

## ➤ Webinar rules:

- You'll be muted all along the Webinar
- There's a *Question* section to ask your questions or send your comments when you want
- The Webinar will be recorded. A link to the recording will be sent to you in few days

# Requirements Completeness

How to detect missing information  
between

Textual requirements & Models



**Ilyes Yousfi**

Senior Key Account Manager  
The REUSE Company  
[ilyes.yousfi@reusecompany.com](mailto:ilyes.yousfi@reusecompany.com)



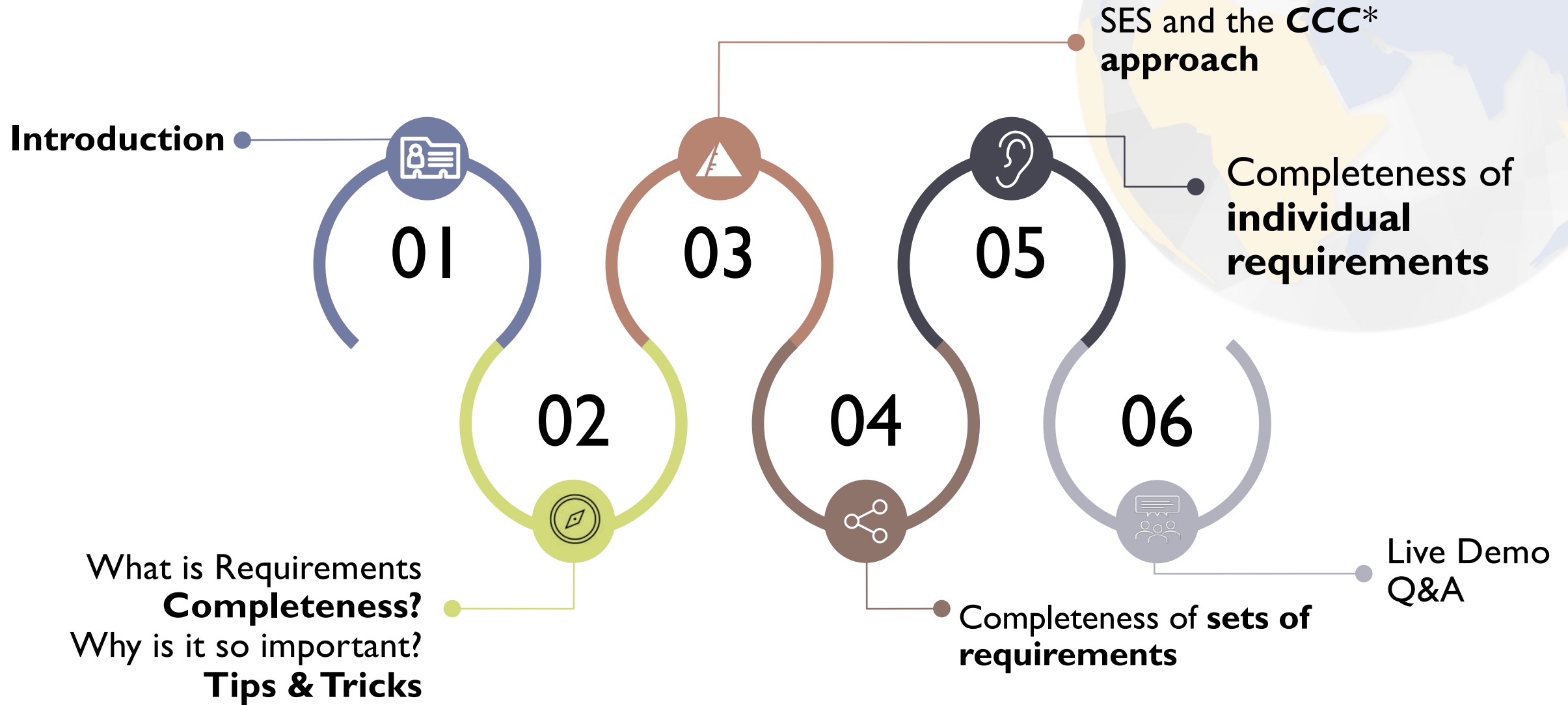
**Cecilia Karlsson**

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



THE  
**REUSE**  
COMPANY

\*Correctness, Consistency, Completeness





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



- RQA – QUALITY Studio
- RAT – AUTHORIZING Tool
- KM – Knowledge Manager
- TRACEABILITY Studio
- V&V Studio
- REx – REQUIREMENTS Engineering Tool for MS Word
- SES ENGINEERING Studio



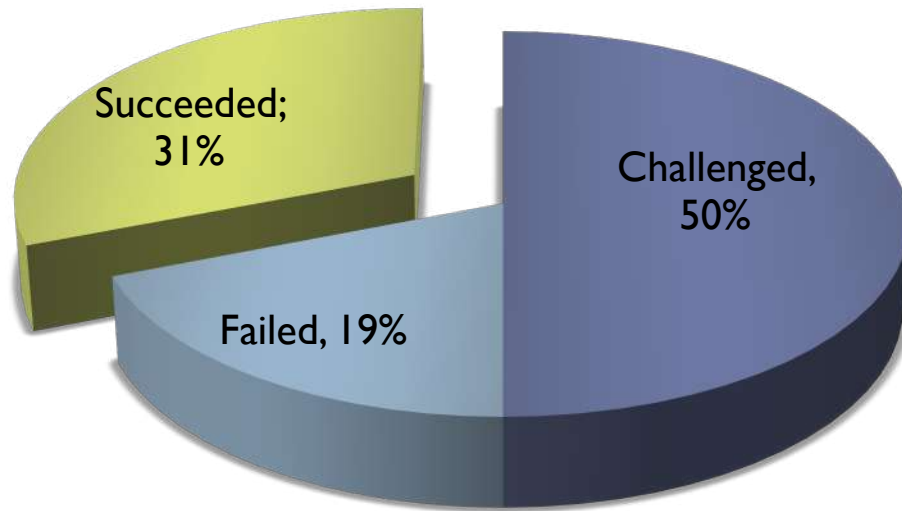
**Ilyes Yousfi**

- **Current position:** Senior Key Account Manager at The REUSE Company
- Master's degree from the University of Montreal (Canada) and the IMT Atlantique School of Engineering (France).
- Background in energy and mechanical engineering
- Involved in a research project around the environmental impacts of end-of-life management of aircrafts (2014)
- Consulting services to help industry actors leverage and digitalize Systems Engineering activities.
- Passionate about international projects and learning languages, Ilyes speaks 4 languages fluently: English, French, German and Spanish.



**Requirements  
Completeness**

**What & Why?  
Tips & Tricks**

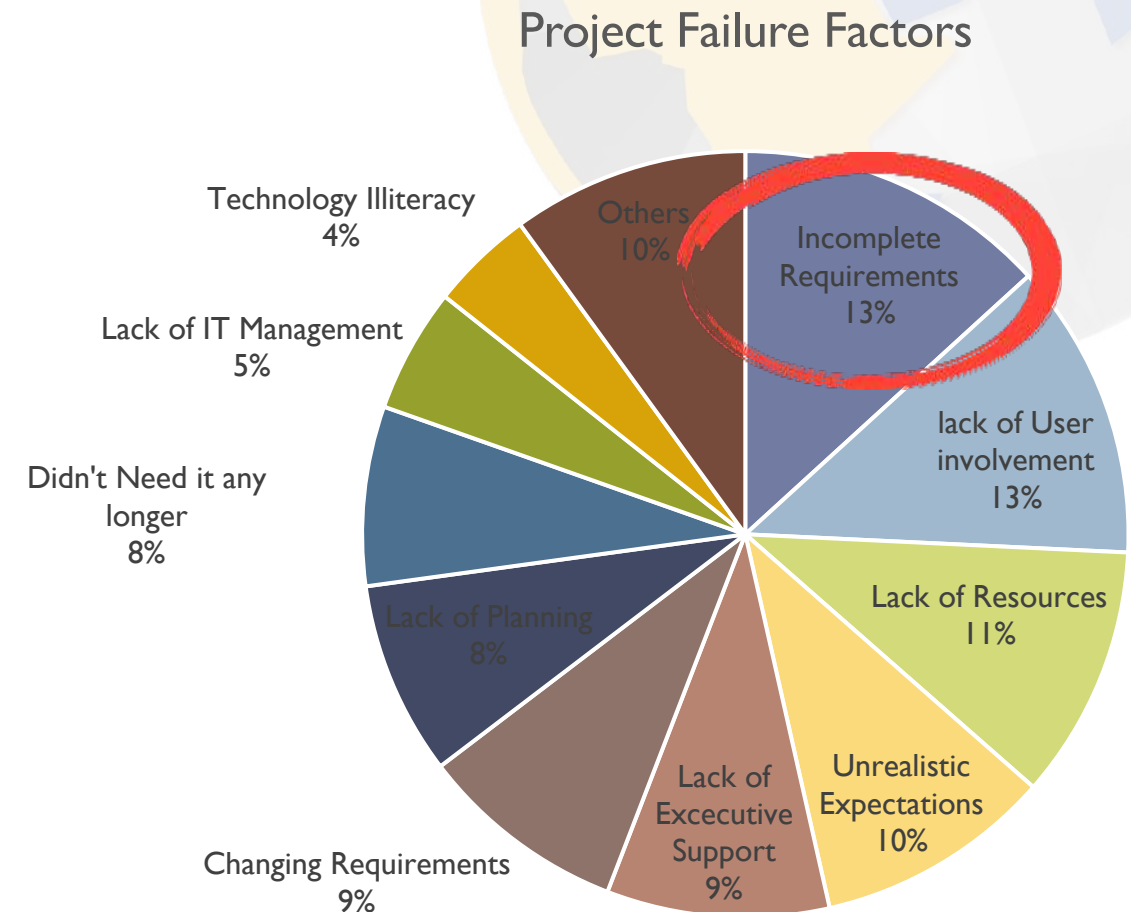


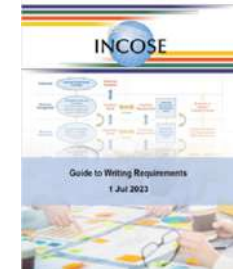
Source: Chaos Report, Standish Group (2018)

- Why are so many projects failing ?
- How to avoid facing situations that could lead to project failure ?

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%

Source: Chaos Report, Standish Group (2018)





# 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 to 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...

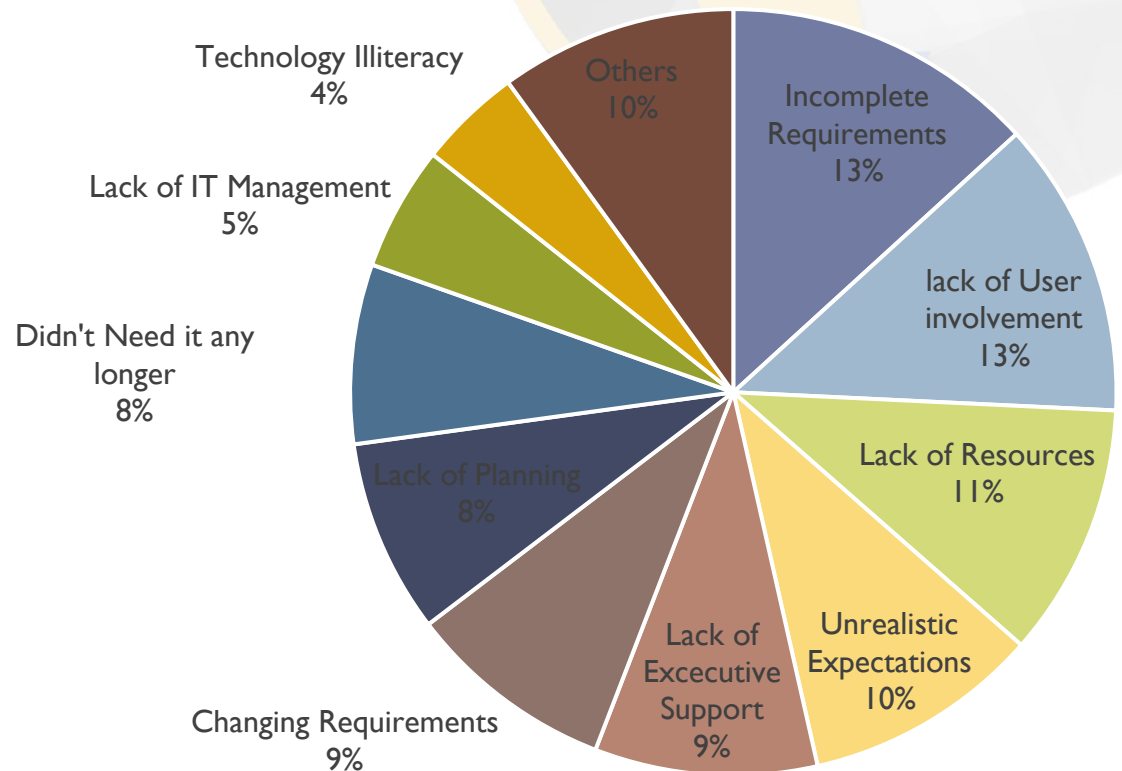


- How is requirements completeness affecting project success?
  - Lack of User/Customer acceptance
  - Underestimated Project cost & schedule
  - Failed development – Technology illiteracy
  - Unverifiable requirements
  - Unsafe requirements & systems

“Major system failures and accidents are often caused by missing requirements.”

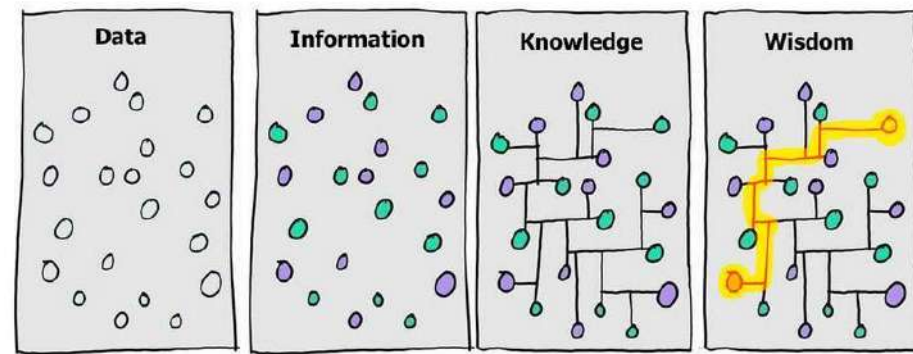
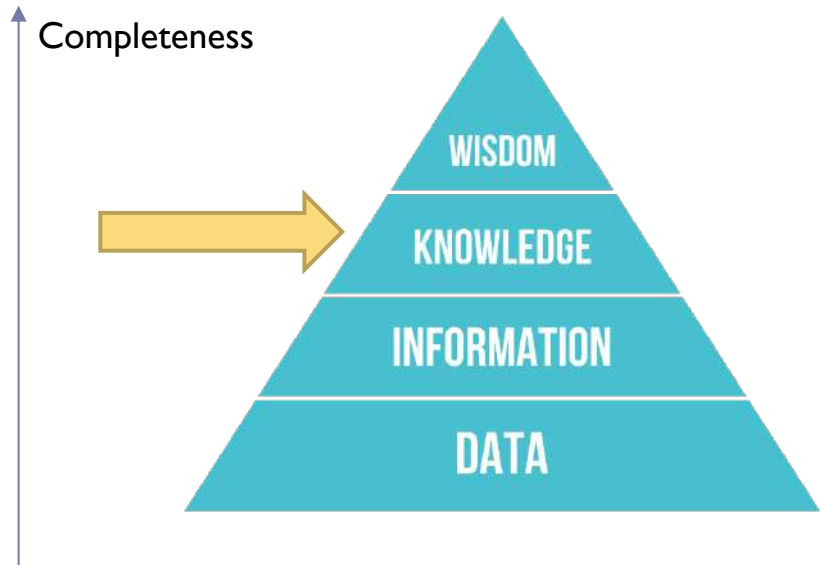
*D. Firesmith*

Project Failure Factors



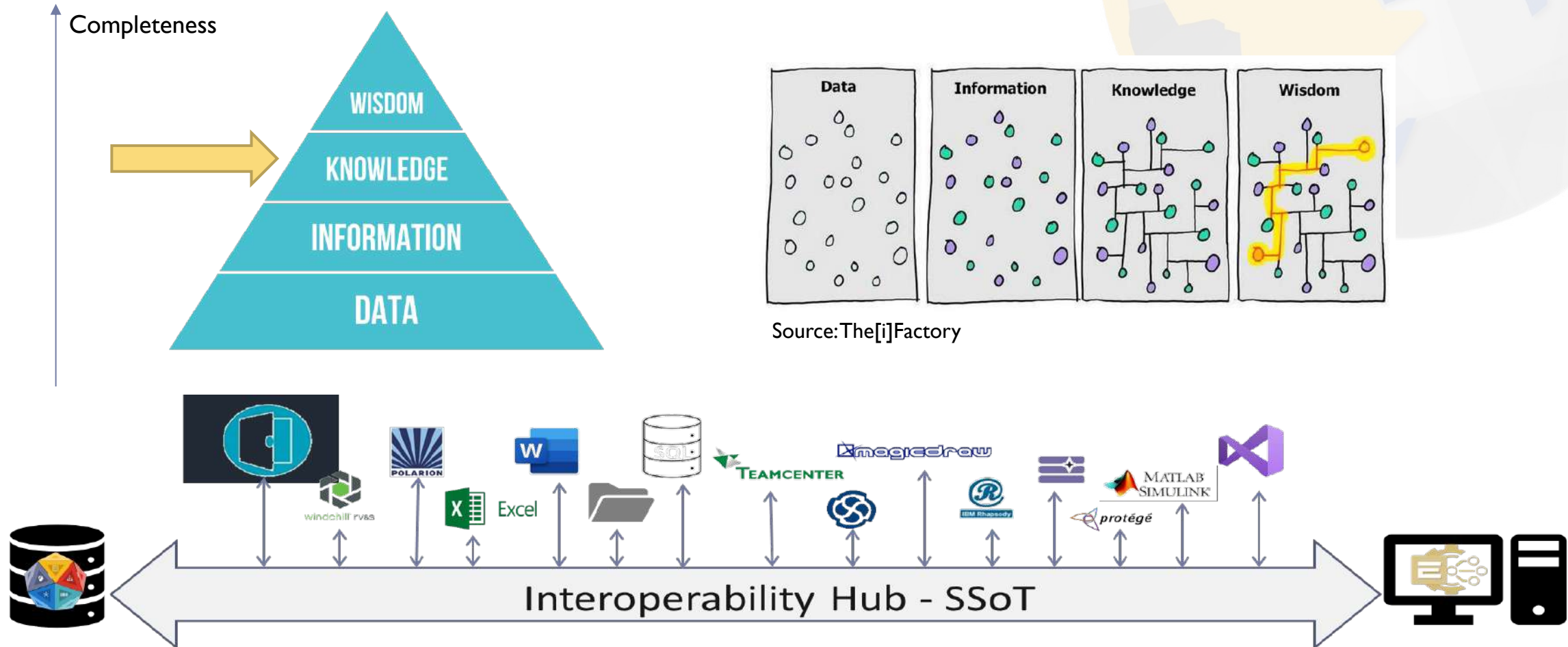


➤ Need for a complete Requirements repository

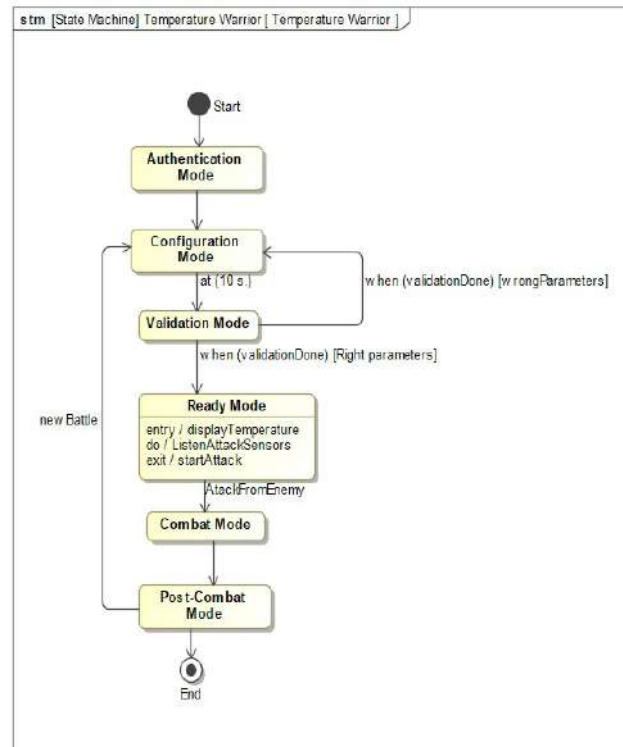
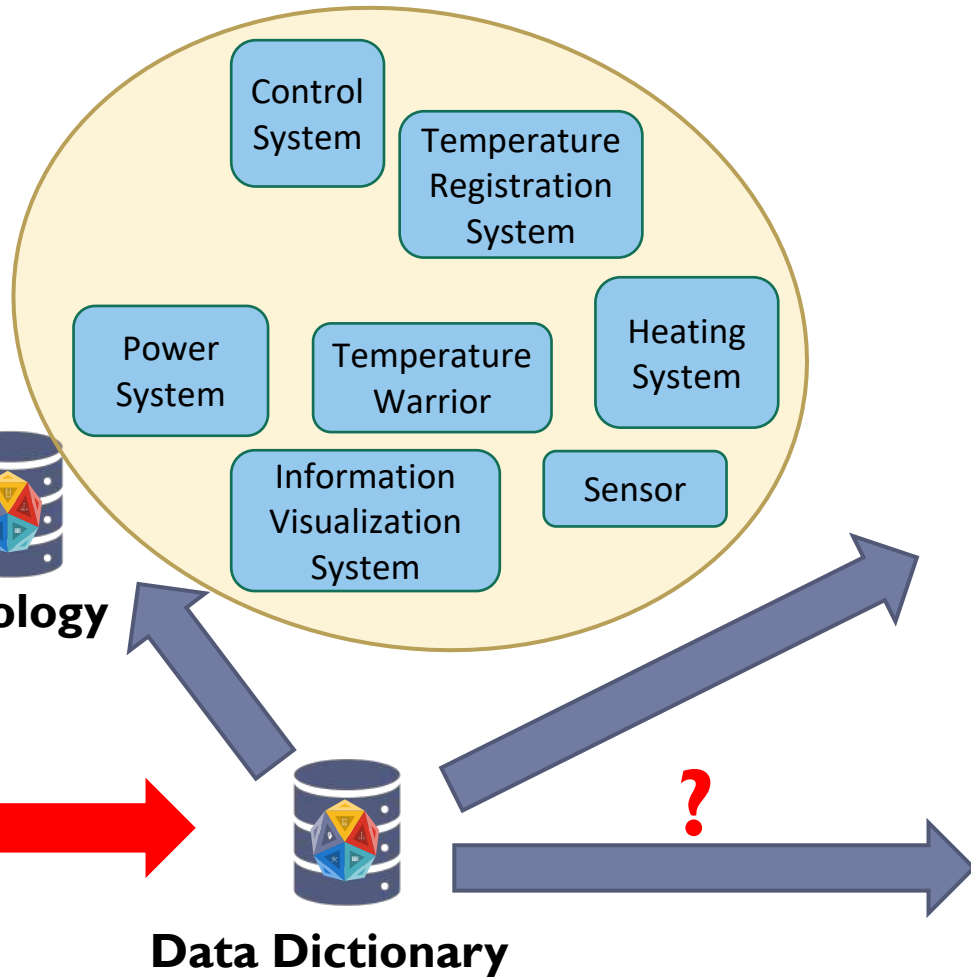


Source: The[i]Factory

➤ Need for a complete...and synchronized Requirements repository



➤ Need for a complete **Data Dictionary**



More info:

**Data Dictionaries**  
 The springboard towards high-quality system requirements

Ilyes Yousfi  
 Senior Key Account Manager  
 The REUSE Company  
 /ilyes.yousfi@reusecompany.com

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


**THE REUSE COMPANY**

➤ INCOSE Guide to Writing Requirements : Needs & Requirements Characteristics

**Characteristics of individual requirements**

**Characteristics of sets of requirements**

<b>C1 - NECESSARY</b>	<b>C10 - COMPLETE</b>
<b>C2 - APPROPRIATE</b>	<b>C11 - CONSISTENT</b>
<b>C3 - UNAMBIGUOUS</b>	<b>C12 - FEASIBLE</b>
<b>C4 - COMPLETE</b>	<b>C13 - COMPREHENSIBLE</b>
<b>C5 - SINGULAR</b>	<b>C14 - ABLE TO BE VALIDATED</b>
<b>C6 - FEASIBLE</b>	<b>C15 - CORRECT</b>
<b>C7 - VERIFIABLE</b>	
<b>C8 - CORRECT</b>	
<b>C9 - CONFORMING</b>	



➤ INCOSE Guide to Writing Requirements : Needs & Requirements Characteristics

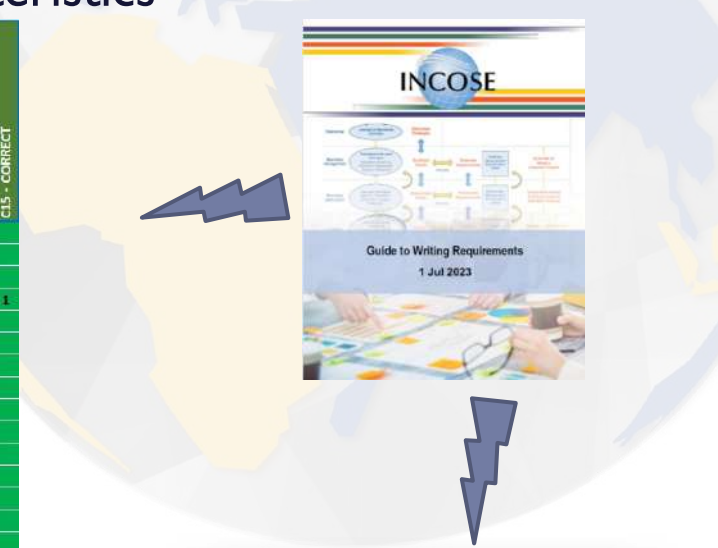
**INCOSE GtWR v4.0**

➤ **15 Characteristics** of individual and sets of needs and requirements.

➤ **42 Rules** to help formulate individual and sets of needs and requirements.

➤ **49 Attributes** are attached to need or requirement statements to form need or requirement expressions.

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	C15 - CORRECT
Accuracy	R01	Sentence Structure		1	1				1	1	1						
	R02	Use Active Voice		1	1	1			1								
	R03	Subject Verb		1	1				1				1		1		
	R04	Use Defined Terms			1				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	1			1	1							
Non Ambiguity	R12	Correct Grammar			1				1	1	1						
	R13	Correct Spelling			1				1								
	R14	Correct Punctuation			1				1								
	R15	Logical Condition			1				1								
	R16	Avoid Not			1				1	1							
Singularity	R17	Oblique			1				1								
	R18	Single Sentence			1		1		1	1					1		
	R19	Avoid Combinators			1		1										
	R20	Avoid Purpose		1													
	R21	Avoid Parentheses			1												
	R22	Enumeration			1		1										
	R23	Context			1	1	1										
Completeness	R24	Avoid Pronouns			1				1								
	R25	Use Of Headings			1												
Realism	R26	Avoid Absolutes					1	1	1				1				
Conditions	R27	Explicit			1				1	1							
	R28	Explicit Lists			1				1								
Uniqueness	R29	Classify									1	1					
Abstraction	R30	Express Once		1							1	1					
Abstraction	R31	Solutionfree		1													
Quantifiers	R32	Universals			1				1	1							
Tolerance	R33	Value Range			1		1	1	1				1				
Quantification	R34	Measurable			1		1	1	1				1				
	R35	Temporal Indefinite			1		1		1								
Uniform Language	R36	Use Consistent Terms			1				1				1	1	1	1	1
	R37	Define Acronyms			1				1			1	1	1	1	1	1
	R38	Avoid Abbreviations							1			1	1	1	1	1	1
	R39	Style Guide				1	1		1			1	1	1	1	1	1
Modularity	R40	Decimal Format			1				1			1					
	R41	Related Requirements			1				1	1	1	1	1	1	1	1	1
	R42	Structured										1	1	1	1	1	1



Att	Attribute	Attributes to Help Define the Requirement and its Intent	Associated with the System of Interest (SOI) Verification	Attributes to Help Maintain and Allow Reuse	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



▶ **Main tips & tricks to reach requirements completeness**

- ▶ Consider all stakeholders: embrace the system context.
- ▶ Consider all the requirement types (Functional, performance, safety, compliance, etc.)
- ▶ Be concise, while asking yourself all the right questions! **How + 5 Wh**
- ▶ It is about what you need the system to do...**and not do!** (sunny day vs. rainy day)
- ▶ Mind the communication with stakeholders



➤ **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.





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

**Technical Requirements –  
Allocation Hierarchically to PBS**

Functional Requirements  
Performance Requirements  
Interface Requirements

**Operational Requirements –  
Drive Functional Requirements**

Mission Timeline Sequence  
Mission Configurations  
Command and Telemetry Strategy

**Reliability Requirements – Project Standards –  
Levied Across Systems**

Mission Environments  
Robustness, Fault Tolerance, Diverse Redundancy  
Verification  
Process and Workmanship

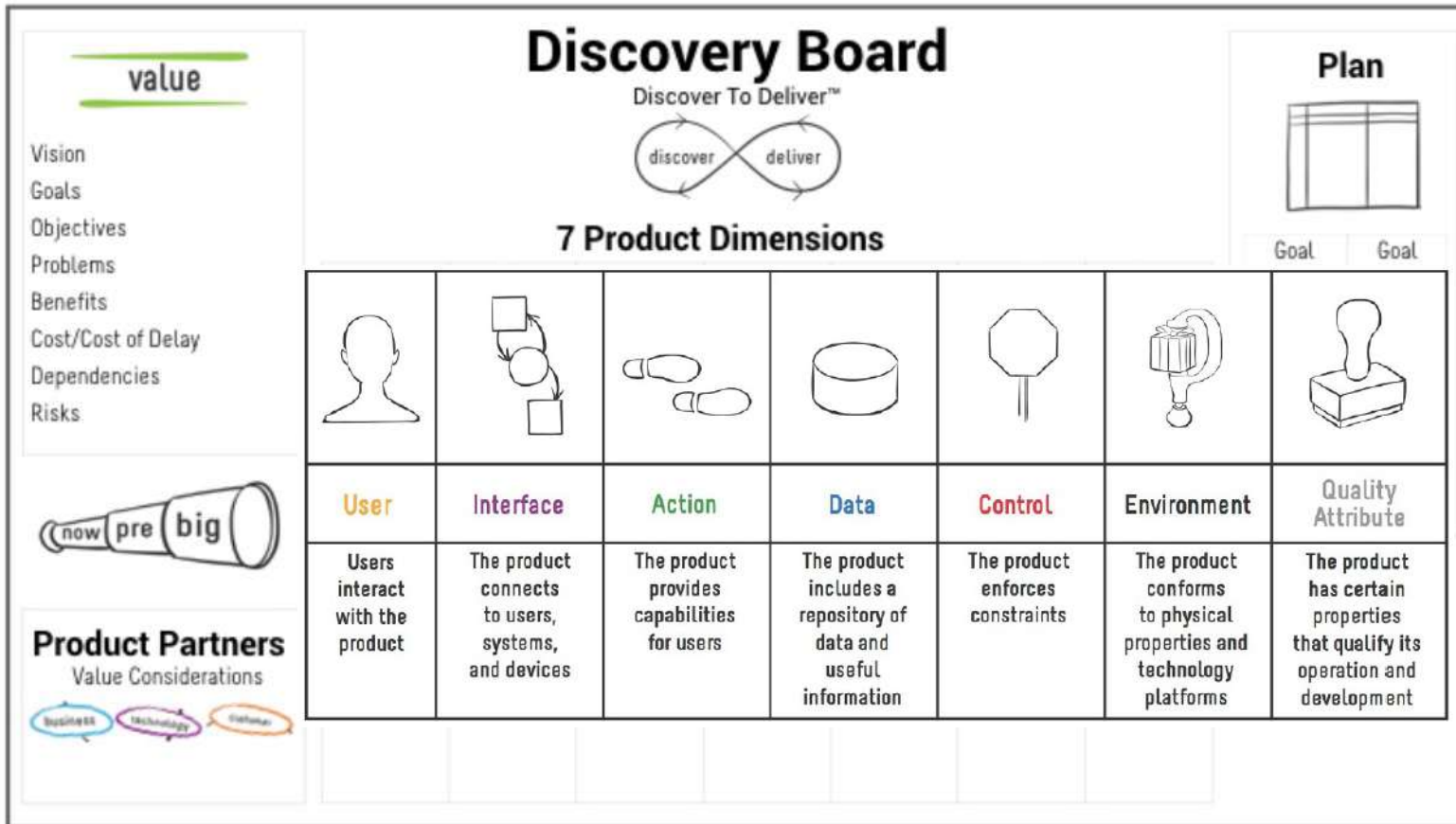
**Safety Requirements – Project Standards –  
Levied Across Systems**

Orbital Debris and Reentry  
Planetary Protection  
Toxic Substances  
Pressurized Vessels  
Radio Frequency Energy  
System Safety  
...

**Specialty Requirements – Project Standards –  
Drive Product Designs**

Producibility  
Maintainability  
Asset Protection  
...

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



Do not forget safety requirements!



- Writing high quality requirements starts with asking the proper questions:
  - The **What** vs the **How**
  - But also and mainly the **Why**
  
- What: Sol, SoS, entities and interfaces
- Who: stakeholders
- What: functions
- How much/many: performance
- While: state or mode
- When: trigger



➤ Avoid prevalence of 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**

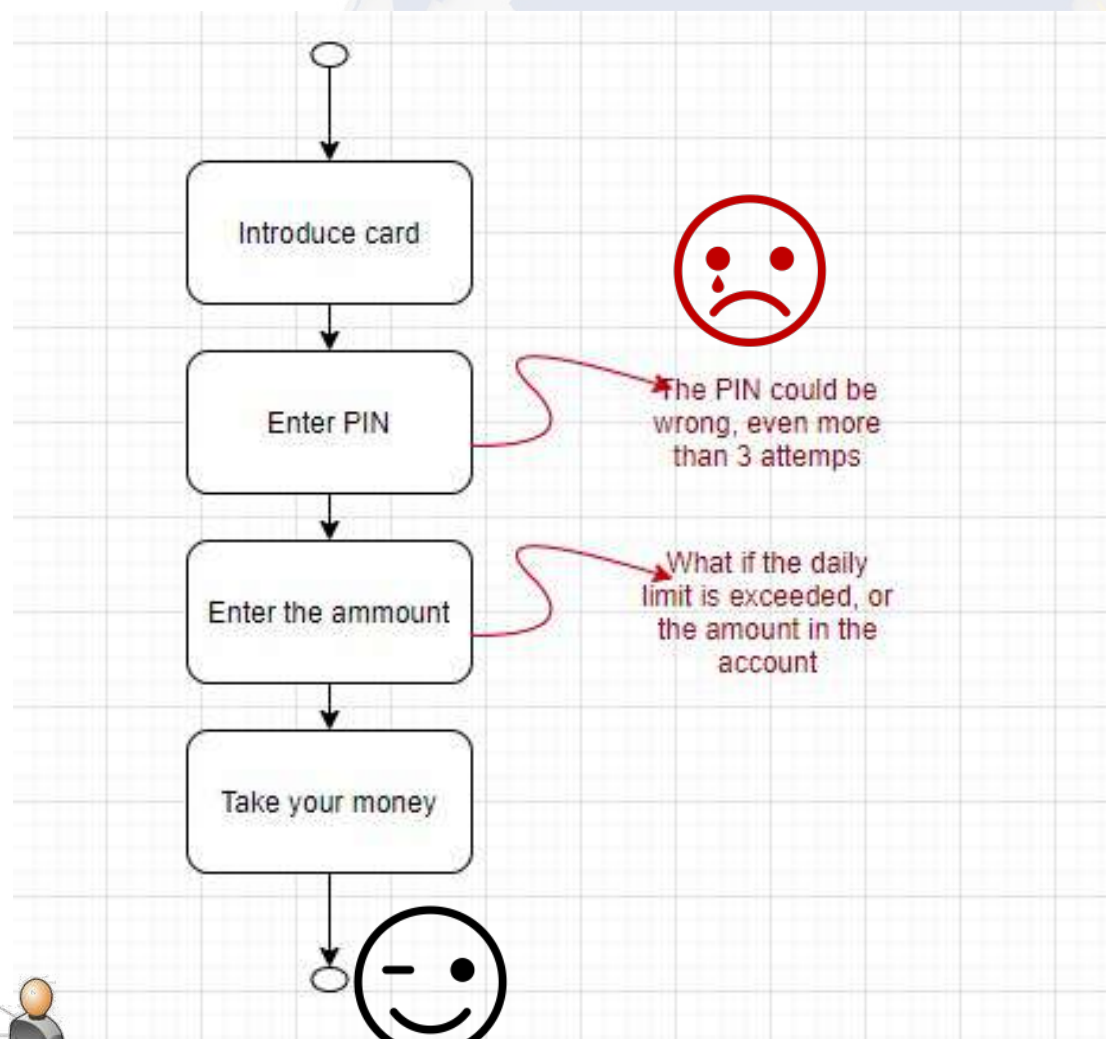
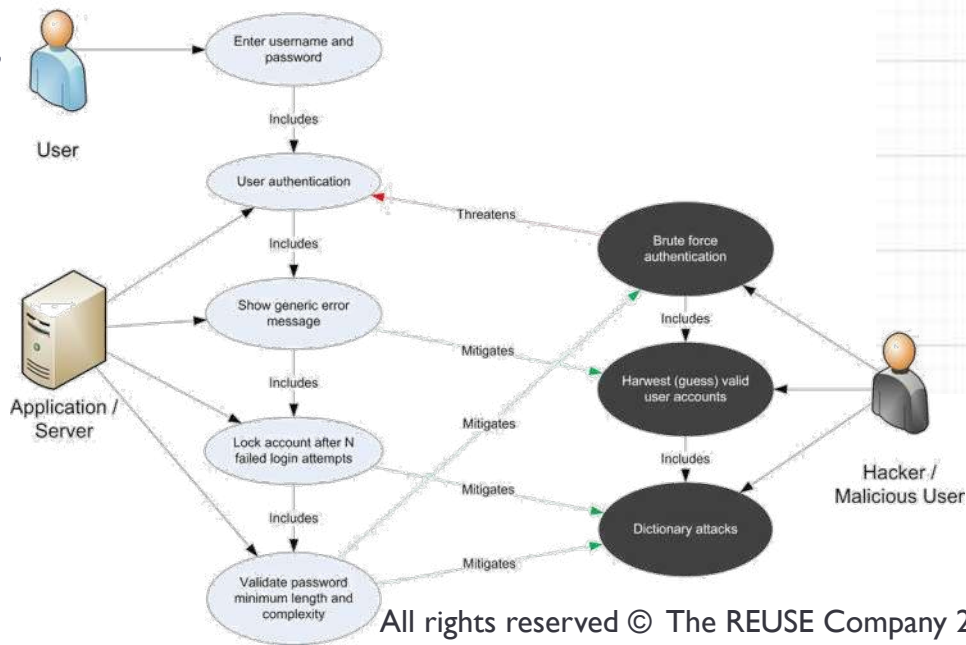


- When: system life cycle



- 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





- 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 & 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
- › 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



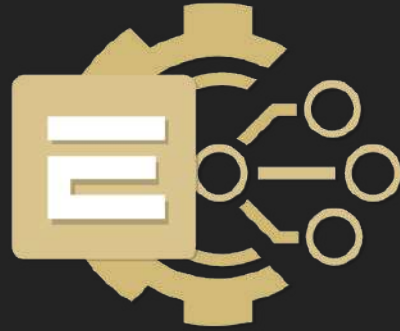
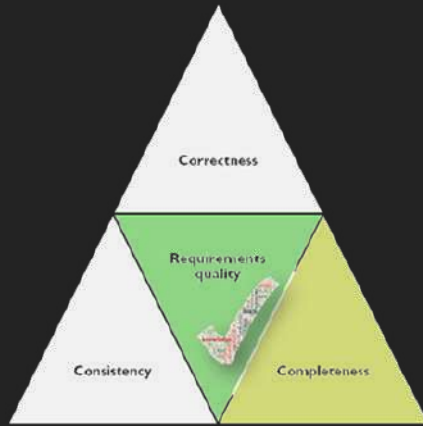
# PROJECT SUCCESS RATES

## AGILE VS WATERFALL



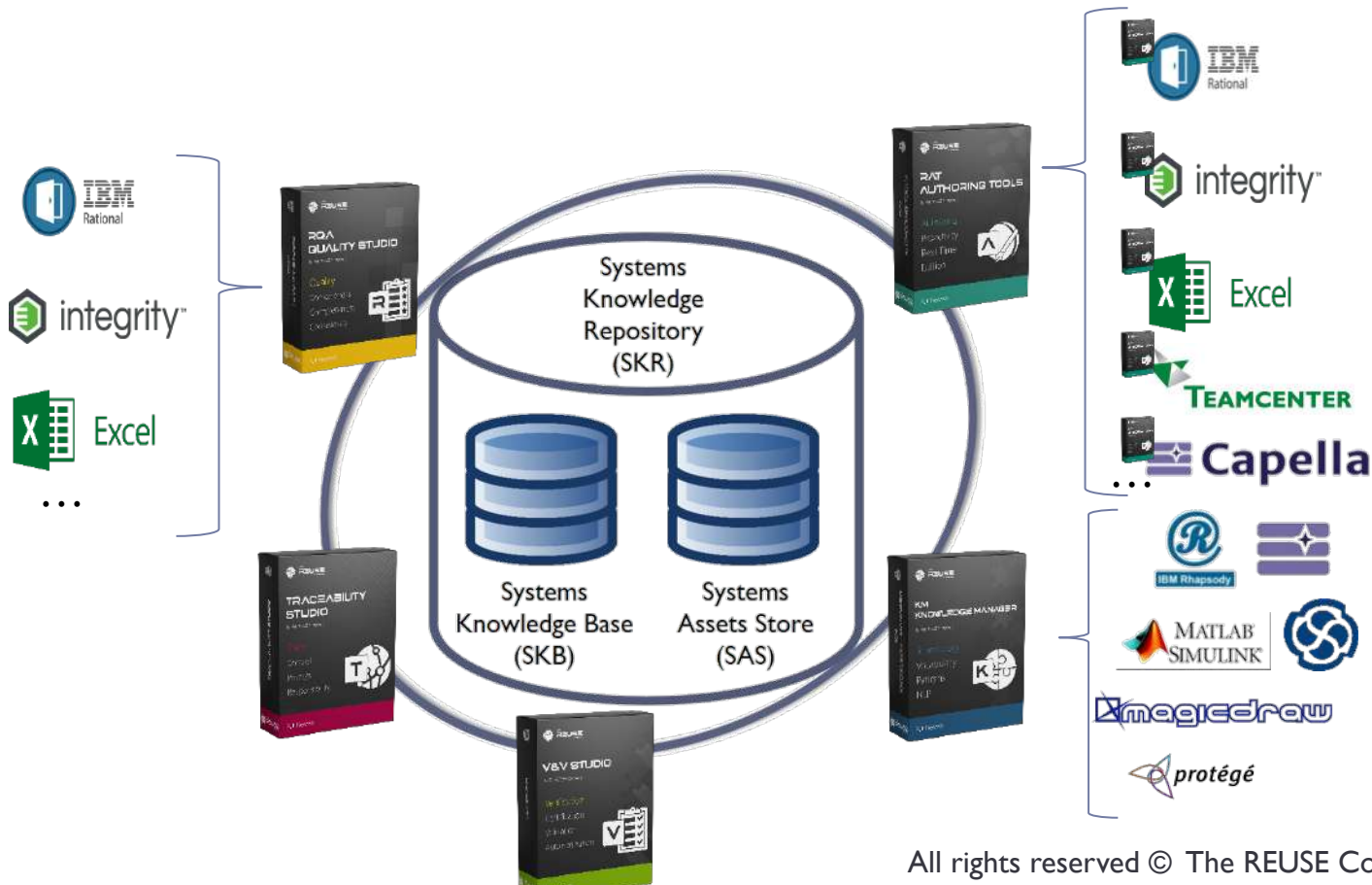
WWW.VITALITYCHICAGO.COM

Source: Standish Group Report 2020



# SES 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)

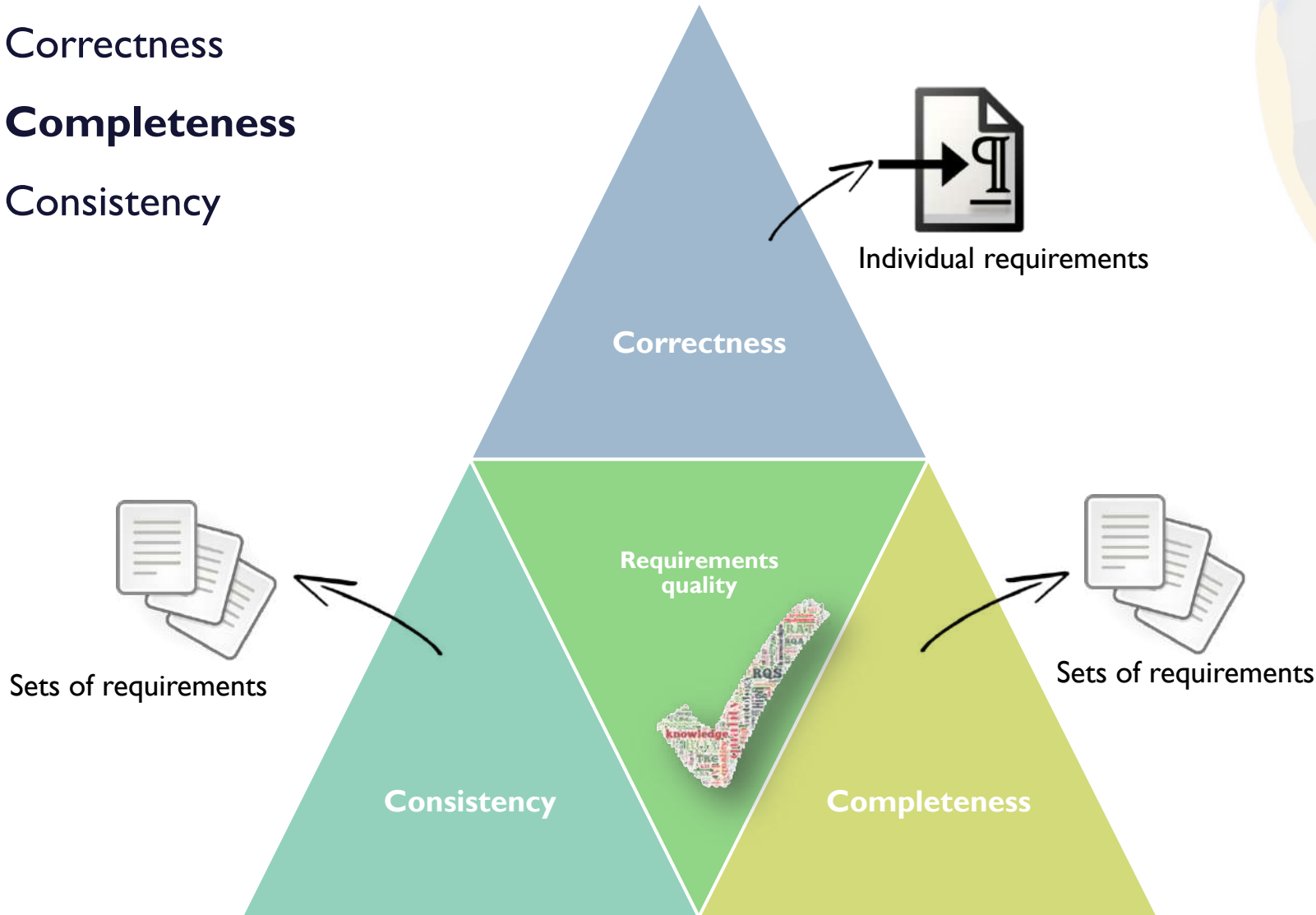


- **RQA – QUALITY Studio** : setup, check and manage the quality of a requirements specification
- **Requirements Authoring Tool (RAT)**: guide authors while they create/edit requirements
- **KM – KNOWLEDGE Manager (KM)**: manage your project knowledge base to support requirements specifications: dictionaries, glossaries, concept maps, knowledge models, ontologies, patterns...



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

- Correctness
- **Completeness**
- Consistency



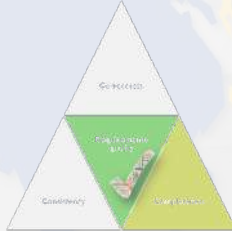


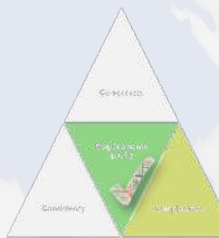
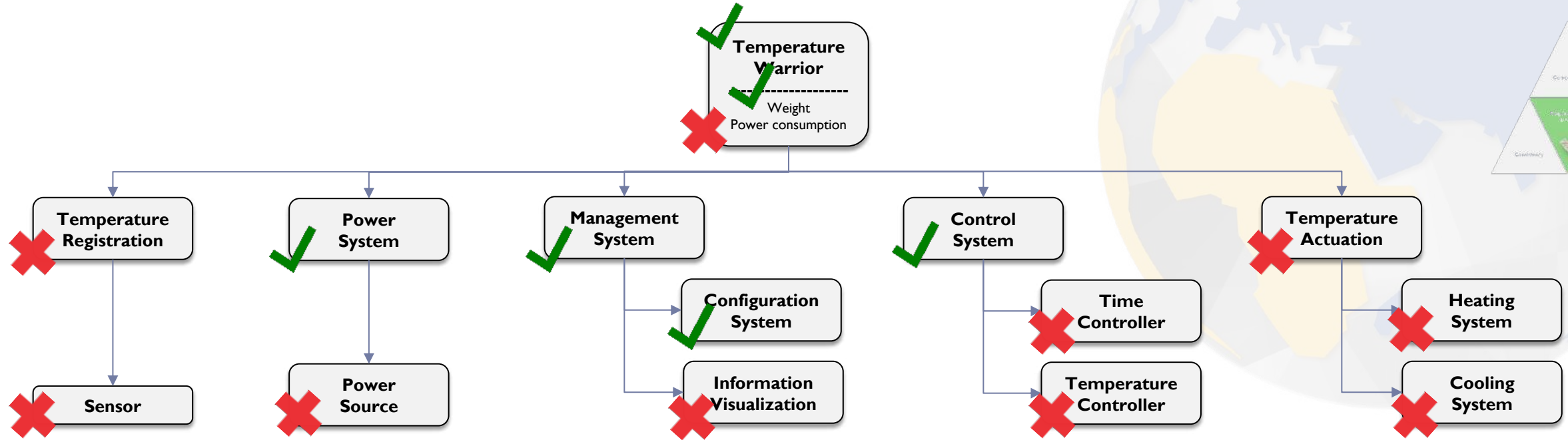
# **Completeness of Sets of requirements**



➤ **Completeness at specification/project level:**

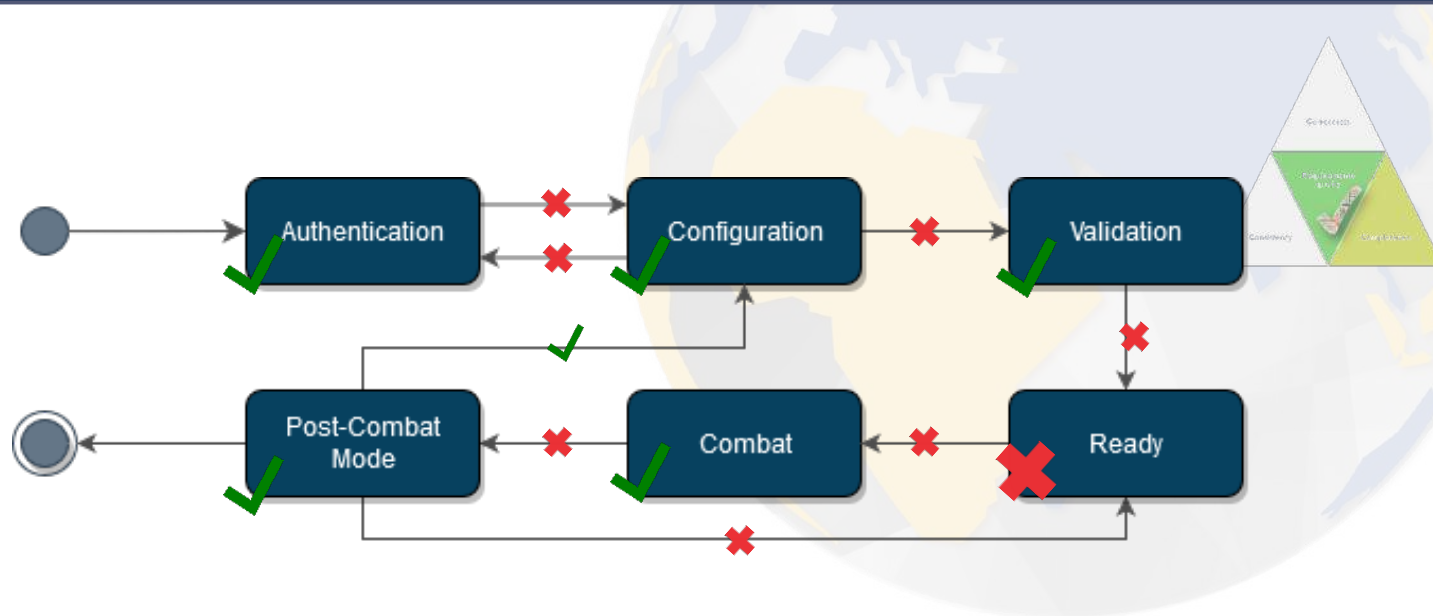
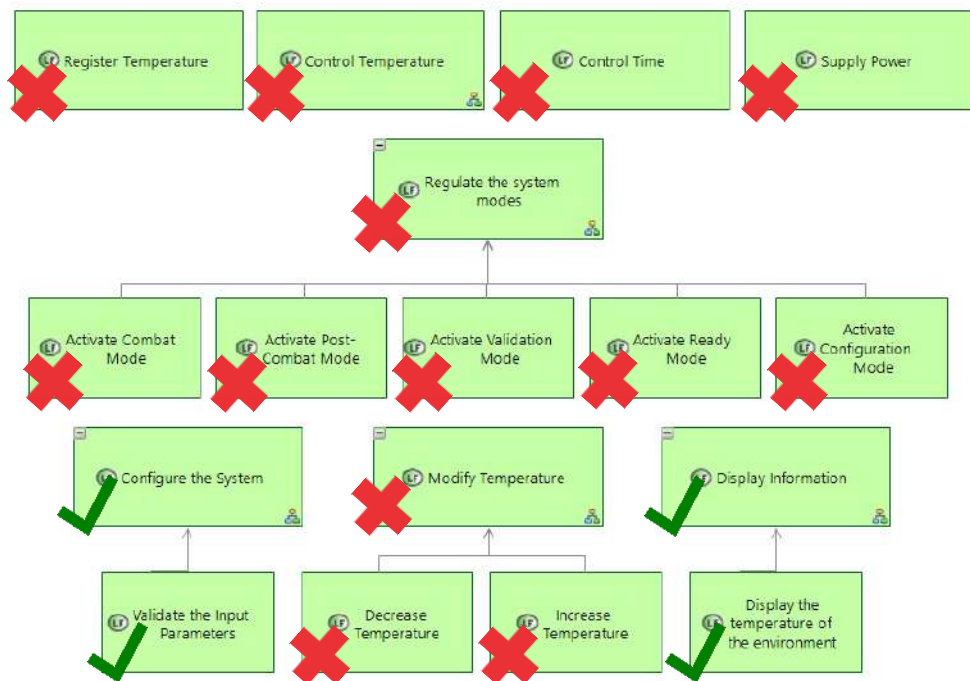
- ✓ Are all the expected requirements types involved in your specifications?
- ✓ Are all the key concepts (from the ontology or from other models, e.g. blocks, states, signals, properties...) properly covered?
  - Does it include requirements for all the system elements according to a block diagram (architecture)?
  - Does it include requirements describing the behavior of the system elements in any of their expected states and modes?
  - Does it include requirements mentioning all the signals?
- ✓ Are all the properties stated for every system element?





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.





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 of 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

**While**

**<state>**

**when**

**[Condition]**

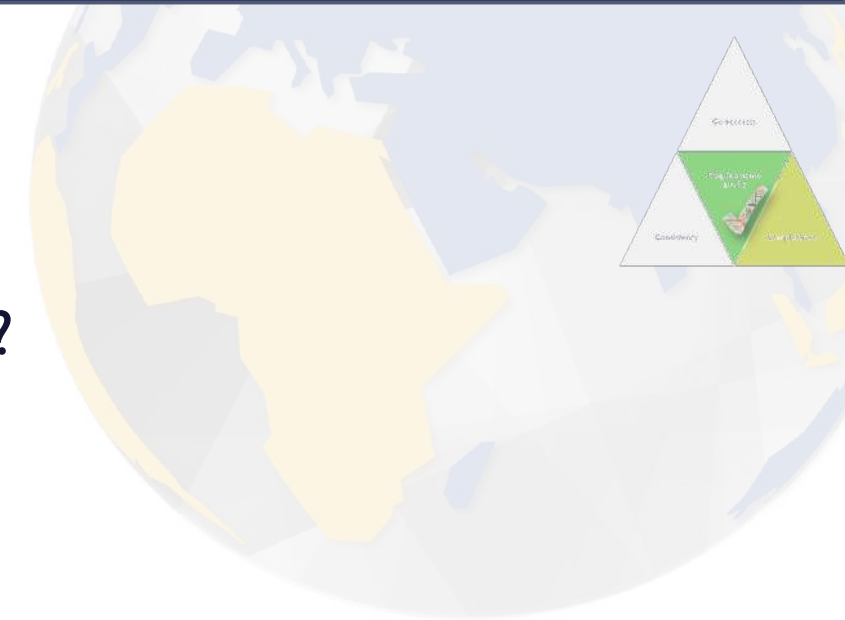
**<Subject>**

**Shall**

**<Action>**

**<Object>**

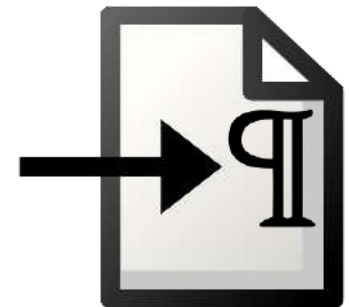
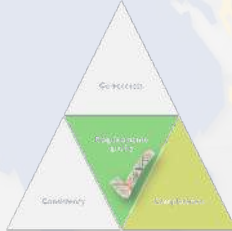
**[Constraint]**





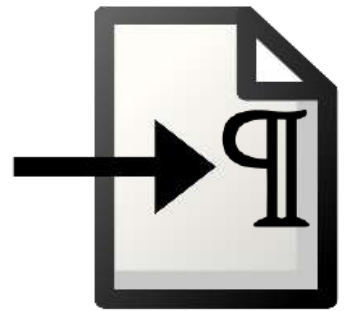
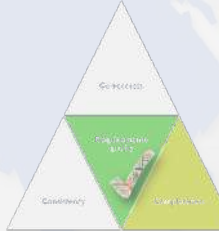
## ➤ Completeness at requirement level:

- 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.  $12V \pm 0.5V$ 
  - “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
    - ...



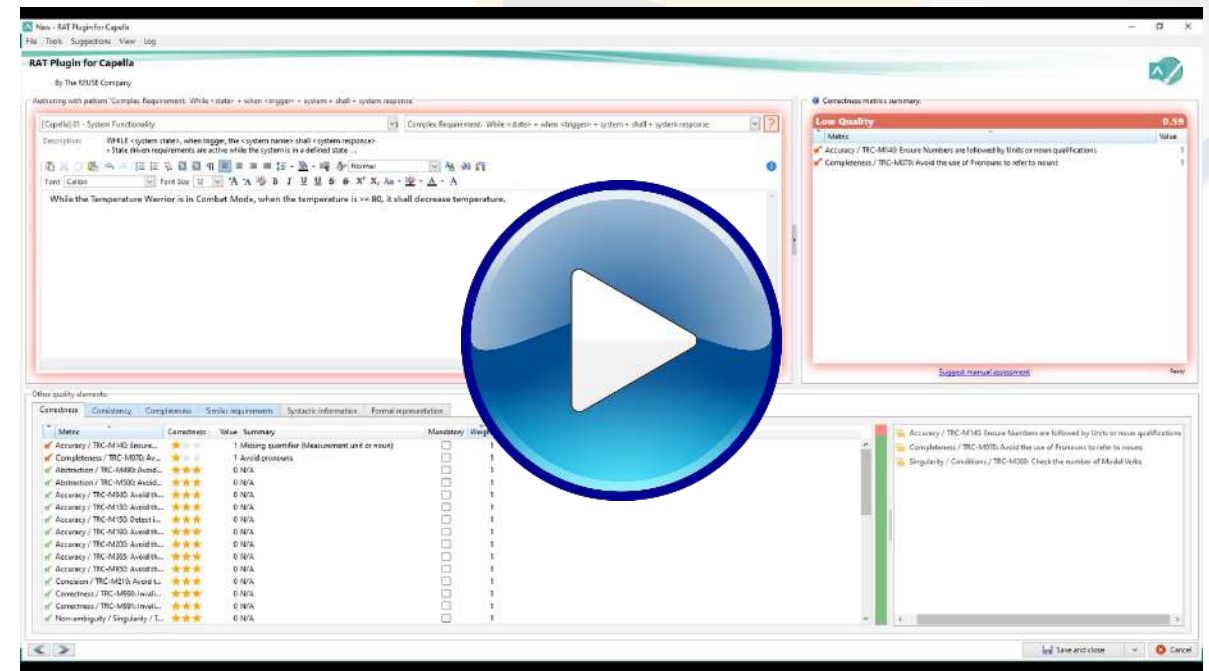


**Requirements  
Completeness  
Assessment  
Demo**

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

➤ **Steps:**

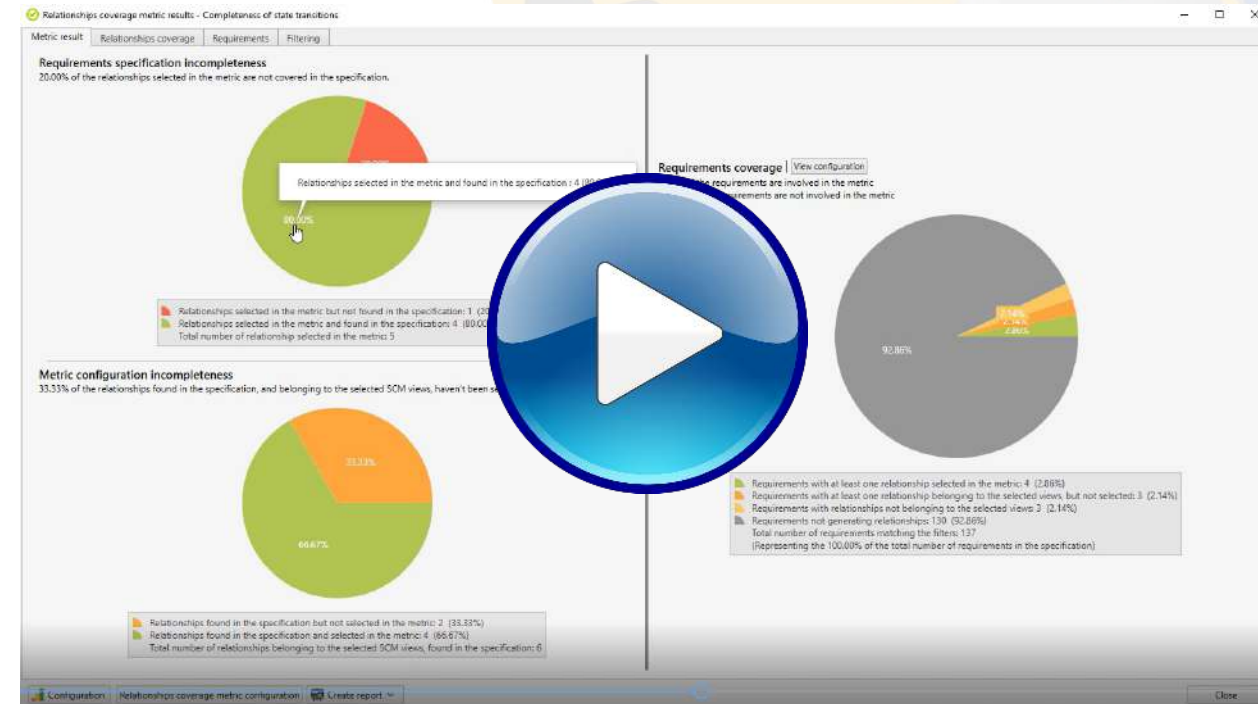
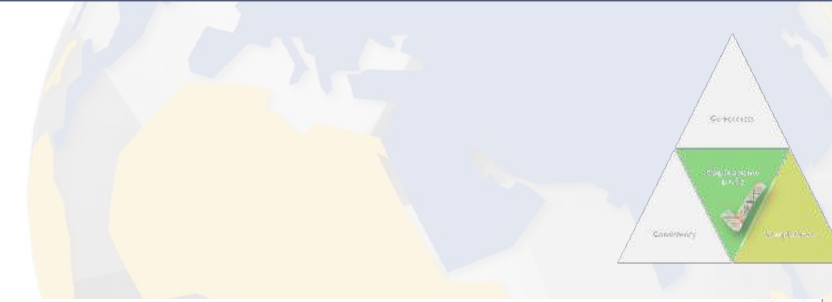
1. Open RAT – **AUTHORING** Tools as an add-on in Capella
2. Choose a **pattern** based on the type of desired req
3. Fill out the individual elements based on the structure of the pattern
4. What if the structure of the pattern is not met
5. What if the requirement includes all the mandatory blocks



➤ **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

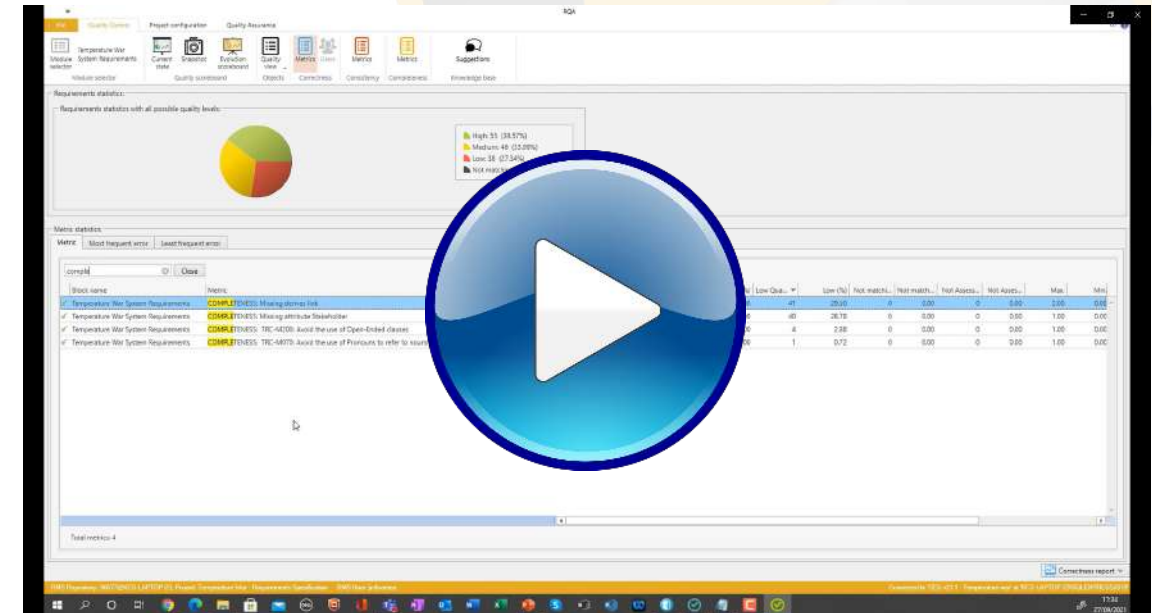




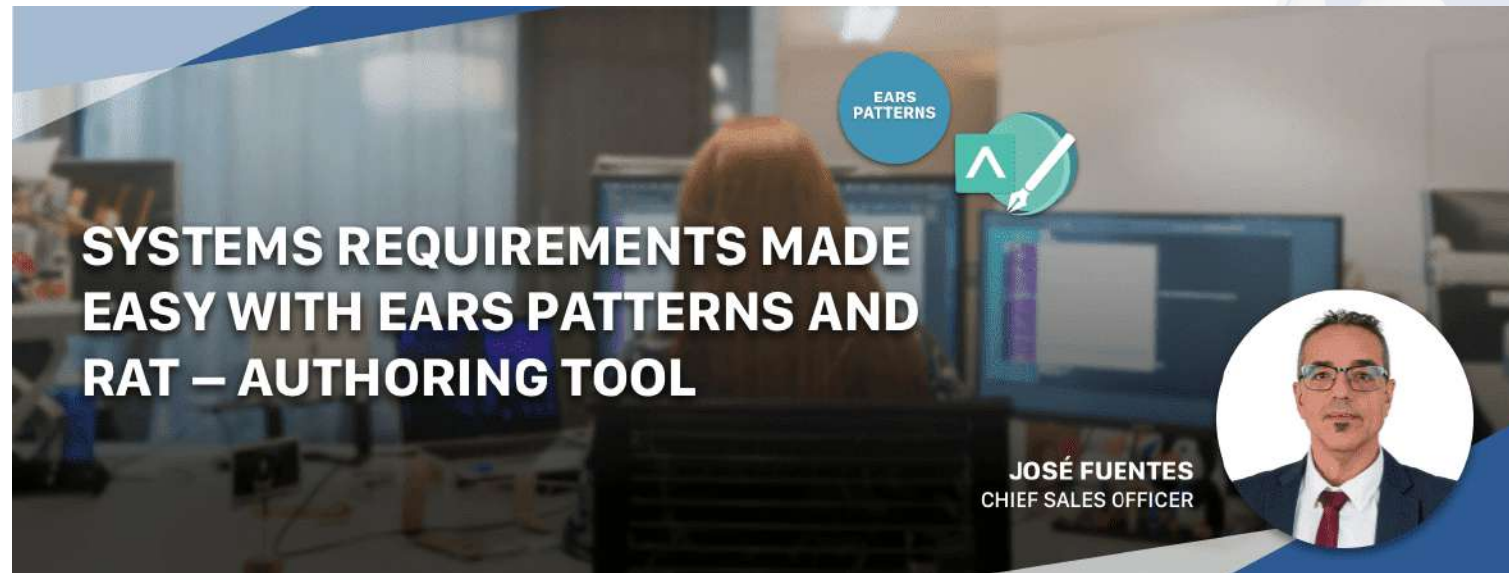
➤ Use case #3: Other completeness checks in RQA

➤ 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







- Following agreed-upon patterns improves the overall quality of textual requirements
- EARS patterns implemented on top of your own requirements management tool
- RAT – AUTHORIZING Tool connects patterns and models
- Improved consistency of requirements
  
- **Dates: May 14 & 16, 2024**





> [www.reusecompany.com](http://www.reusecompany.com)

- > Information about the Systems Engineering Suite
- > Resources (Webinars, handbooks)
- > Request for demo
- > Support Forum

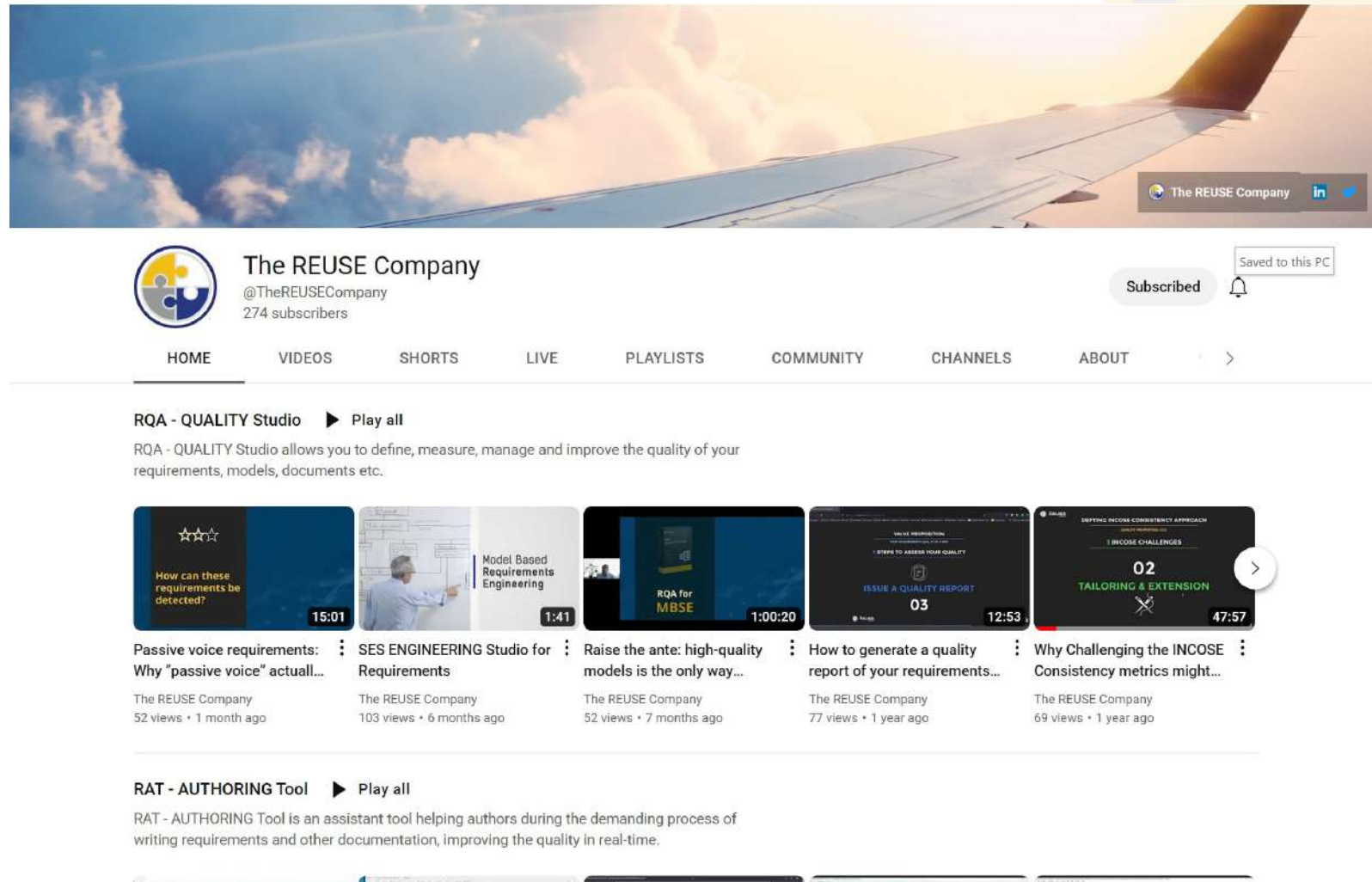
Enabling SMART Systems Engineering

Products Services Resources TRC Forum Support Company Contact

<p>Requirements management through ALLG Contracts</p>	<p>Connecting the Dots: Interoperability between your favourite Systems Engineering tools</p>	<p>Semantic traceability: how to keep the digital thread all along the SE lifecycle</p>	<p>The MBSE Podcast Trust us we are Systems Engineers MBSE around the world: Spain with Juan Llorens</p>
<p>Passive voice requirements: Why "passive voice" actually can become a nightmare</p>	<p>(In Spanish) Invitados al podcast 'Sistemistas': V&amp;V ¿Qué es qué?</p>	<p>Connecting textual requirements and Capella models (Invited presenters)</p>	<p>Requirements Management: Managing data over entire life cycles</p>
<p>How to kick off your KM – KNOWLEDGE Management project</p>	<p>Taming the System Engineering Life cycle using Connectivity and Interoperability: the SES ENGINEERING Studio</p>	<p>Raise the ante: high-quality models is the only way forward after high-quality requirements</p>	<p>Digitalizing the V&amp;V process on both sides of the V-Model</p>

➤ Subscribe to our Youtube channel to follow our latest content!

<https://www.youtube.com/user/TheREUSECompany>



The REUSE Company  
@TheREUSECompany  
274 subscribers

Subscribed

HOME VIDEOS SHORTS LIVE PLAYLISTS COMMUNITY CHANNELS ABOUT

**RQA - QUALITY Studio** ▶ Play all

RQA - QUALITY Studio allows you to define, measure, manage and improve the quality of your requirements, models, documents etc.

- How can these requirements be detected? 15:01
- Model Based Requirements Engineering 1:41
- RQA for MBSE 1:00:20
- ISSUE A QUALITY REPORT 03 12:53
- 02 TAILORING & EXTENSION 47:57

Passive voice requirements: Why "passive voice" actual... The REUSE Company 52 views • 1 month ago

SES ENGINEERING Studio for Requirements The REUSE Company 103 views • 6 months ago

Raise the ante: high-quality models is the only way... The REUSE Company 52 views • 7 months ago

How to generate a quality report of your requirements... The REUSE Company 77 views • 1 year ago

Why Challenging the INCOSE Consistency metrics might... The REUSE Company 69 views • 1 year ago

**RAT - AUTHORING Tool** ▶ Play all

RAT - AUTHORING Tool is an assistant tool helping authors during the demanding process of writing requirements and other documentation, improving the quality in real-time.



Ilyes Yousfi



ilyes.yousfi@reusecompany.com



+34 627 08 66 01



@ReuseCompany



<https://www.linkedin.com/in/ilyesyousfi/en>





THE  
**REUSE**  
COMPANY

