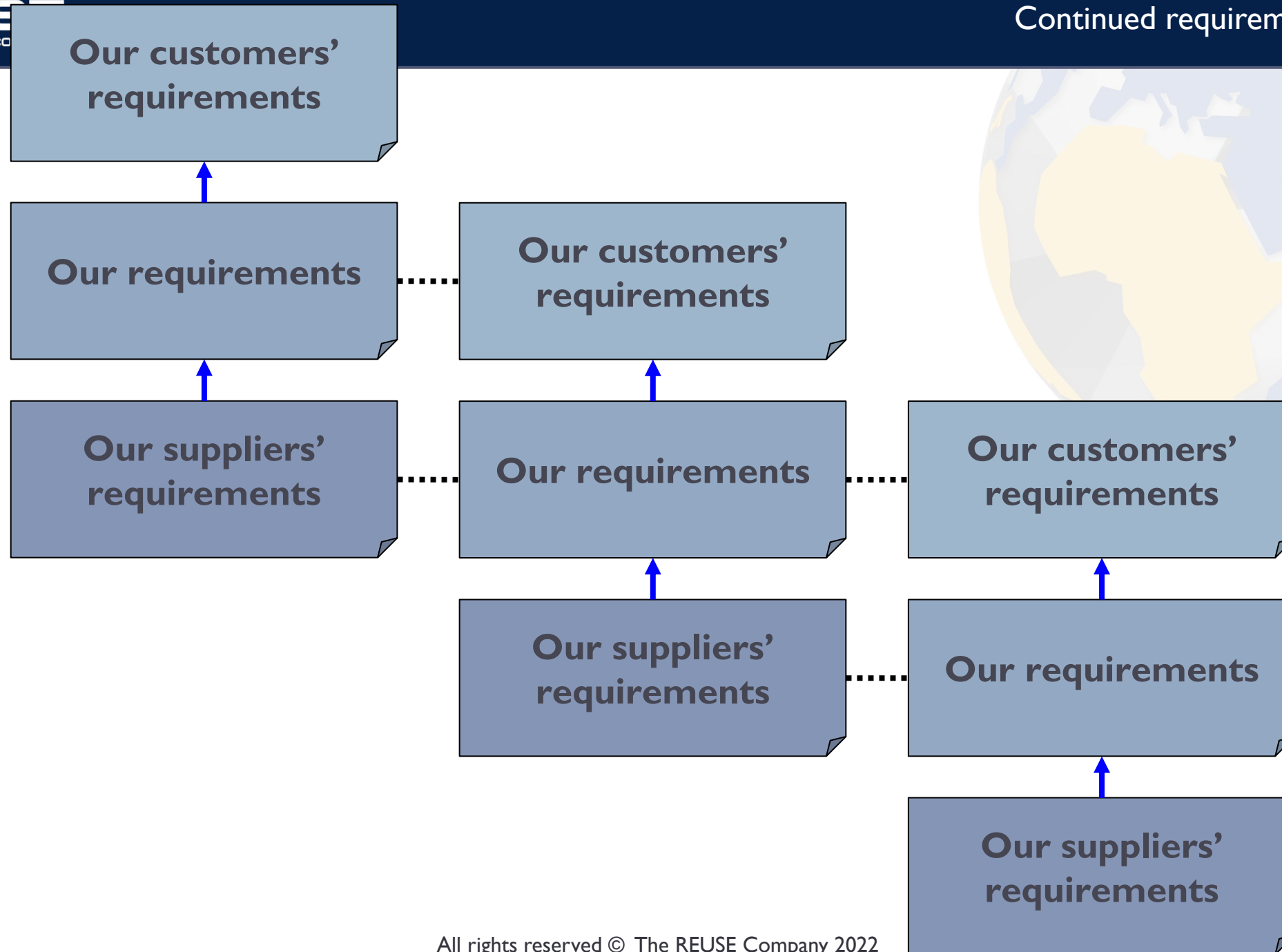




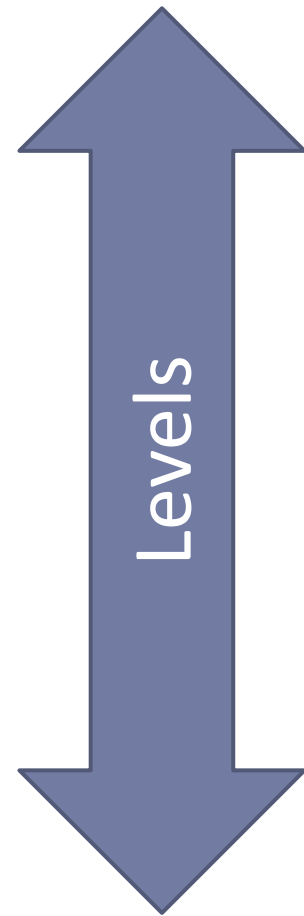


- Consistent with ISO/IEC/IEEE 15288
- Allows traceability between high level needs and actual realisation
- Clear (contractual) interfaces
- Structured Change Management of Requirements
- Verification at each level

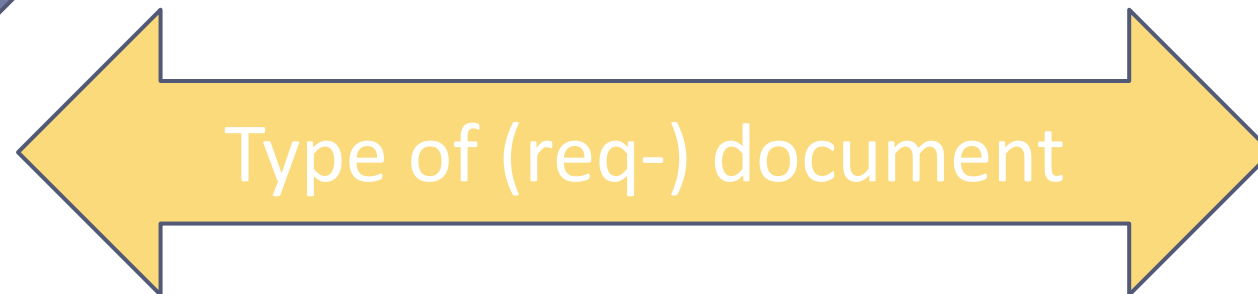








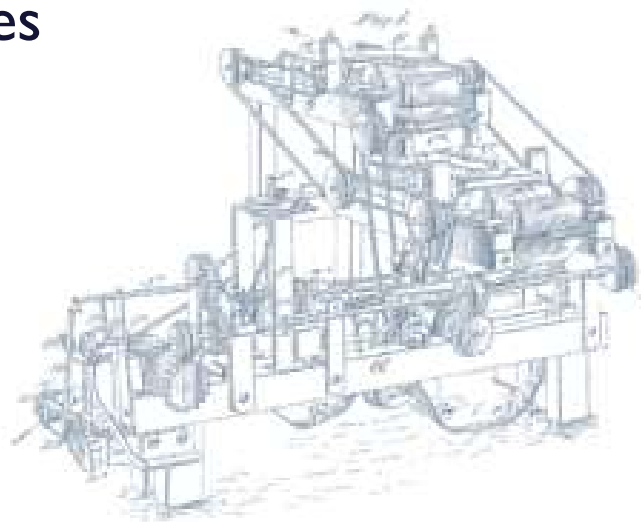
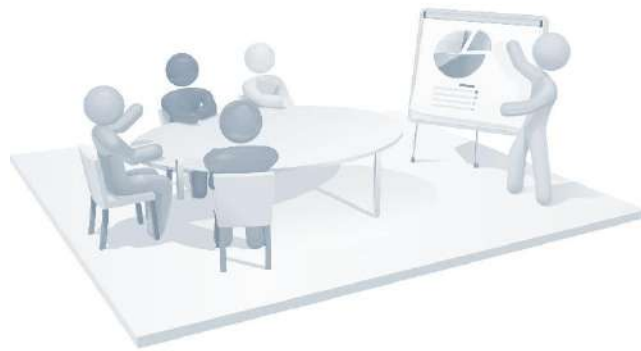
Levels

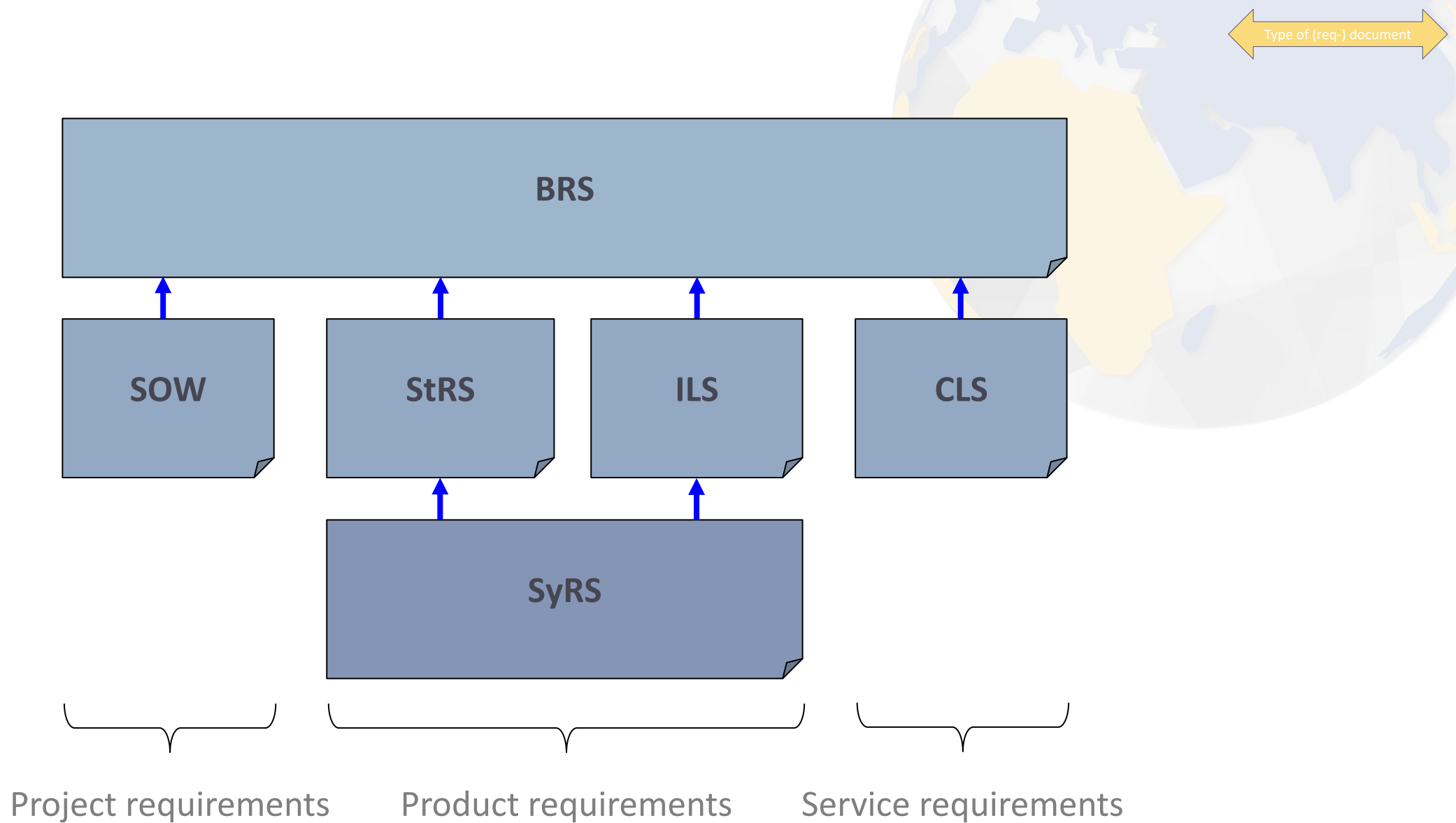


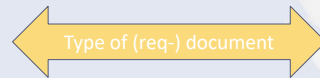
Type of (req-) document



- Main principle:
  - Separate different types of requirements (e.g. project requirements, product requirements, service requirements) in separate requirements documents
- Main reason:
  - Different life cycles
  - Different principles (and data need) for verification
  - May involve different parties







*"The SRR shall be planned and performed to cover the scope complying to the contract and the Contract Delivery Requirement List (CDRL) review documentation listing."*

SOW

*"The Temperature Warrior shall maintain the environment temperature within the defined range"*

StRS

ILS

*"Upon request, the Contractor shall perform preventive and corrective maintenance by using the agreed service network."*

CLS

*"While the Temperature Warrior is in Configuration Mode, the minimum temperature input parameter of the Temperature Warrior shall be 12 °C"*

SyRS

*"The height of the Temperature Warrior shall be inferior to 50 cm"*

*"When the activity of the Temperature Warrior is interrupted, the Temperature Warrior shall notify the administrator in less than 0.5 seconds"*

Project requirements

Product requirements

Service requirements



# Traceability



### 3.1.23

#### requirements traceability

identification and documentation of the derivation path (upward) and allocation/flow-down path (downward) of requirements (3.1.19) in the requirements set

#### Note 1 to entry:

One or more requirements upward from which a requirement is derived are called parent requirements.

A requirement downward that is derived from one or more parent requirements is called a child requirement.

*Source: ISO/IEC/IEEE 29148:2018(E)*



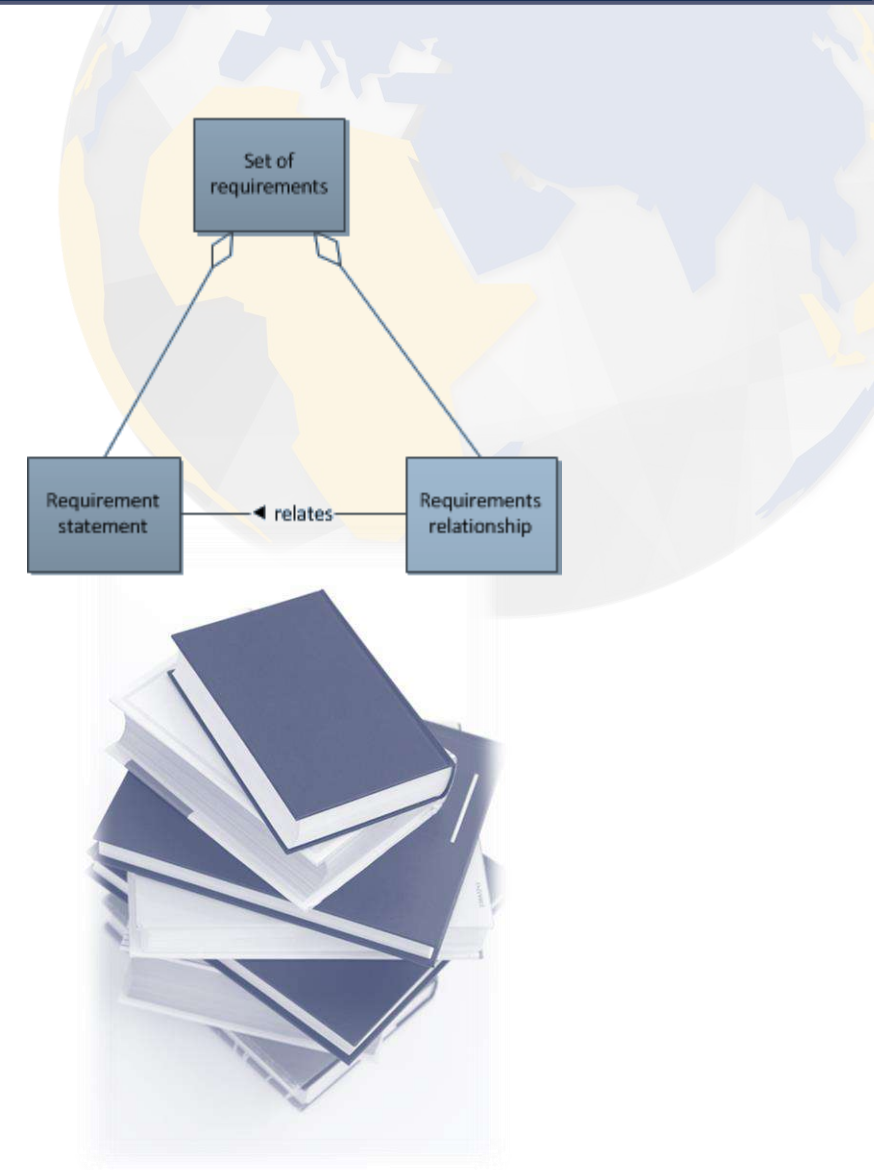
- Origin
  - What is the origin of this requirement?
- History
  - How has data evolved over time?
- References
  - Reference to decisions, plans and reports
- Realised by means of: (tool dependent)
  - Linking
  - Attributes
  - History features
  - Version handling
  - Other: ...





- **Change Control**
  - Requirements should be layered and traced in such a way that the impact of change can be analysed
- **Design Documentation**
  - Requirements should be layered and traced in such a way that the design satisfaction relationship between layers of requirements can be documented
- **Verification Documentation**
  - Requirements should be organized and traced in such a way that the relationship between requirements and verification artefacts can be documented

*INCOSE-TP-2010-006-01: Guide for Writing Requirements*

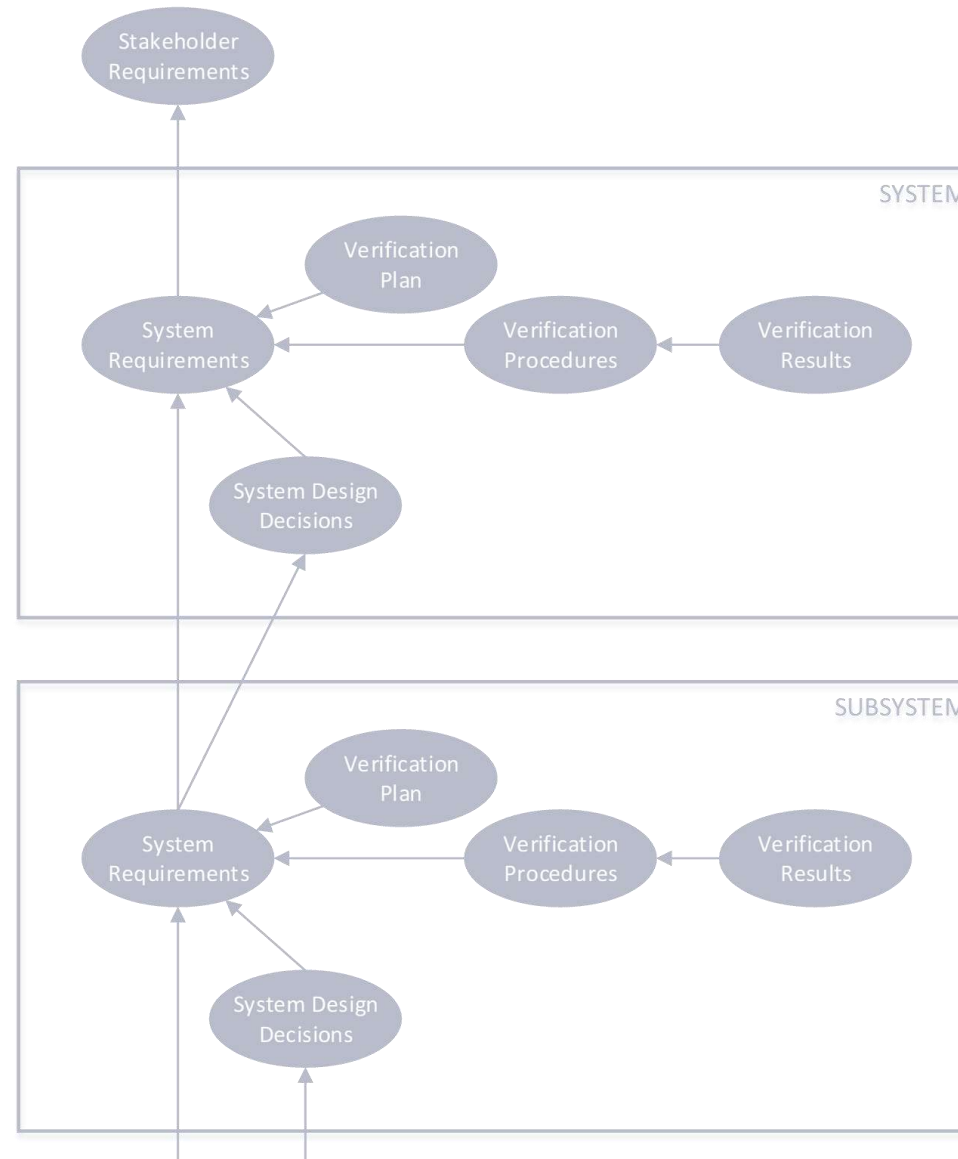






- Being able to understand the design and being able to “navigate” it
- Diminish the dependency on experts that “have the traceability in their head”
- Assess Completeness
  - Show that all parent requirements have been met by child requirements (i.e. have been taken care of in lower levels and design)
  - Show that all requirements have a matching verification procedure
  - Show that all requirements have been verified
- Enables automation, e.g. generating cross reference report, generating metrics for test procedure coverage
- Producing verification metrics for requirements (e.g. percentage of requirements that have a test procedure, percentage that have a certain verification result)







# Attributes



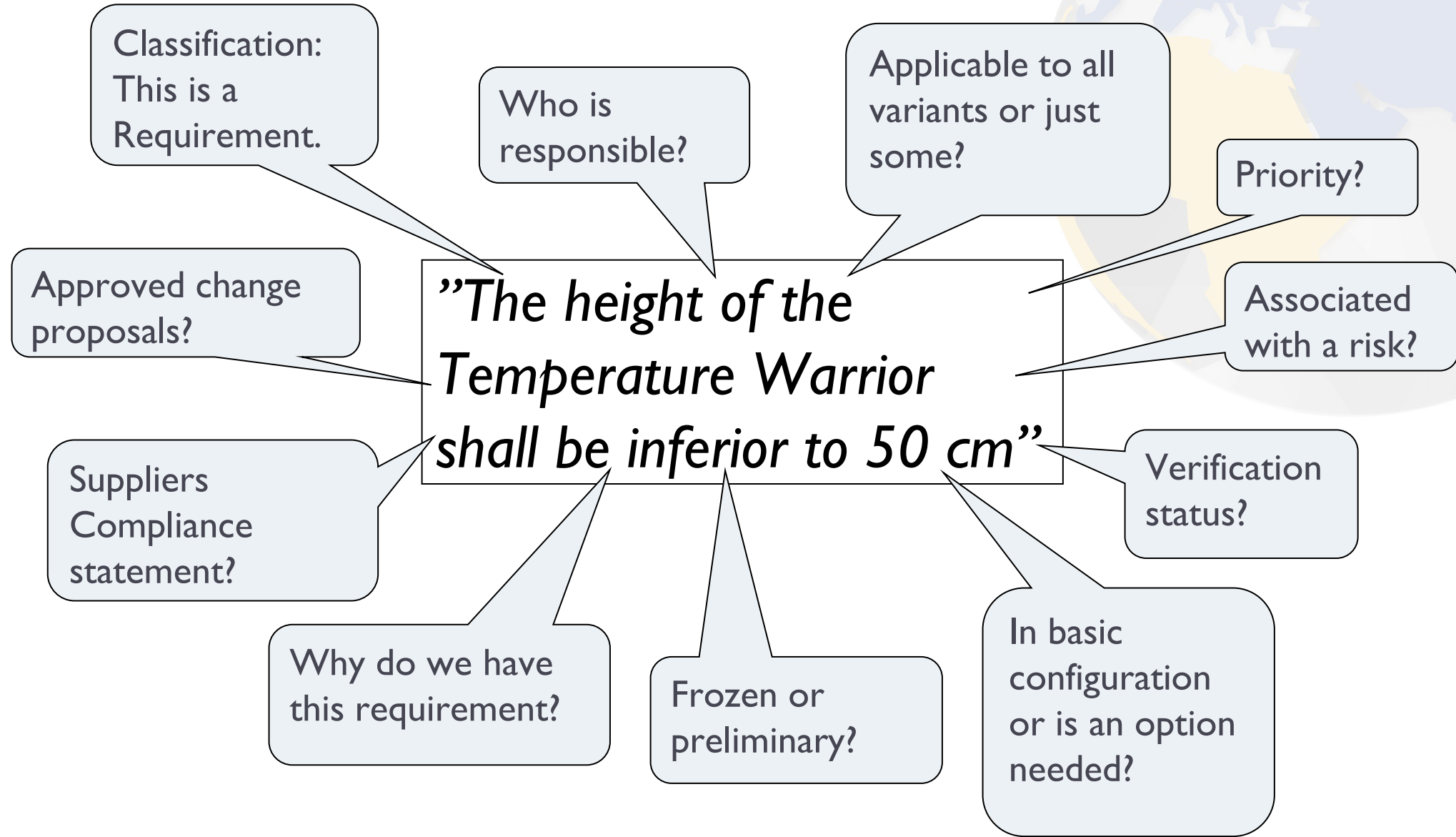
### 3.1.2

#### attribute

inherent property or characteristic of an entity that can be distinguished quantitatively or qualitatively by human or automated means

Source: ISO/IEC/IEEE 29148:2018(E)

*”Attributes capture important additional information  
(to a requirement)”*





- › A12 – Unique Identifier
- › A13 – Unique Name
- › A14 – Originator/Author
- › A15 – Date Requirement Entered
- › A16 – Owner
- › A17 – Stakeholders
- › A18 – Change Board
- › A19 – Change Status
- › A20 – Version Number
- › A21 – Approval Date
- › A22 – Date Of Last Change
- › A23 – Stability
- › A24 – Responsible Person
- › A25 – Requirement Verification Status
- › A26 – Requirement Validation Status
- › A27 – Status (Of Requirement)
- › A28 – Status (Of Implementation)
- › A29 – Trace To Interface Definition
- › A30 – Trace To Peer Requirements
- › A31 – Priority
- › A32 – Criticality
- › A33 – Risk
- › A34 – Key Driving Requirement (KDR)
- › A35 – Additional Comments
- › A36 – Type/Category

From **INCOSE Guide for Writing Requirements Summary Sheet**

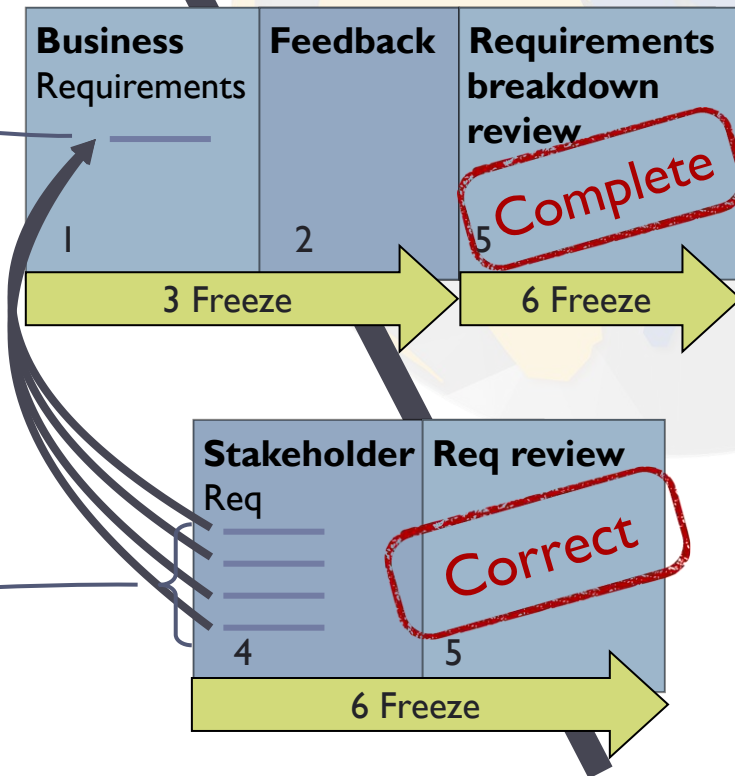
A1 - A7	Attributes to help define the requirement and its intent
A8 - A11	Attributes associated with the SOI and its verification
A12 - A36	Attributes to help maintain the requirements
A37 - A44	Attributes to show applicability and allow reuses



- A set of requirements often consist of different levels (BRS, StRS, SyRS)
  - ➔ Need to check completeness: requirements break-down between levels
  
- Verification activities often not 1:1 mapped to requirements
  - ➔ Need to check completeness: verification coverage
  
- Often more than one party involved with a (contract) Interface
  - ➔ Need for extended attributes for status and ownership

These  
requirements  
are all fine...

... but do they  
together satisfy all  
aspects of this  
requirement?



Similar approach to check Completeness in verification coverage: Do all the planned verification actions together fully verify the associated requirement?



- For each class of information, define an attribute scheme to specify what attributes you use and how to use them:
  - Attribute name
  - Use/Implication
  - Possible values (text, enumerations etc.)
  - Conditions for setting the attribute
  - Mandatory or Optional?
  - Responsible party
  - Freeze rules
- Restrict attributes to the defined attribute scheme!
  - Random attributes create confusion

Don't know how to use it?  
Do you really need it?



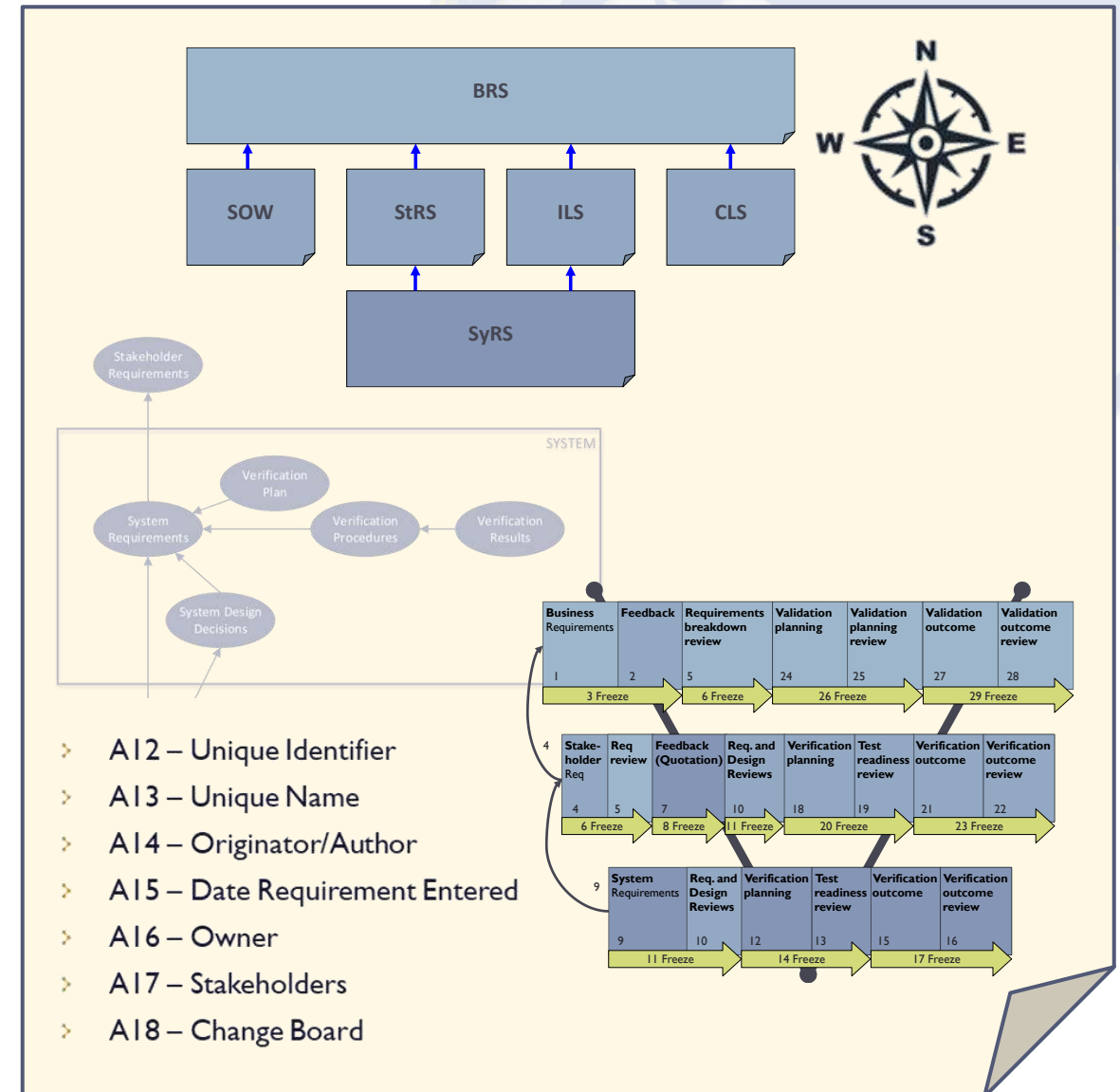
- "We must not think of the things we could do with, but only the things that we cannot do without"  
George in Three men in a boat  
(Jerome K. Jerome)

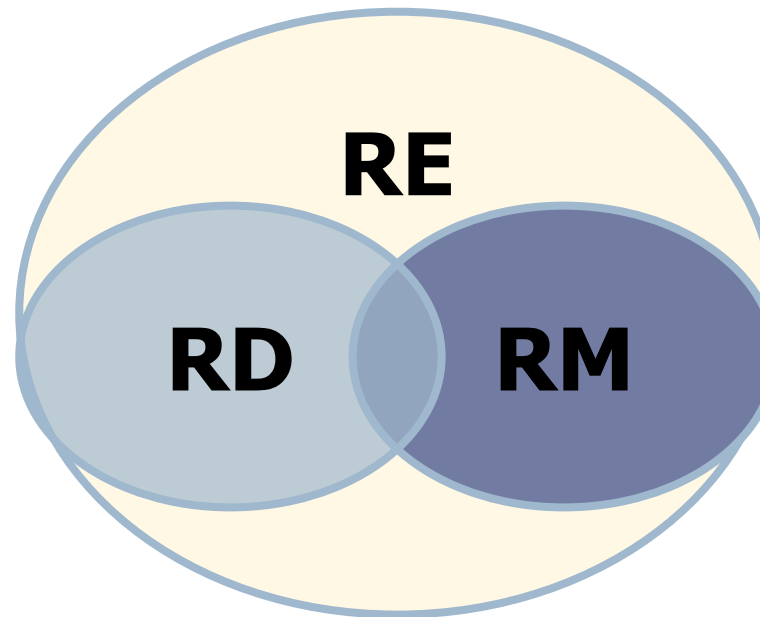




**Draw the map**

- Main components of the map:
  - Specification structure
  - Traceability rules
  - Attribute sets
  - Requirements Life Cycle
- Drawing the map is an iterative process
  - Initial ideas should be documented during Project start-up
  - Details and changes will come later on
  - Revisit during project progress





Requirements Development (RD)  
+ Requirements Management (RM)  
Requirements Engineering (RE)



**Demonstration**



# Define attributes



# Work with data in MS Word





# Trace data in different source formats



Interested?

- ▶ Contact us at [contact@reusecompany.com](mailto:contact@reusecompany.com)

Or

- ▶ Contact directly the presenter



**Lars Anderberg**  
The REUSE Company  
[lars.anderberg@reusecompany.com](mailto:lars.anderberg@reusecompany.com)



## Upcoming webinar: Connecting textual requirements and Capella Models

SES ENGINEERING Studio: Achieving the perfect equilibrium between Textual Requirements and Models in Capella enhanced by Automatic Interoperability, Quality & Traceability operations



When: Thursday, June 23

Where: <https://www.eclipse.org/capella/webinars.html>



THE  
**REUSE**  
COMPANY

