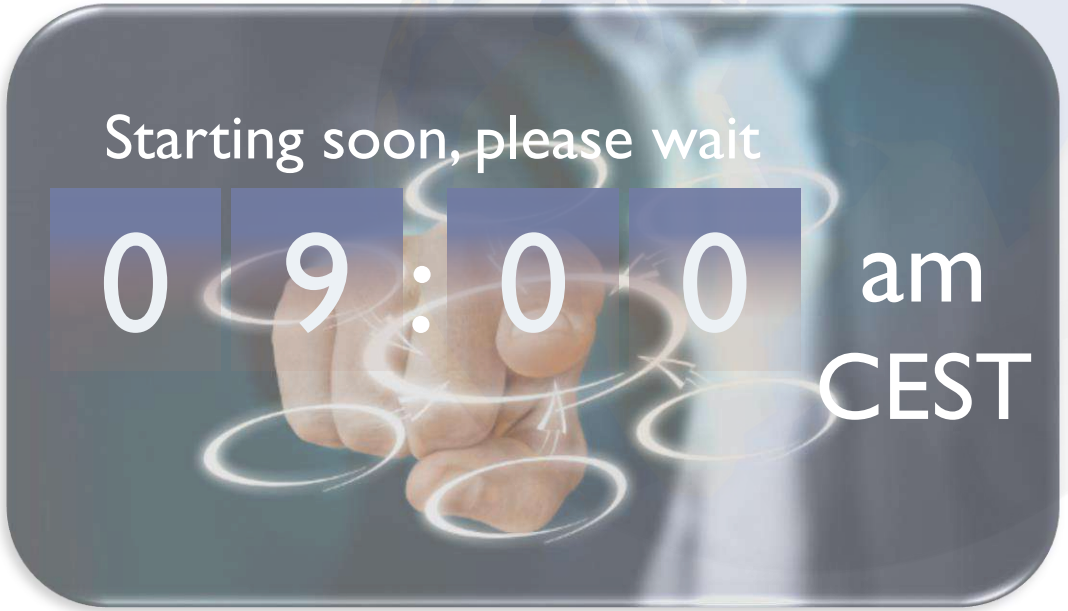


- Webinar rules:
  - You'll be muted all along the Webinar
  - There's a chatting box to ask your questions or send your comments when you want
  - Please address these comments and questions to the user "The REUSE Company" and not to the presenter directly
  - If you have any technical issue please use this chatting box, or mail us at: support@reusecompany.com
  - The Webinar will be recorded. A link to the recording will be sent to you in few days

Requirements Completeness: tips and tricks towards High-Quality specifications

José M. Fuentes
The REUSE Company
Chief Operating Officer
jose.fuentes@reusecompany.com

Cecilia Karlsson
Marketing & Communication
The REUSE Company
cecilia.karlsson@reusecompany.com

THE REUSE COMPANY

- Introduction to The REUSE Company and the speakers
- What is *Requirements Completeness*
- *Requirements Completeness*: tips and tricks
- The SES Suite and the CCC Approach
- Completeness for sets of requirements
- Completeness for individual requirements
- Live demo
- Q&A

01 The company was established in **1999**

As a spin-off of a University in Madrid

02 **System + Software Engineers**

Smart combination between Company staff and R&D from Academia

03 **Headquarters:**
Madrid (Spain)

International offices:
Stockholm (Sweden)
Tokyo (Japan) Delegation

2022:
USA
Chicago/Detroit/Miami

04 To promote a **reusable**, **scalable** and global solution to a **smart** and **interoperable** Systems Engineering environment, by offering a **semantic knowledge centric** approach.

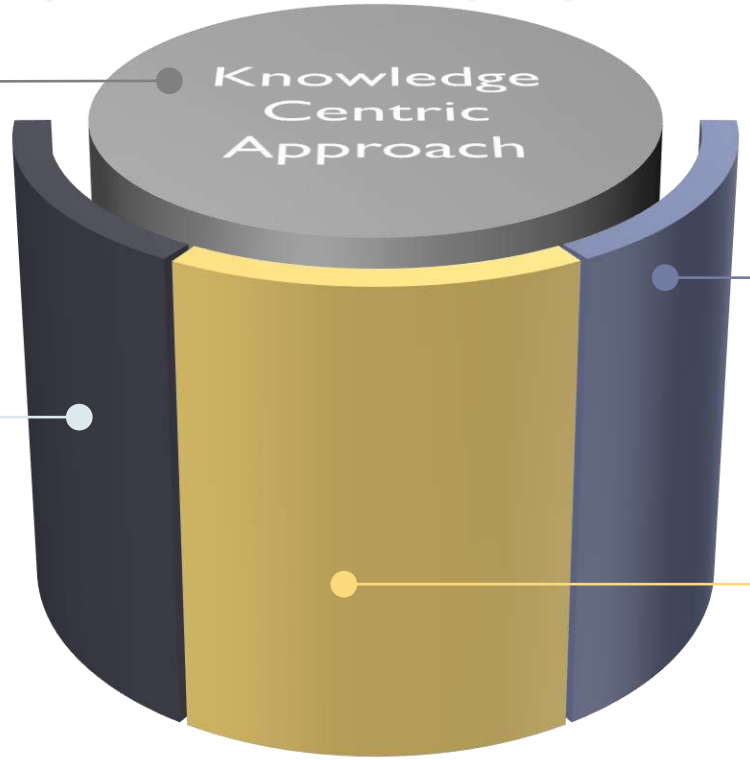
T(he) R(euse) Q(company)(ompany)

Knowledge Centric Systems Engineering (KCSE)

Global Repositories –
Archiving –
Configuration Management –

TRACEABILITY

Links and Interoperability –
Transformations –
Change management –



QUALITY (ies)

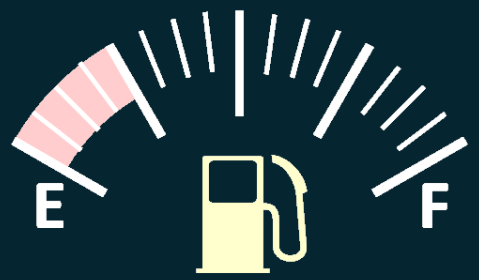
- Quality Management
- Verification & Validation management
- Risks Management
- Smart Authoring

REUSABILITY

- Interoperability
- Retrieval & Archiving
- Adaptability to Existing Toolsets

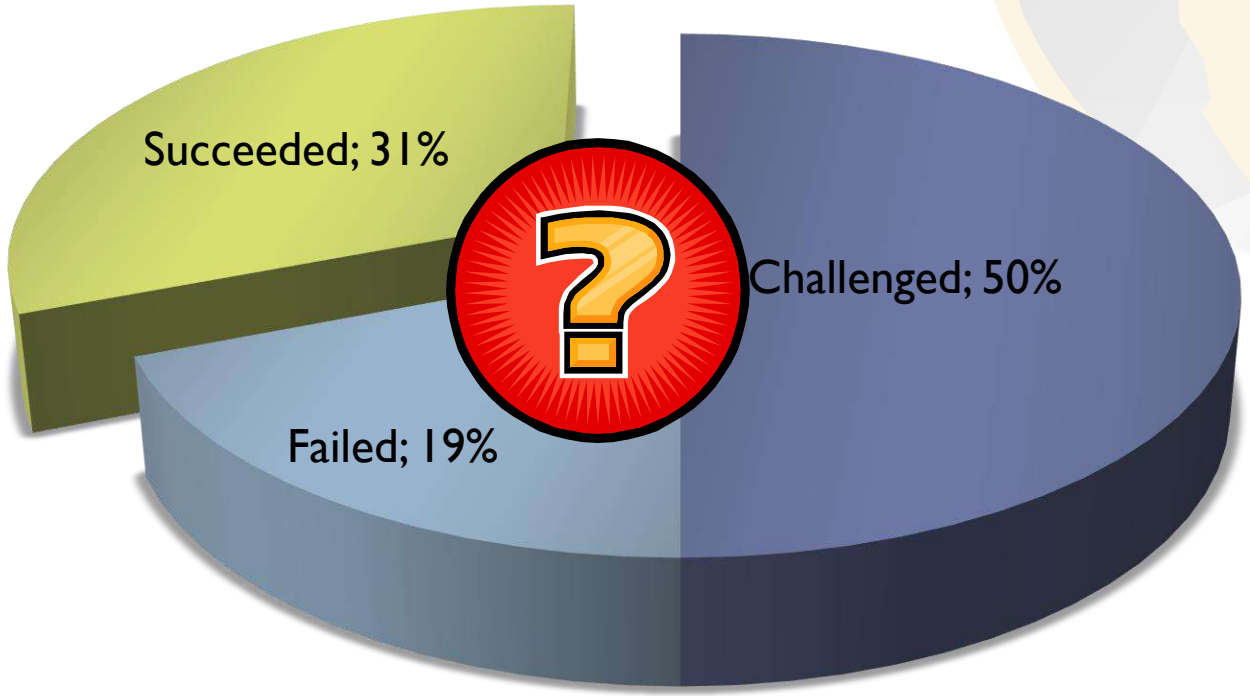
José Fuentes

- **Current position:** Chief Operating Officer at The REUSE Company
- Product manager of the Systems Engineering Suite tools during the last 5 years
- INCOSE CSEP Certified
- Graduated in the INCOSE Institute for Technical Leadership
- Active contributor to the INCOSE Guide for Writing Requirements



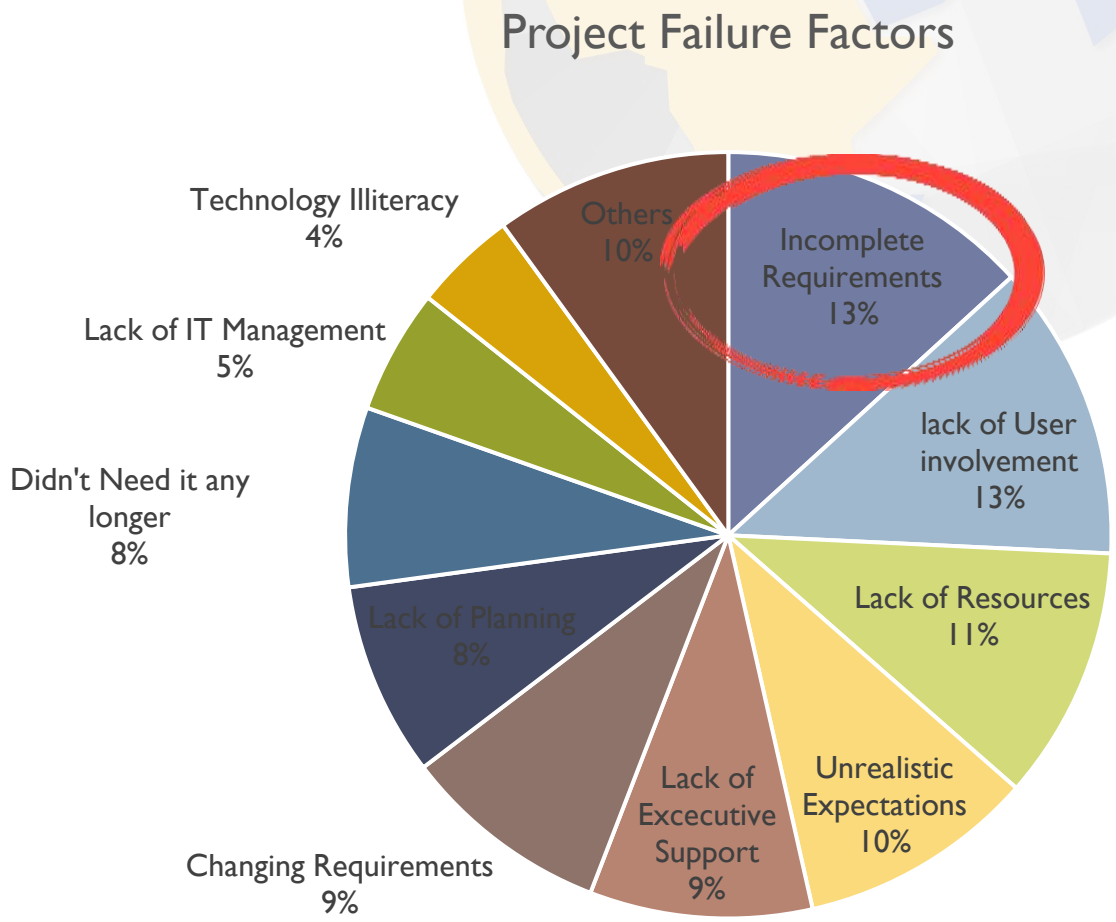
Requirement's completeness

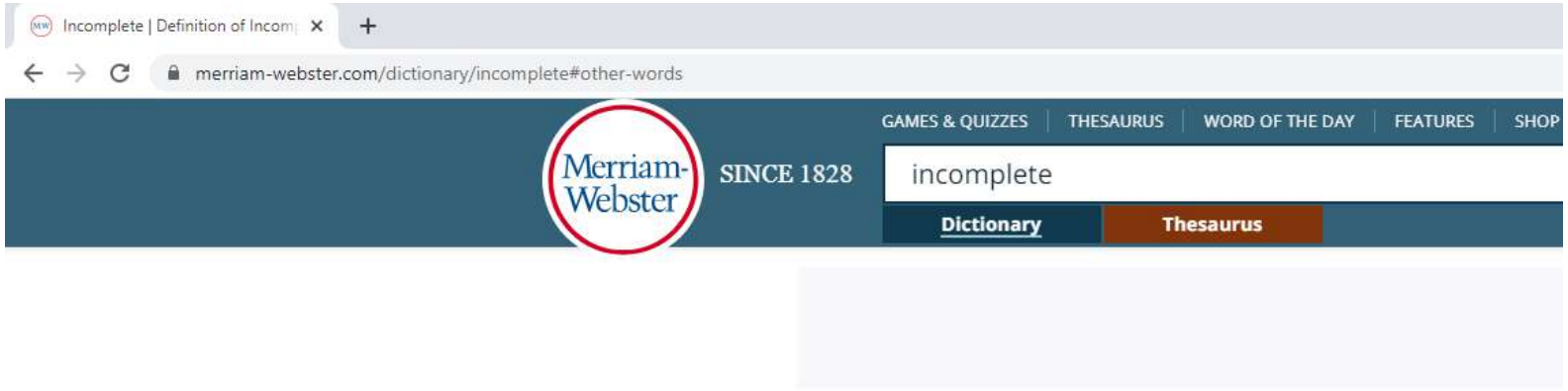
Chaos Report, 2018



Chaos Report, 2018

Project Success Factors	% of Responses
1. User Involvement	15.9%
2. Executive Management Support	13.9%
3. Clear Statement of Requirements	13.0%
4. Proper Planning	9.6%
5. Realistic Expectations	8.2%
6. Smaller Project Milestones	7.7%
7. Competent Staff	7.2%
8. Ownership	5.3%
9. Clear Vision & Objectives	2.9%
10. Hard-Working, Focused Staff	2.4%
Other	13.9%





incomplete adjective

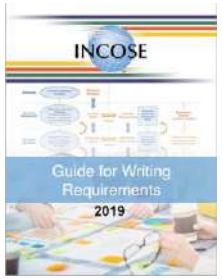
Save Word

in·com·plete | \ ˌin-kəm-ˈplēt \

Definition of *incomplete*

1 : not complete : UNFINISHED: such as

a : lacking a usually necessary part, element, or step
// spoke in *incomplete* sentences
// an *incomplete* set of golf clubs
// an *incomplete* diet



INCOSE Guide for Writing Requirements:
The requirements set stands alone such that it sufficiently describes the necessary capabilities, characteristics, constraints, interfaces, standards, regulations, and/or quality factors to meet the entity needs without needing other information
Minimum and sufficient set…

CHARACTERISTICS OF NEED AND REQUIREMENT STATEMENTS									CHARACTERISTICS OF SETS OF NEEDS AND REQUIREMENTS				
C1 - NECESSARY	C2 - APPROPRIATE	C3 - UNAMBIGUOUS	C4 - COMPLETE	C5 - SINGULAR	C6 - FEASIBLE	C7 - VERIFIABLE	C8 - CORRECT	C9 - CONFORMING	C10 - COMPLETE	C11 - CONSISTENT	C12 - FEASIBLE	C13 - COMPREHENSIBLE	C14 - ABLE TO BE VALIDATED
1	2	29	12	8	2	21	4	8	3	9	5	8	7

INCOSE GfWR

- **Characteristics**
- **Rules**
- **Attributes**

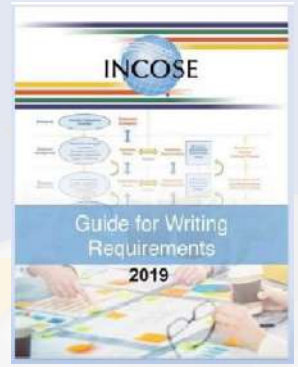
Completeness:

R24 – Avoid the use of pronouns and indefinite pronouns

R25 – Avoid relying on headings to support explanations or understanding of the requirements

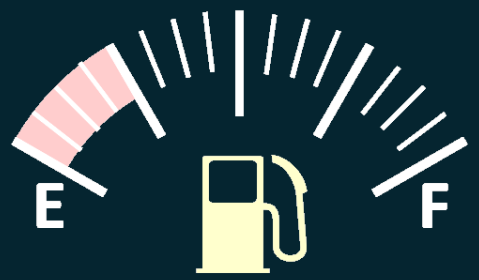
41 Rules / 14 Characteristics

			CHARACTERISTICS OF NEED AND REQUIREMENT STATEMENTS									SETS OF NEEDS AND REQUIREMENTS				
Type	Rule Number	Rule name	C1 - NECESSARY	C2 - APPROPRIATE	C3 - UNAMBIGUOUS	C4 - COMPLETE	C5 - SINGULAR	C6 - FEASIBLE	C7 - VERIFIABLE	C8 - CORRECT	C9 - CONFORMING	C10 - COMPLETE	C11 - CONSISTENT	C12 - FEASIBLE	C13 - COMPREHENSIBLE	C14 - ABLE TO BE VALIDATED
Accuracy	R01	Sentence Structure			1				1							
	R02	Use Active Voice			1				1							
	R03	Subject Verb		1	1				1			1				1
	R04	Use Defined Terms			1				1				1		1	1
	R05	Use Definite Articles			1				1							
	R06	Units			1	1			1	1						
	R07	Avoid Vague Terms			1	1			1							
	R08	No Escape Clauses			1	1			1							
	R09	No Open Ended			1	1	1		1							
Concision	R10	Superfluous Infinitives			1				1							
	R11	Separate Clauses			1											
Non Ambiguity	R12	Correct Grammar			1						1					
	R13	Correct Spelling			1											
	R14	Correct Punctuation			1											
	R15	Logical Condition			1											
	R16	Avoid Not			1				1							
	R17	Oblique			1				1							
Singularity	R18	Single Sentence			1	1	1		1		1				1	
	R19	Avoid Combinators			1		1									
	R20	Avoid Purpose					1									
	R21	Avoid Parentheses					1									
	R22	Enumeration			1		1									
	[illegible]	[illegible]			1		1									
Completeness	R24	Avoid Pronouns			1	1			1							
	R25	Use Of Headings				1										
Realism	R26	Avoid Absolutes						1	1					1		
Conditions	R27	Explicit				1			1							
	R28	Explicit Lists			1				1							
Uniqueness	R29	Classify										1	1	1		
	R30	Express Once	1								1		1	1		
Abstraction	R31	Solutionfree		1												
Quantifiers	R32	Universals			1				1	1						
Tolerance	R33	Value Range			1	1		1	1	1				1		
Quantification	R34	Measurable			1	1			1					1		
	R35	Temporal Indefinite			1	1			1							
Uniform Language	R36	Use Consistent Terms			1					1	1		1		1	1
	R37	Define Acronyms			1						1		1		1	1
	R38	Avoid Abbreviations									1		1		1	1
	R39	Style Guide				1	1				1		1		1	1
Modularity	R40	Related Requirements									1		1		1	
	R41	Structured										1	1		1	1



46 Attributes

Att	Attribute	Attributes to Help Define the Requirement and its Intent	Associated with the System of Interest (SOI) Verification	Attributes to Help Maintain the Requirements	Attributes to Show Applicability and Allow Reuse
A01	Rationale*	1			
A02	SOI Primary Verification or Validation Method*	1			
A03	SOI Verification or Validation Approach	1			
A04	Trace to Parent*	1			
A05	Trace to Source*	1			
A06	Condition of Use	1			
A07	States and Modes	1			
A08	Allocation*	1			
A09	SOI Verification or Validation Level		1		
A10	SOI Verification or Validation Phase		1		
A11	SOI Verification or Validation Results		1		
A12	SOI Verification or Validation Status		1		
A13	Unique Identifier*			1	
A14	Unique Name			1	
A15	Originator/Author*			1	
A16	Date Requirement Entered			1	
A17	Owner*			1	
A18	Stakeholders			1	
A19	Change Board			1	
A20	Change Status			1	
A21	Version Number			1	
A22	Approval Date			1	
A23	Date of Last Change			1	
A24	Stability			1	
A25	Responsible Person			1	
A26	Need or Requirement Verification Status*			1	
A27	Need or Requirement Validation Status*			1	
A28	Status (of the Need or Requirement)			1	
A29	Status (of Implementation)			1	
A30	Trace to Interface Definition			1	
A31	Trace to Peer Requirements			1	
A32	Priority*			1	
A33	Criticality or Essentiality*			1	
A34	Risk (of Implementation)*			1	
A35	Risk (Mitigation)			1	
A36	Key Driving Need or Requirement (KDN/KDR)			1	
A37	Additional Comments			1	
A38	Type/Category			1	
A39	Applicability				1
A40	Region				1
A41	Country				1
A42	State/Province				1
A43	Application				1
A44	Market Segment				1
A45	Business Unit				1
A46	Business (Product)Line				1

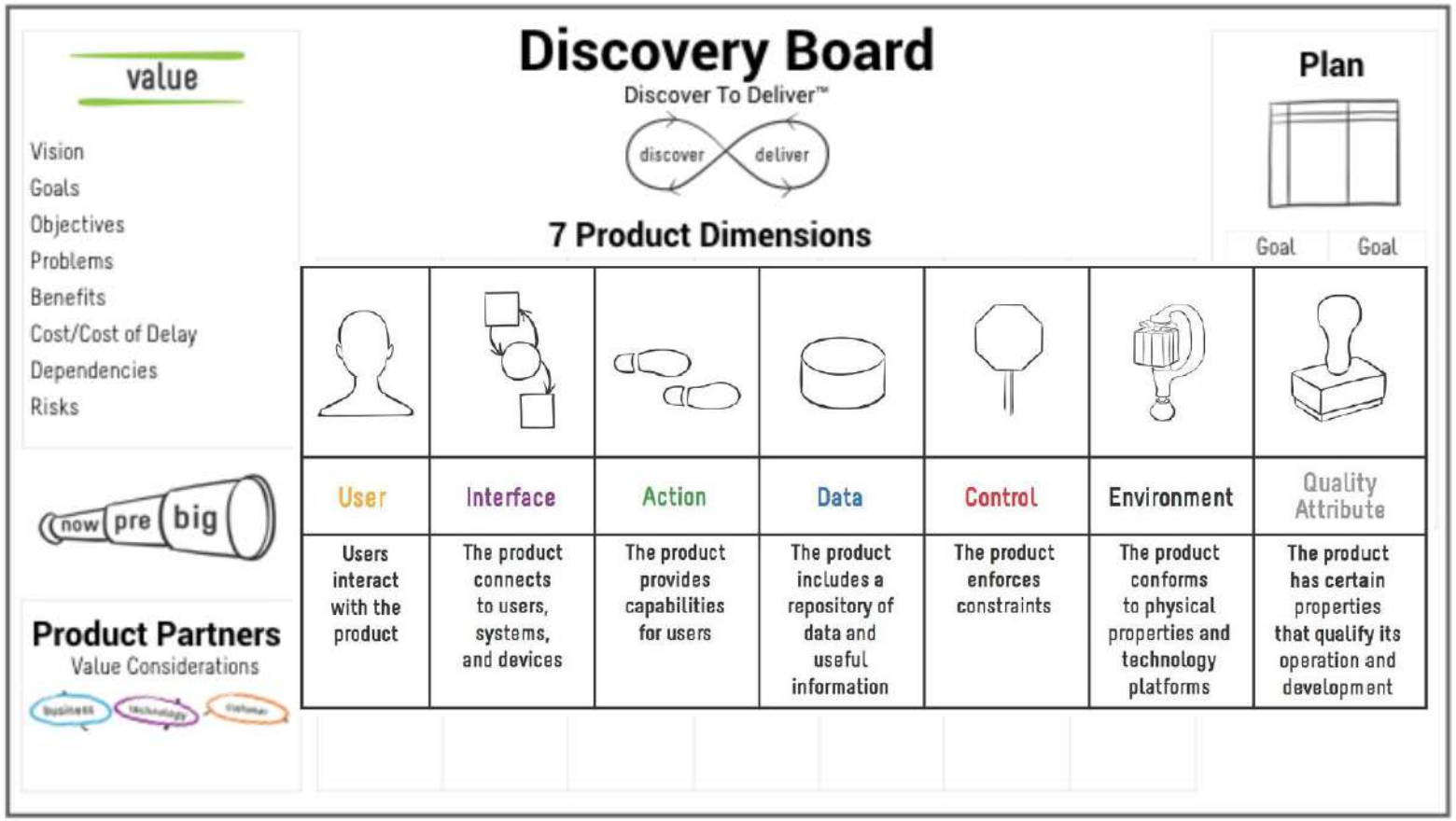


Requirement's completeness

Tips and tricks

- **Every stakeholder is important:**
  - The Product or Service
    - Contains no stakeholders
  - The System (of interest)
    - The Product or Service plus the people who operate the product or deliver the Service
    - Also often includes training, support and maintenance
  - The Containing System
    - Those who immediately benefit from the functions carried out by the System or Interface with it
    - Are usually, but not necessarily, different from the operators
  - The Wider Environment
    - People who are affected indirectly, such as derived benefit of induced harm.

- The 7 product dimensions: "Discover to deliver: Agile product planning and analysis". Ellen Gottesdiener, Mary Gorman



Do not forget *safety* requirements!

- Mind all different types of requirements:
  - According to the NASA

    Systems Engineering Handbook:

- Writing high quality requirements starts with asking the proper questions:
  - The ***What*** vs the ***How***
  - But also and mainly the ***Why***

  - What: SoI, SoS, entities and interfaces
  - Who: stakeholders
  - What: functions
  - How much/many: performance
  - While: state or mode
  - When: trigger

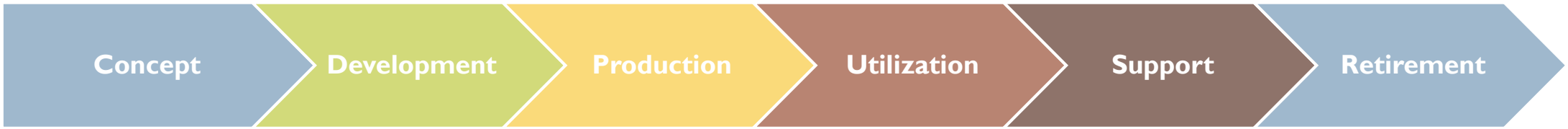
- Avoid ubiquitous requirements:
- EARS patterns: *The <system name> shall <system response>*

SysR-001. The car shall accelerate 0 to 100 km/h in less than 10 seconds

SysR-001.
While the gearbox is in D and the parking brake is not engaged,
when the driver steps the gas pedal,
the car shall….
in less than 200 ms.

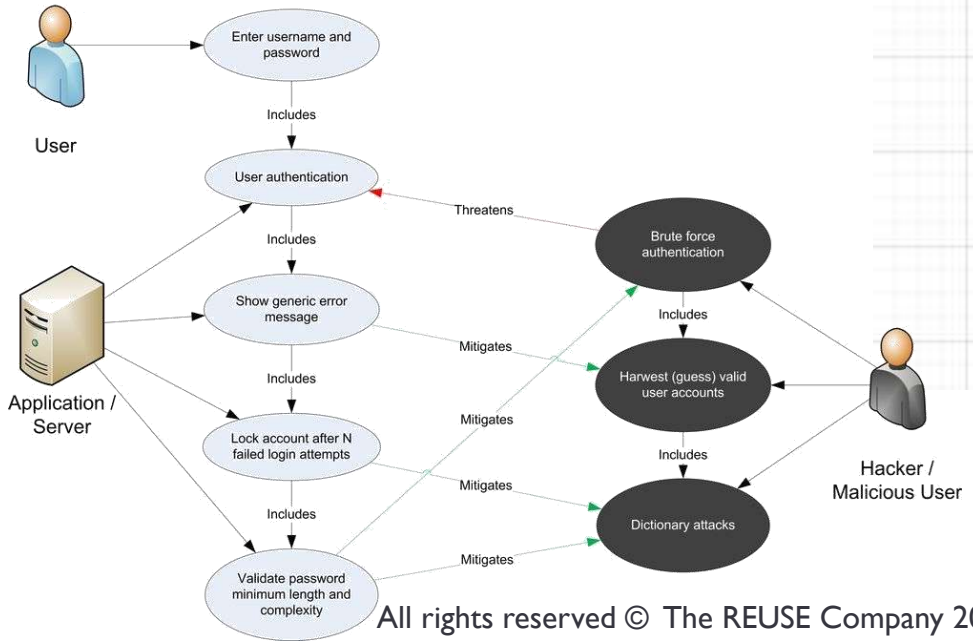
- Writing high quality requirements starts with asking the proper questions:
  - The ***What*** vs the ***How***
  - But also and mainly the ***Why***

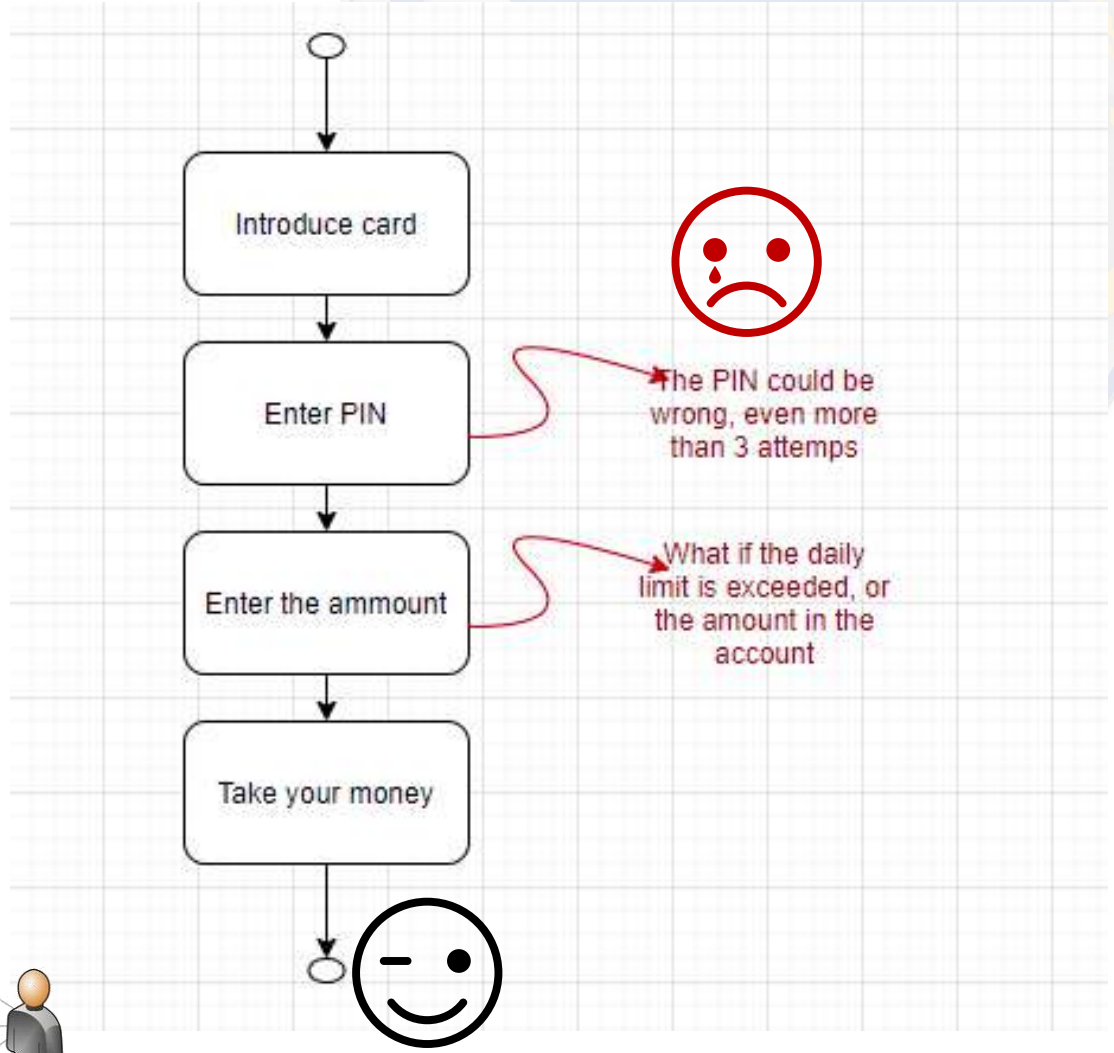
  - When: system life cycle



  - Don't forget the transition requirements: from the current system to the new system

- Do not focus just on the happy path…
- … work out requirements to manage the exceptions, unwanted scenarios…!
- Cover all the use cases, but also the mis-use cases

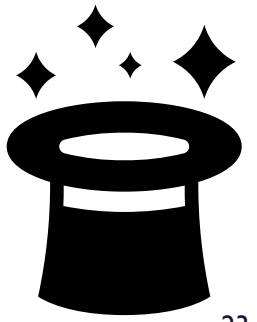




- Apply the principles of mindfulness:
  - Non-judgment: there are no good or bad stakeholders
  - Patience: requirements cannot be collected in one day
  - Beginners' mind: do not force your solutions or ideas, listen first
  - Confidence: trust the stakeholders
  - Acceptance: some workshops will address more needs, but some not. No worries, just keep it up!
- And the principles of zen:
  - "If you walk, just walk. If you sit, just sit; but whatever you do, don't wobble" "When hungry, eat your rice; when tired, close your eyes"
  - "When dealing with reqs, deal with reqs, do not rush into a solution"

- Dealing with stakeholders is all about communication
- Apply the principles of Neuro-Linguistic Programming (NLP):
  - "The map is not the territory"
  - Our mental filters affect the reality as is expressed by stakeholders
  - "Generalization – Deletion – Distortion" leads to incompleteness:
    - Generalization: only the "common" case is addressed, the special cases are not considered
    - Deletion: some aspects are considered as irrelevant, and thus removed from the discourse
    - Distortion: whatever that is not the way I think is "adapted" to my way of thinking

- **Other tips:**
  - Identify all stakeholders and improve communication channels with them all
    - **Completeness is a perception of the mandatory needs of stakeholders**
  - Use of requirements checklists
  - Use templates:
    - For the document itself
    - For the structure of the textual part of the requirement (aka patterns)
    - For the attributes, links that must come with the requirement
  - Requirements reuse
  - Improve the communication skills of your business analysts
  - Establish a formal inspection process
  - Consider the elicitation process as an iterative and recursive process
  - In cascading down requirements, if the first layer is not complete, "nothing" can be done to keep the remaining layers complete

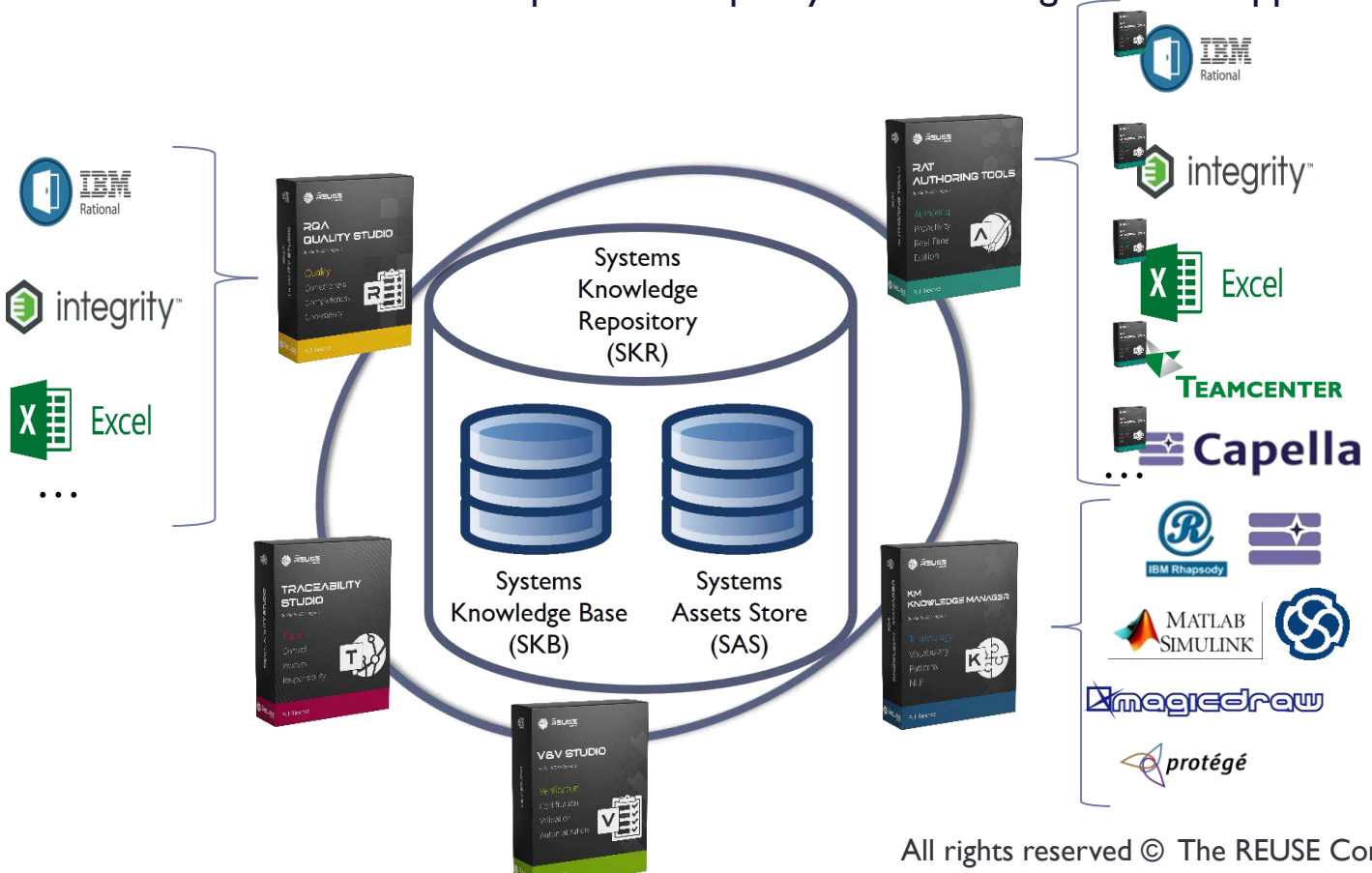


- And remember, **questions** really matter:
  - "It's smarter to ask "too many" questions and look like an idiot the first day, …
  - …than not asking at all and demonstrate that you're actually an idiot the last day"

- **Asking the right question is the key to requirements completeness!!**

The SE Suite and the CCC Approach

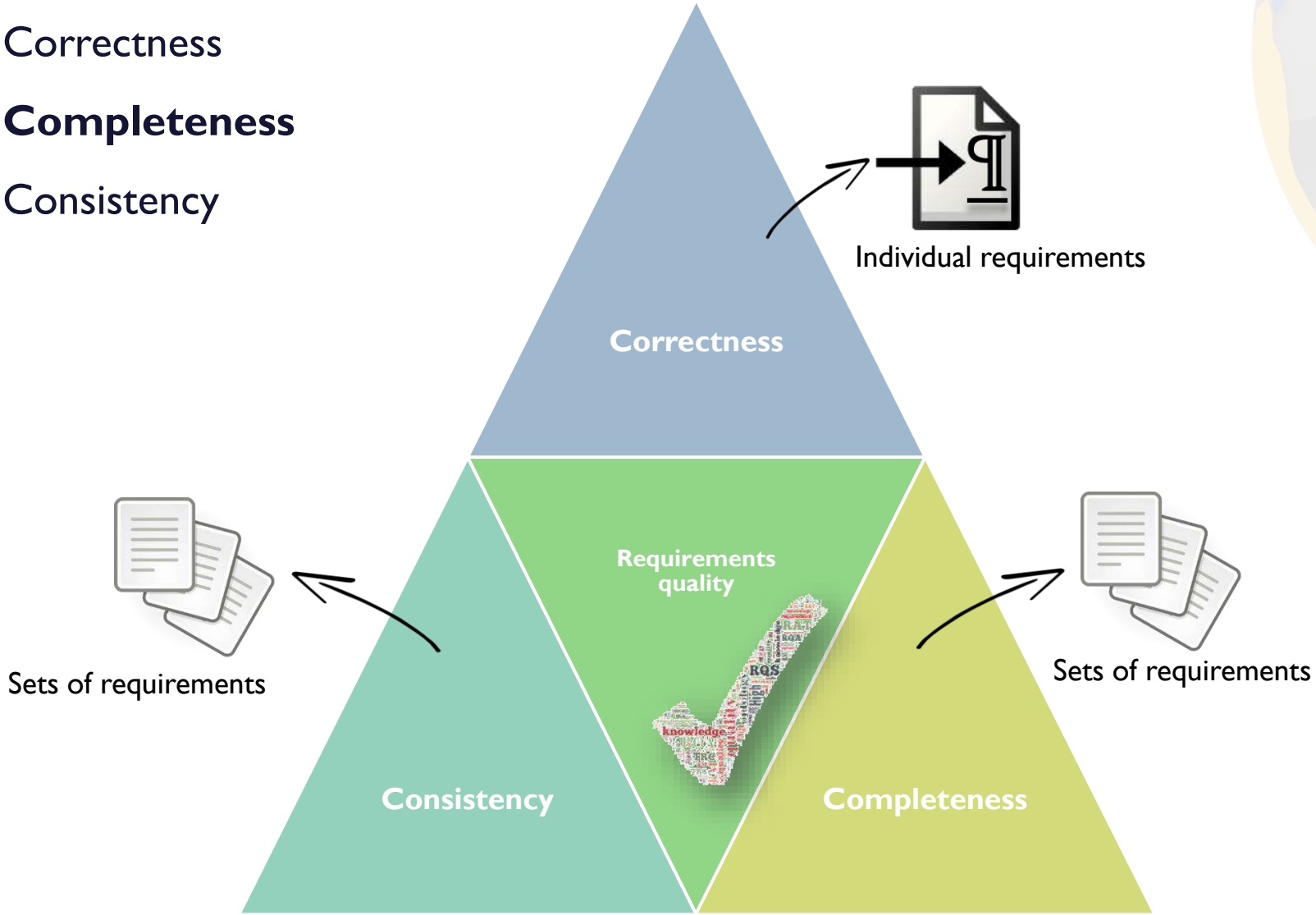
- The Systems Engineering Suite (SES) tackles requirements quality management by offering a set of tools and processes
- Automatic measurement of requirements quality metric
- Support to Requirements Authoring
- SES Suite models requirements quality metrics using the CCC approach (Correctness, Consistency and Completeness)

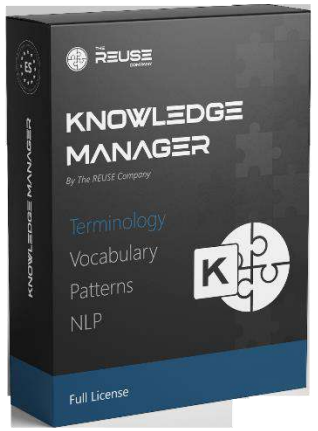


- **RQA Quality Studio / V&V Studio:** to setup, check and manage the quality of a requirements specification
- **Rich Authoring Tool (RAT):** to assist authors while they are creating or editing requirements
- **Knowledge Manager (KM):** to manage knowledge around a requirements specification: dictionaries, glossaries, concept maps, knowledge models, ontologies, patterns…

The CCC approach in RQA – QUALITY Studio to assess requirements quality :

- Correctness
- **Completeness**
- Consistency

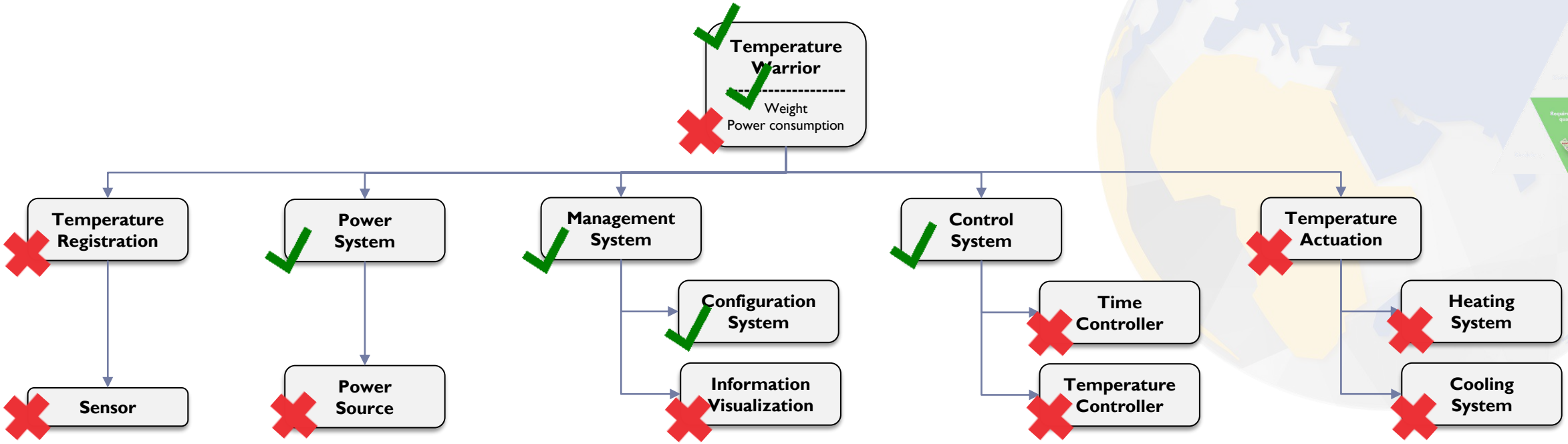




Completeness for sets of requirements

- Completeness at specification/project level:
  - Are all the expected requirements types involved in your specifications?
  - Are all the key concepts (from the ontology or from other models, e.g. blocks, states, signals, properties…) properly covered?
    - Does the whole set of requirements documents include requirements for all the elements of the system according to a block diagram (architecture)?
    - Does the spec. include requirements describing the behavior of the system elements in any of their possible states and modes?
    - Does the spec. include requirements mentioning all the signals?
  - Are all the properties stated for every system element?
    - For those properties in a model whose value is to be provided in the spec, is the value actually provided?

Completeness for sets of requirements: vs models



The **Temperature Warrior** shall have a **Control System**.

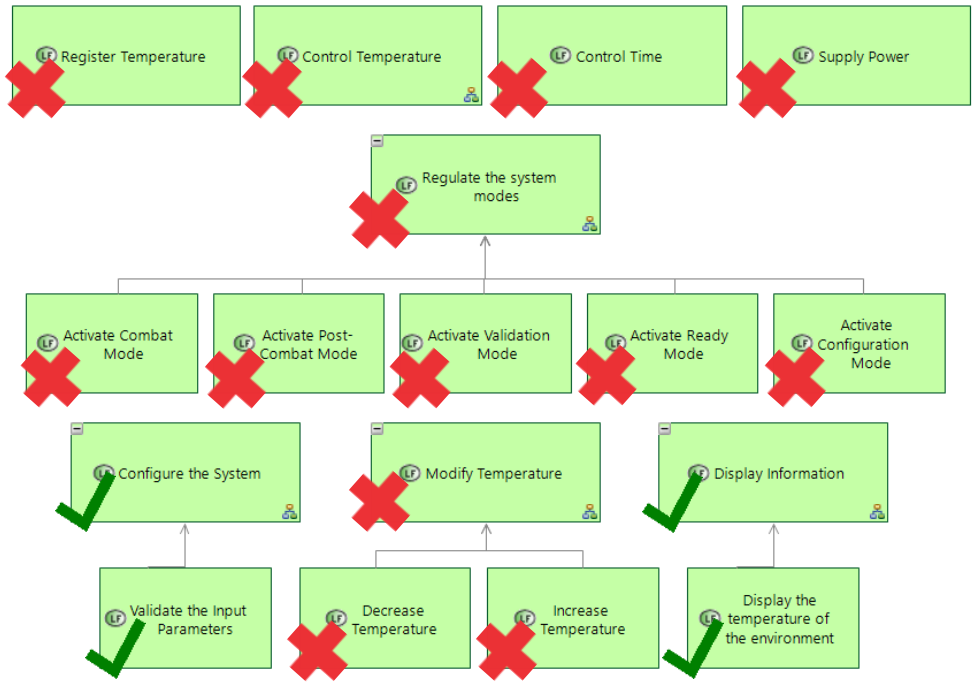
The **weight** of the Temperature Warrior shall be **less than 50 kg**.

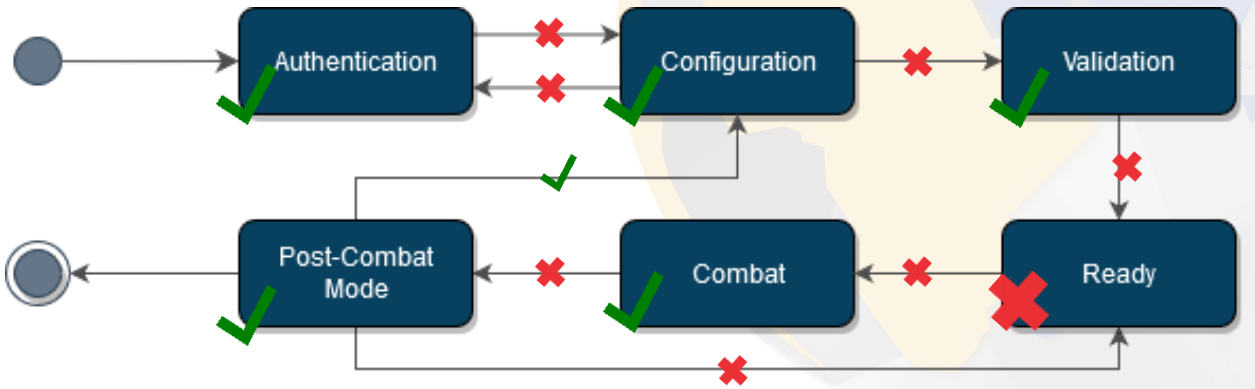
The Temperature Warrior shall have a **Power System**.

The Temperature Warrior shall have a **Management System**.

The Management System shall a **Configuration System.**

Completeness for sets of requirements: vs models





When the Temperature Warrior is loaded and operating, the Temperature Warrior shall enter the **Authentication Mode**.

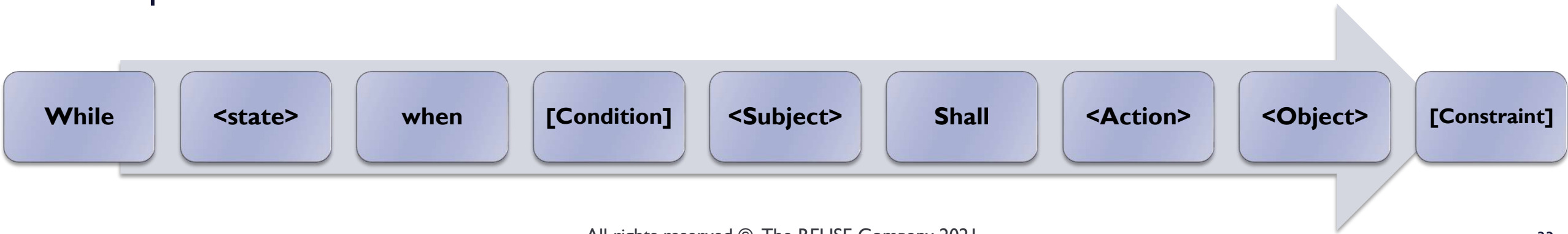
While the Temperature Warrior is in the **Post-Combat Mode** (and when the Administrator selects the New Combat command), the Control System shall activate the **Configuration Mode**.

When the **Validation Mode** is initialized, the Temperature Warrior shall **validate the required parameters,** according to the displayed instructions on the Client's GUI.

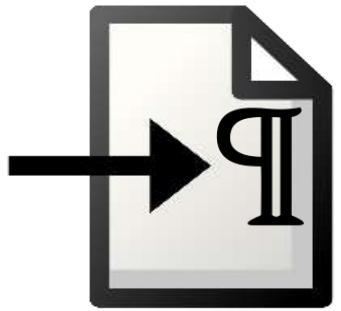
While the Temperature Warrior is in **Combat Mode**, the Temperature Warrior **shall display on a screen the temperature registered** by the sensor.

Completeness for individual requirements

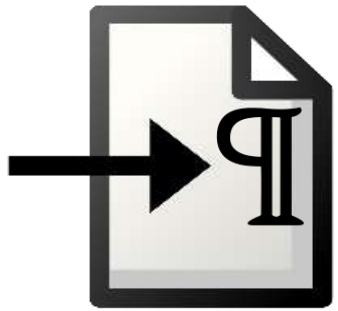
- Completeness at requirement level:
  - "ActionX shall be done"
    - What system/subsystem is responsible for this action?
    - When the system is in which state/mode?
    - Under what conditions?
    - How fast / how well?: performance
    - For what actor?
  - Does every requirement include all the agreed parts (condition, subject…): following patterns



- Completeness at requirement level:
  - Every number must quantify an entity, or measurement unit
    - "When the temperature > 30, the A/C System shall …": "30" what!?
  - State the values for the mentioned properties with tolerances: e.g. 12V±0.5V
    - "When the level of oil is 5 liters, the car shall…": difficult to find 5.0000 liters
    - If the value for the tolerance is too little, is almost as having no tolerance at all
  - No open ended: "*etc*", "*among others*",…
  - No vague content
  - No TBD, TBC, TBx…



- A requirement is not only the statement, also the expression:
  - Links:
    - Are your requirements properly linked? At the different levels?
    - To requirements at other levels, to models, to test cases…
    - Missing links is also a source of incompleteness
  - Attributes:
    - Missing the necessary information in other attributes makes your specification not complete
  - Both make difficult to meet other quality characteristics:
    - Traceable
    - Ranked
    - Verifiable
    - …



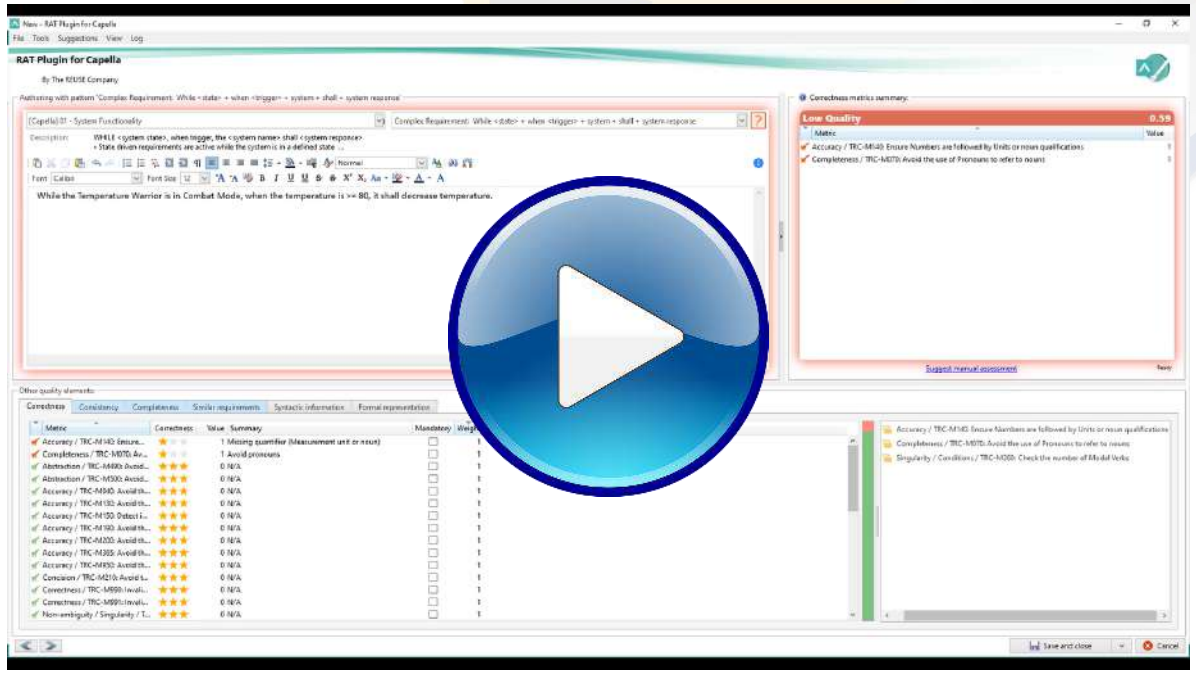
Checking requirements completeness

Demo

Requirements Completeness: Live Demo: Use case #1

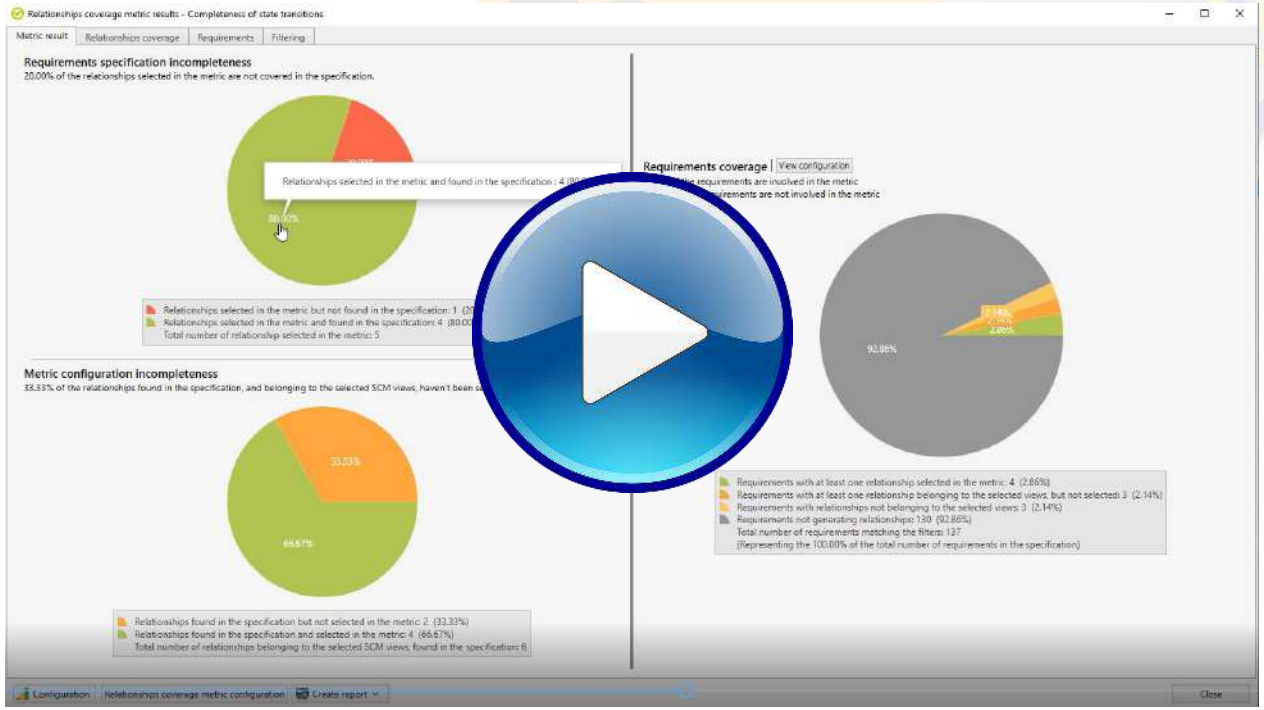
- **Use case #1:** Write requirements based on patterns – Completeness for individual requirements
- **Steps:**

1. Open RAT – AUTHORING Tools as an add-on in Capella
2. Choose a **pattern** based on the type of desired req
3. Fill out the individual elements based on the structure of the pattern
4. What if the structure of the pattern is not met
5. What if the requirement includes all the mandatory blocks



- **Use case #2:** Completeness reqs vs model
- **Steps:**

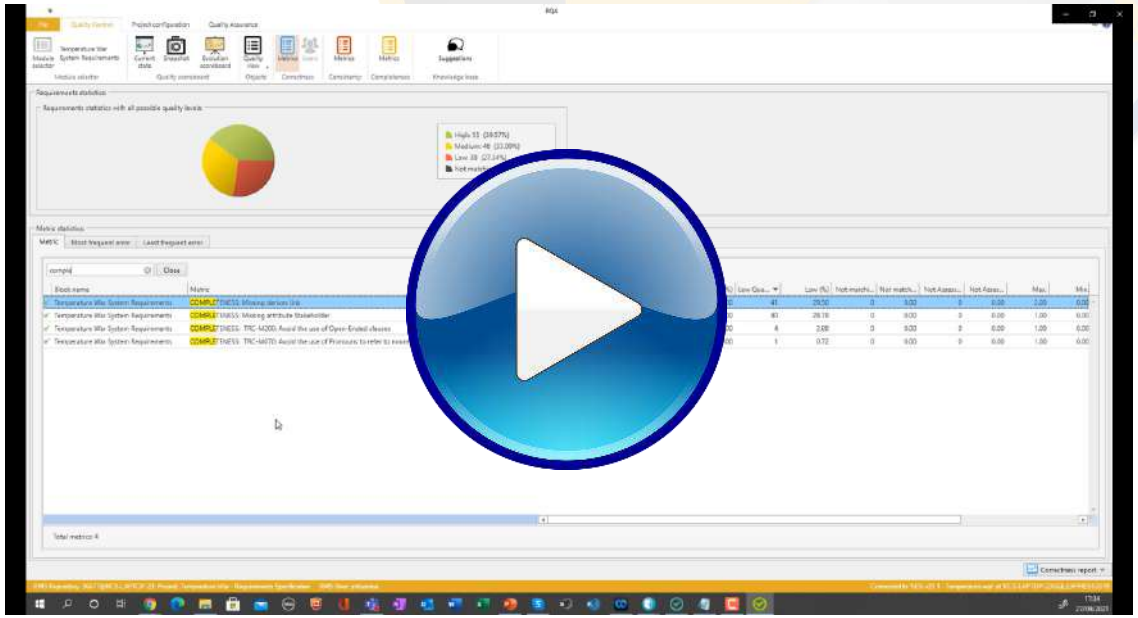
1. The **RQA – QUALITY Studio** is already connected to a formal module in DOORS and a model in Capella
2. Check the names of the states, are all those names used in the specification?
3. Check the transitions among states
4. Select all the documents in the spec, and check if all the components are mentioned in the spec



Requirements Completeness: Live Demo: Use case #3

- **Use case #3:** Other completeness checks in RQA
- **Steps:**

1. Open the **RQA – QUALITY** Studio
2. Connect to a formal module in DOORS, the TW SysRS
3. Check for metrics like: use of TBx, use of open-ended, use of pronouns…
4. Detection of missing values in attributes
5. Detection of missing values in links
6. Detection of missing tolerance values
7. Make sure which types of requirements are not involved in the specification




Q&A

CONSISTENCY
CHALLENGING the INCOSE Consistency Metrics

- **Why CHALLENGING the INCOSE Consistency Metrics might benefit your requirements?**
  - Consistency is one of the most undervalued yet extremely important issues whenever we refer to the requirements quality. As one of the CCC (Correctness, Consistency and Completeness) quality properties, Consistency is core to achieving any successful requirements management, and thus, neglecting it may lead to fatal errors in the project's development.
  - More specifically, this factor that preserves the coherence between requirements, particularly technical ones concerning any system, subsystem, or component. For instance, tracking the presence of overlapping and duplicated data, confronting the excess or lack of tolerance, and checking the measurement units' correspondence.
  - The goal of this webinar is to guide you through the basic notions of Consistency briefly described in the INCOSE Standard and show you how to take them to the next level with several practical examples, supported by The REUSE Company's tools.

- **Dates:** October 19th and 21st, 2021


Q&A

Contact information

José M. Fuentes

jose.fuentes@reusecompany.com

+34 912 17 25 96

@ReuseCompany

https://www.linkedin.com/in/josemiguelfuentes/

THE REUSE COMPANY