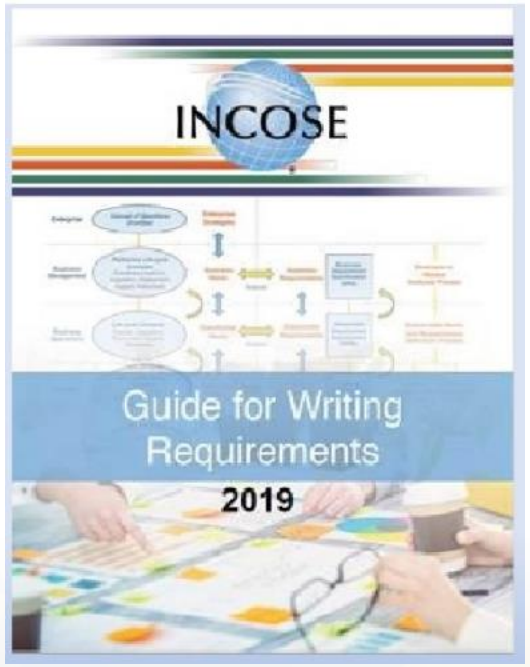


TRC WEBINARS 2020

INCOSE Guide for Writing Requirements: Real-Time Quality Assessment of the INCOSE Rules

Tuesday, 24 March, 2020

Presenters

José M. Fuentes

- Chief Operating Manager
- jose.fuentes@reusecompany.com

Cecilia Karlsson

- Marketing & Communication manager
- cecilia.karlsson@reusecompany.com

Table of Contents

Providing a **knowledge centric** approach to leverage system engineering activities in our customers

contact@reusecompany.com

@reusecompany

Calle Margarita Salas, 16 2-D
28919 – Leganés (Madrid)
SPAIN

+34 912 172 596

About The REUSE Company (TRC)

01 The company was created in **1999**

As a spin-off of a local university in Madrid (Spain)

02 **System + Software Engineers**

Smart combination between Company staff and R&D from Academia

03 **Head Quarters:** Madrid (Spain)

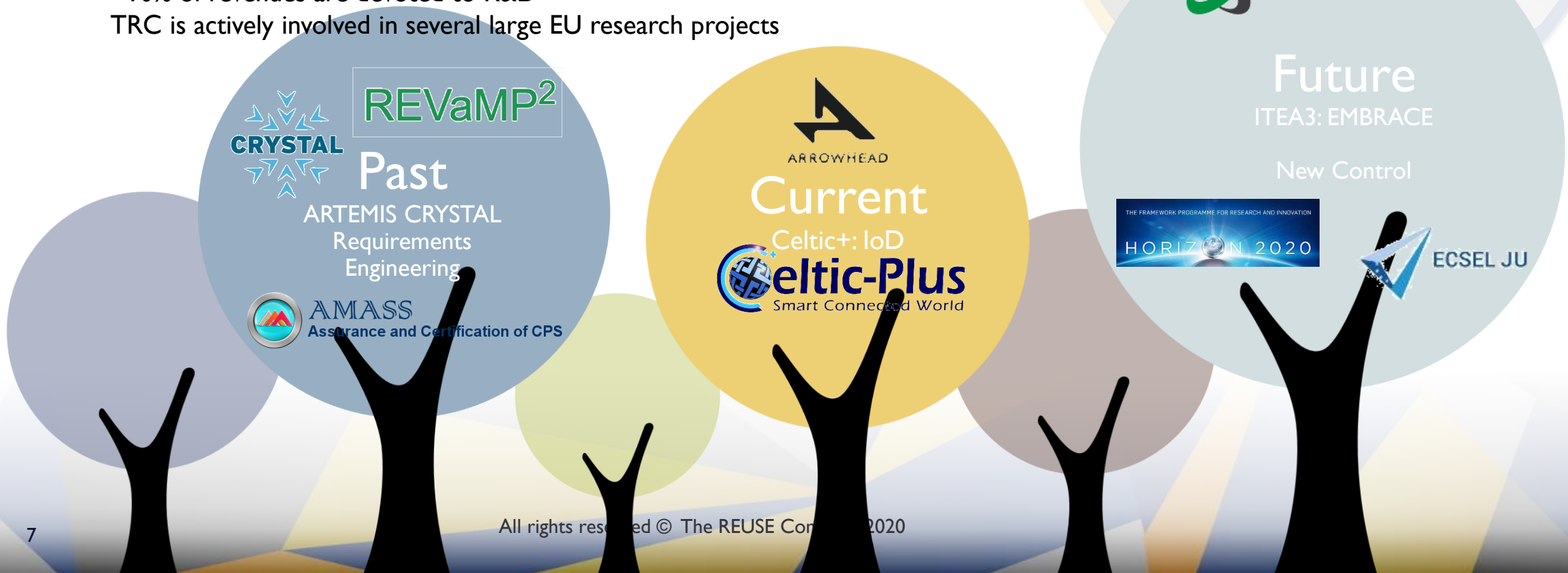
International offices: London (UK) Stockholm (Sweden)

04 Offering a **knowledge centric** approach to leverage system engineering activities in our customers

Research and innovation in our DNA. Public projects

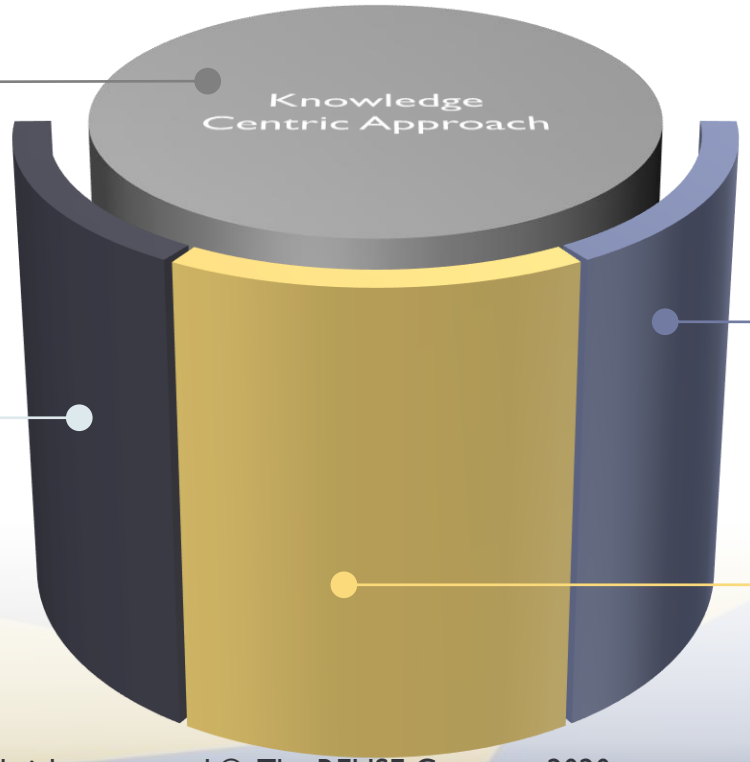
Research and Innovation in our DNA

- Spin-off of Carlos III University of Madrid
- TRC's headquarter is in the Legatec Technology Park of the University
- ≈10% of revenues are devoted to R&D
- TRC is actively involved in several large EU research projects



T(he) R(euse) C(ompany)

Leveraging Systems Engineering activities

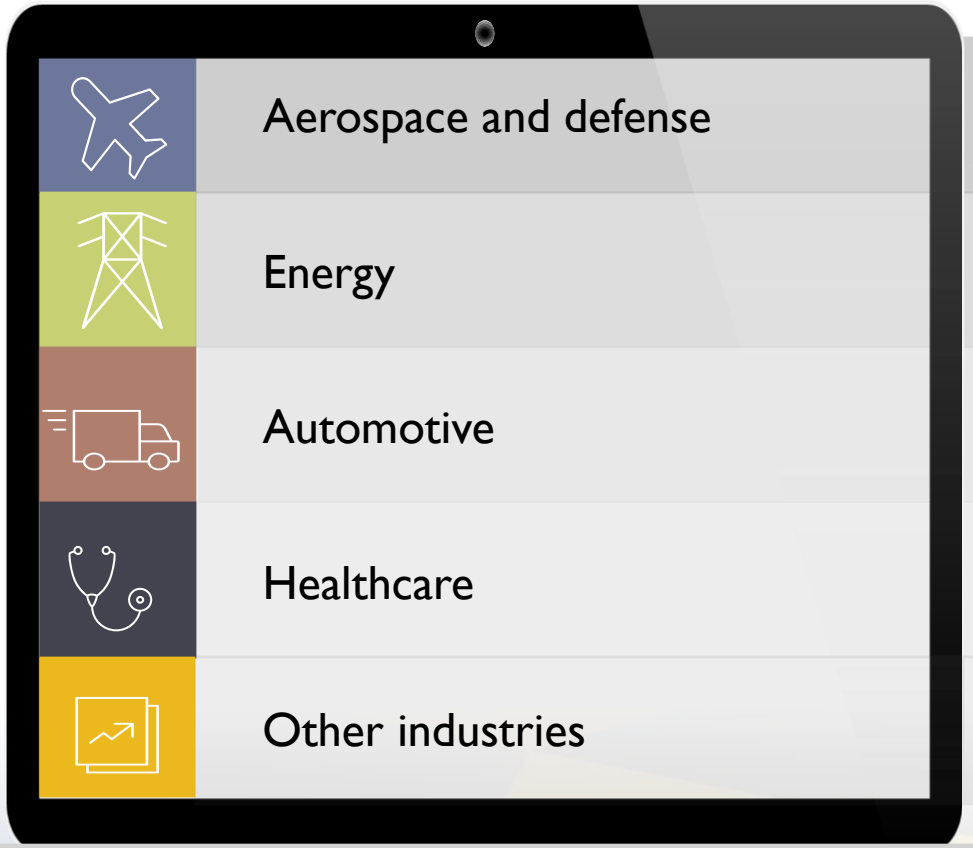


TRACEABILITY

QUALITY

REUSABILITY

Who is using our technology?

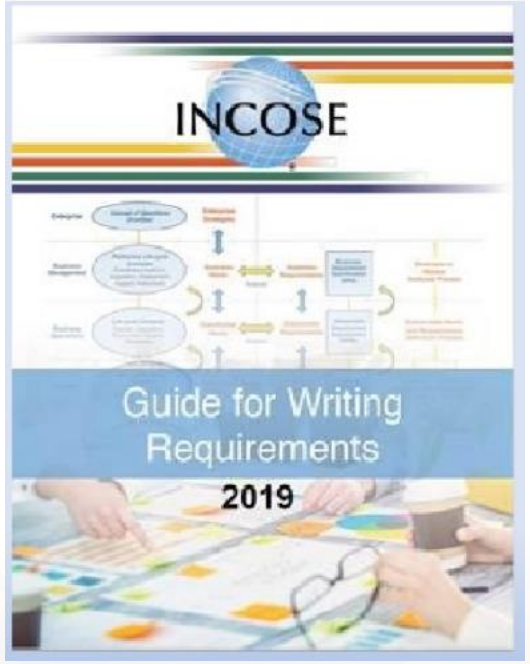


AIRBUS
AIRBUS DEFENCE & SPACE
SAFRAN AIRCRAFT ENGINES
HENSOLDT
AIRBUS GROUP INNOVATIONS
THALES
THALES Communications & Security S.A.S.
aes
arianeGROUP
FUSION FOR ENERGY
iter the way to new energy
EDF
REPSOL
RENAULT
THINK SURGICAL
Health Net
SIEMENS
acciona Agua
orange
rtve
ELRA European Land Registry Association
tirant lo blanch

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
- Member of the board of AEIS – the Spanish chapter of INCOSE
- Active contributor to the INCOSE Guide for Writing Requirements

TRC WEBINARS 2020



TRC

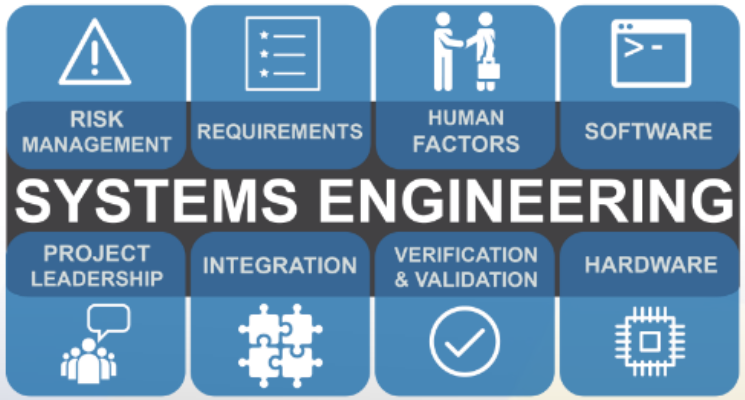
WEBINARS 2020

INCOSE Guide for Writing Requirements: Real-Time Quality Assessment of the INCOSE Rules

Tuesday, 24 March, 2020

WHAT IS INCOSE?

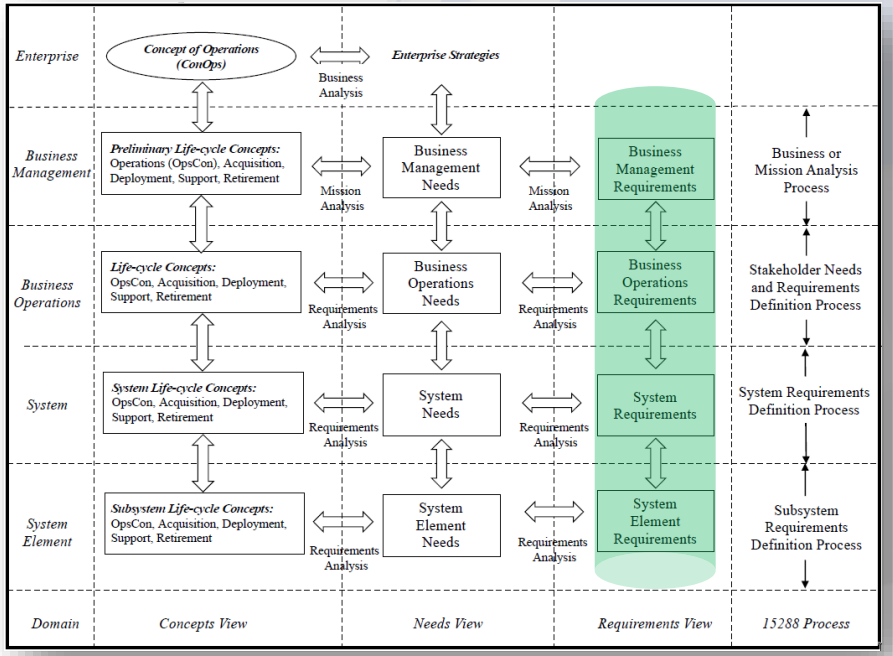
- The **International Council on Systems Engineering (INCOSE)** is a not-for-profit membership organization founded in the 90s to **develop and disseminate the interdisciplinary principles and practices** that enable the realization of successful systems. INCOSE is designed to **connect Systems Engineering professionals** with educational, networking, and career-advancement opportunities in the interest of developing the global community of systems engineers and systems approaches to problems.



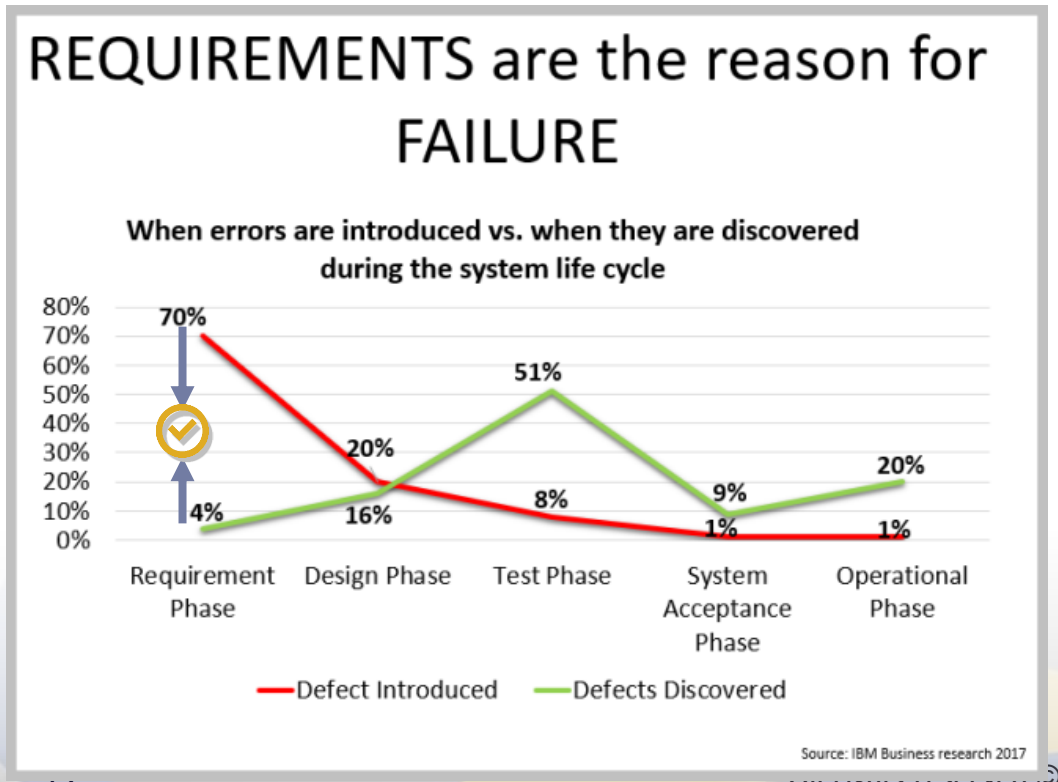
- **Systems Engineering** is an **interdisciplinary approach** and means to enable the realization of **successful systems**. It focusses on defining customer **needs** and required **functionality** early in the development cycle, documenting **requirements**, and then proceeding with **design** synthesis and system **validation** while **considering the complete problem**: operations, cost and schedule, performance, training and support, test, manufacturing, and disposal.

Requirements at the heart of SE process

- *A **requirement statement** is the result of a formal **transformation of one** or more **needs** or parent requirements into an **agreed-to obligation** for an entity to perform some function or possess some quality*



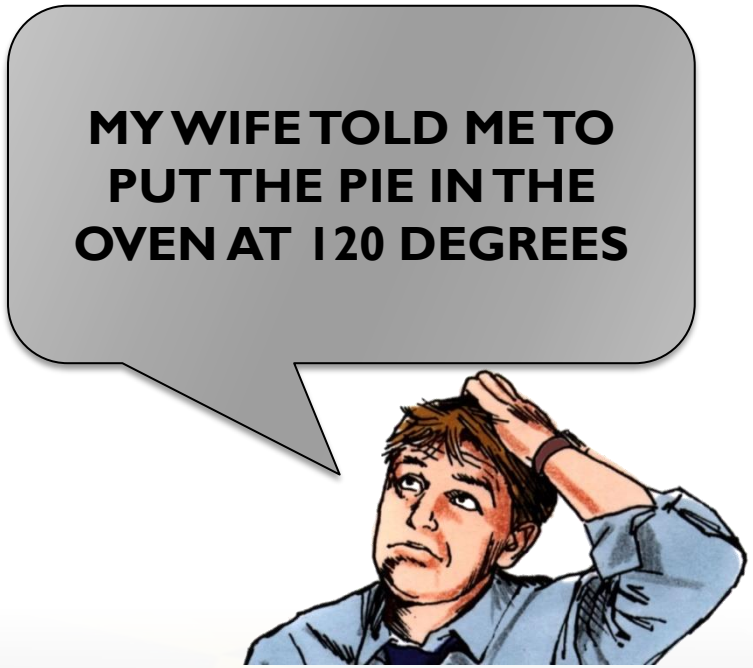
Transformation of concepts into needs into requirements (based on Ryan, 2013).



- *The **textual form of needs and requirements** are not only useful, they are **necessary**. Operational scenarios, use cases, diagrams, and other types of models are also useful and necessary.*
- *Can we measure how correct, how complete, how consistent, how measurable… a specification is??*

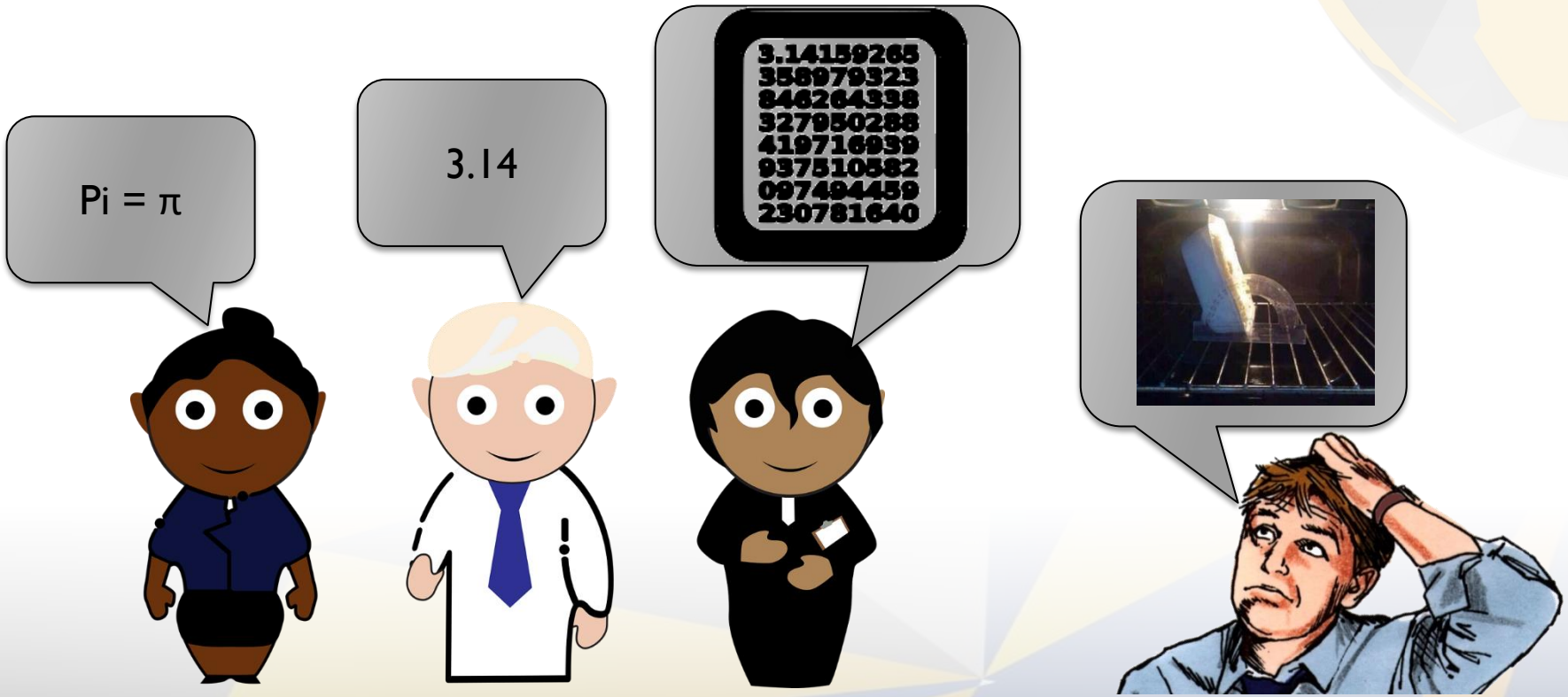
Why focusing on requirements quality

- Because communication is not always that easy:



Why focusing on requirements quality

- Because communication is not always that easy:

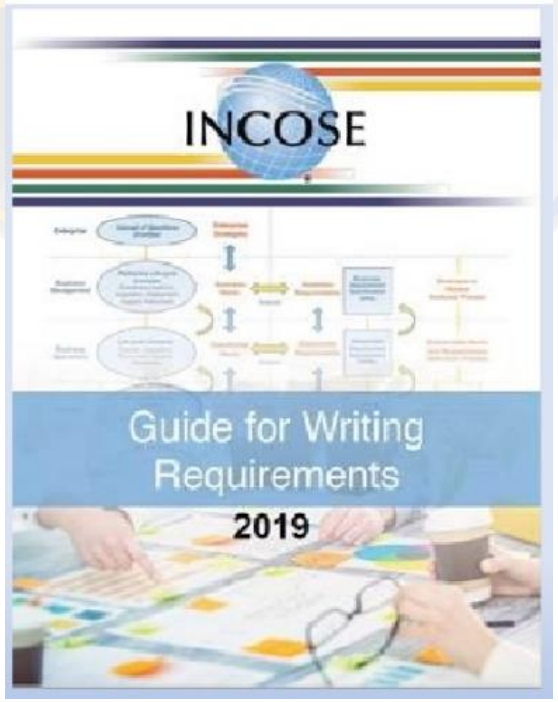


INCOSE Guide for Writing Requirements

The INCOSE RGW (**Requirements Working Group**) in line with its goal

(Expand and promote the body of knowledge of requirements engineering and its benefits within the systems engineering community)

has developed the INCOSE GfWR (**Guide for Writing Requirements**)



The GfWR provides guidance on how to express textual requirements.

The GfWR draw advice into a single, comprehensive set of

characteristics, rules and attributes

for well-formed need and requirement statements.

INCOSE GfWR

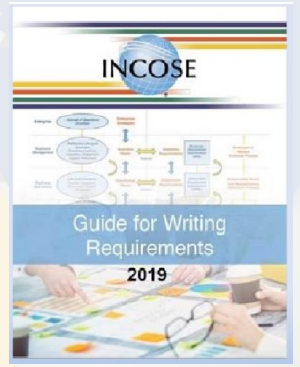
- **Characteristics** of individual and sets of needs and requirements, provides rationale and guidance for helping understand the characteristics.

- **Rules** for individual and sets of needs and requirements that help to formulate them. Included an explanation of the rule and examples of the application of the rule.

- **Attributes** that can be attached to a need or requirement statements to form need or requirement expressions. Also included is guidance on the use of attributes.

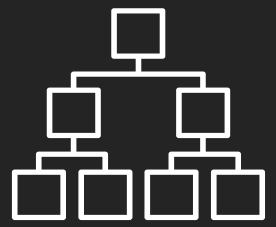
41 Rules / 14 Characteristics

			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									
	R23	Context			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



What is an

Ontology and a knowledge library

What is an Ontology

As **systems become increasingly complex**, the **ability to share and reuse data and information**, including requirements, across organizations both internal and external is **critical to project success**.

An **ontology includes** the formal naming and definition of a set of **terms, entities, data types, and properties** as well as defining the **relationships** between these terms, entities, and data types that are fundamental to the **project and organization** (INCOSE GfWR)

01 Vocabulary/Glossary

Controlled Organizational and Project Vocabulary for a common understanding among stakeholders

02 SCM/Architectures

Capture the system architectures represented in views and models. Stablish relationships among system and system elements, and among other system entities. Classifying information by meaning, nature…

03 Patterns

Representing a set of agreed-upon templates (grammars) to create and maintain consistent textual artifacts

04 Formalization

Representation of assets semantic through SRL – System Representation Language

05 Reasoning

A combination of rules, and actions to infer information from valuable assets and to control the behavioural part of the knowledge library

Example of ontology

Domain specific ⟵ | Common English ⟶

Vocabulary

Aircraft | A380 | A350

System | Operate | Temperature | Environment | Pressure

shall | The | of | at | Lower

Architectures - Conceptual model

Temperature — "Operation Range" → [-60ºC , +60ºC]

System

Aircraft

Fuselage | Wing | Landing gear | A380 | A350

<Operation>

Work

Operate

Environment

Temperature | Pressure

"Greater than (>)"

Patterns

System (*) | Shall | Operation (*) | At | «Minimum» | Environment (*) | Of | NUMBER | MEASUREMENT UNIT

Formalization

The A380 shall be able to operate at a minimum temperature of -70º C → Temperature — "Greater than (>)" → -70 | ºC

Reasoning

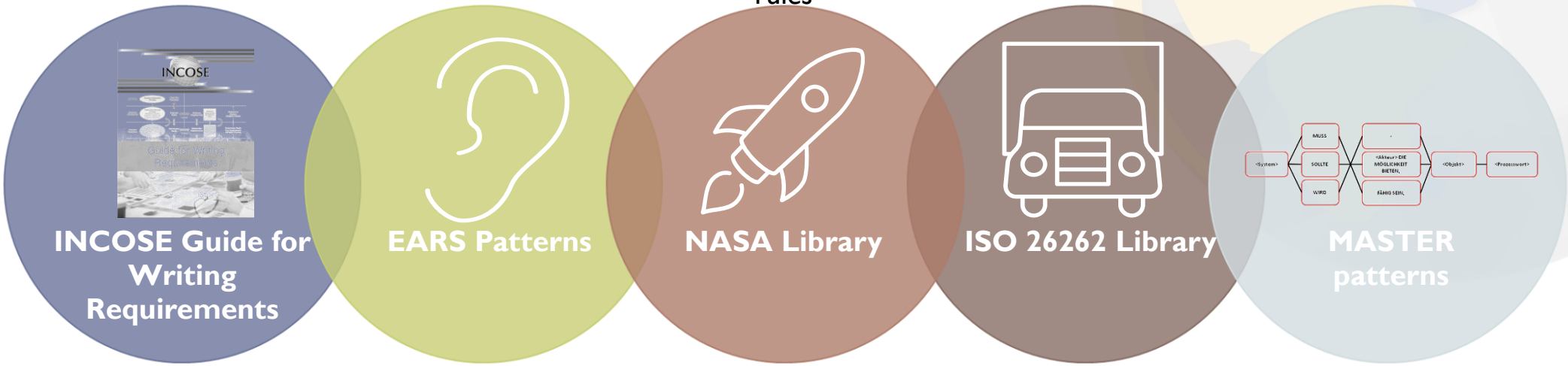
If NUMBER Lower than (<) -60º ºC Or NUMBER Greater than (>) +60º ºC → ✗

What is a Knowledge Library

- A combination of Knowledge items,
  - of different nature,
  - at different levels of abstraction
- Representing a specific business domain or **area of knowledge**
- With the aim of improving the way projects are managed, including:
  - the promotion of the principle: **quality** *right the first time,*
  - enabling semantic search portals to archive and retrieve assets,
  - thus providing tools to **reuse** assets at different level,
  - and reducing **time** to market,
  - improving the way engineers generate (**author**) new assets,
  - enhancing the way items are inspected and **verified,**
  - Enabling real **interoperability** mechanisms and services,
  - reducing **time** to elaborate documents, systems and projects

Knowledge Libraries

NASA
Glossary, patterns and rules



INCOSE
Quality rules for the analysis of textual requirements

EARS
Requirements patterns

ISO 26262
Glossary, patterns and rules

MASTER
Quality rules for requirements and requirements patterns

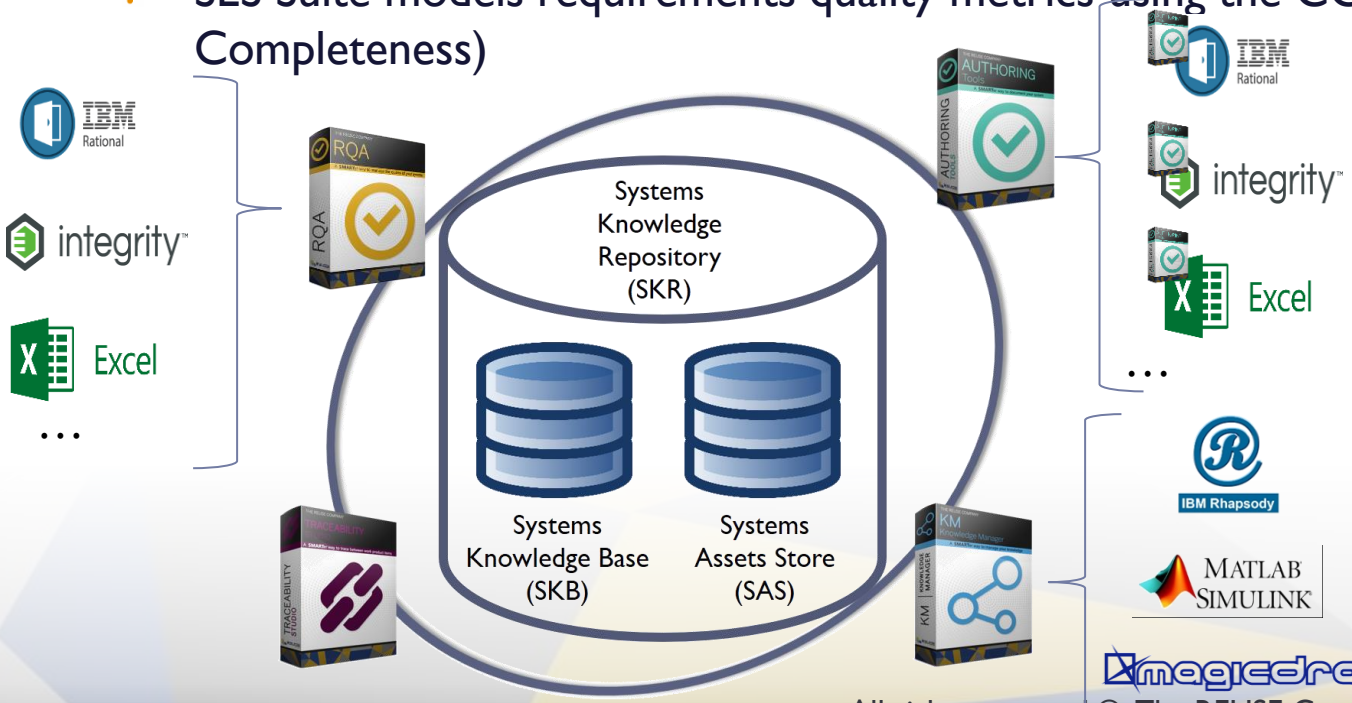
Knowledge Base

Users

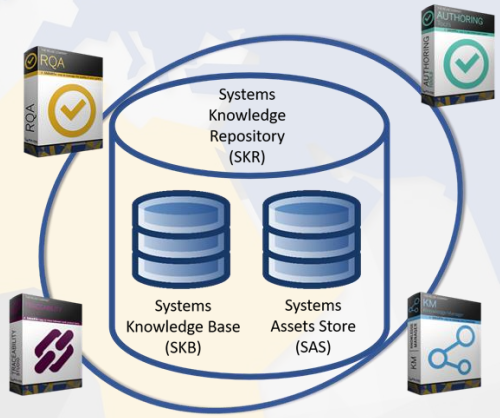

The Systems
Engineering Suite

The TRC Systems Engineering Suite

- The Systems Engineering Suite intends to tackle 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:** 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…



RQA - QUALITY Studio

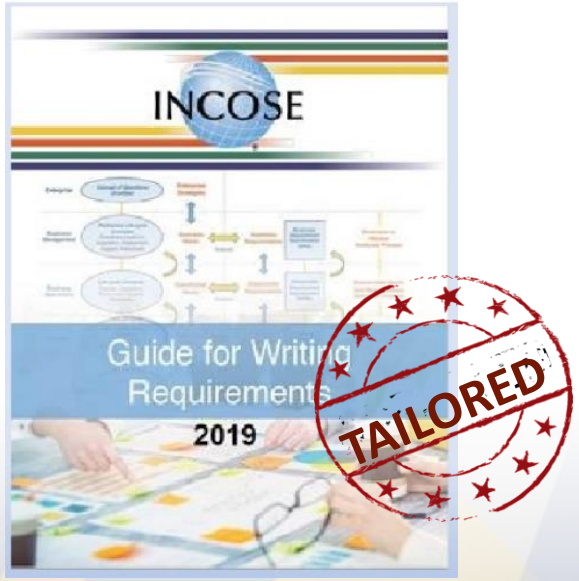
A tool to **automate** the routine **quality inspection** and analysis of different types of engineering items minimizes the cost of quality appraisals, while increasing the consistency and overall quality of the projects.

Quality Analysis of Requirements and all Kind of Engineering Items

RQA covers all the engineering items generated during the systems engineering life cycle. Quality managed not only within **requirements**, but also within logical models (UML or SysML), physical models (MODELICA, Simulink, etc.), 3D models, test cases, FMEA tables... and even textual documents.

Customizable Quality Functions

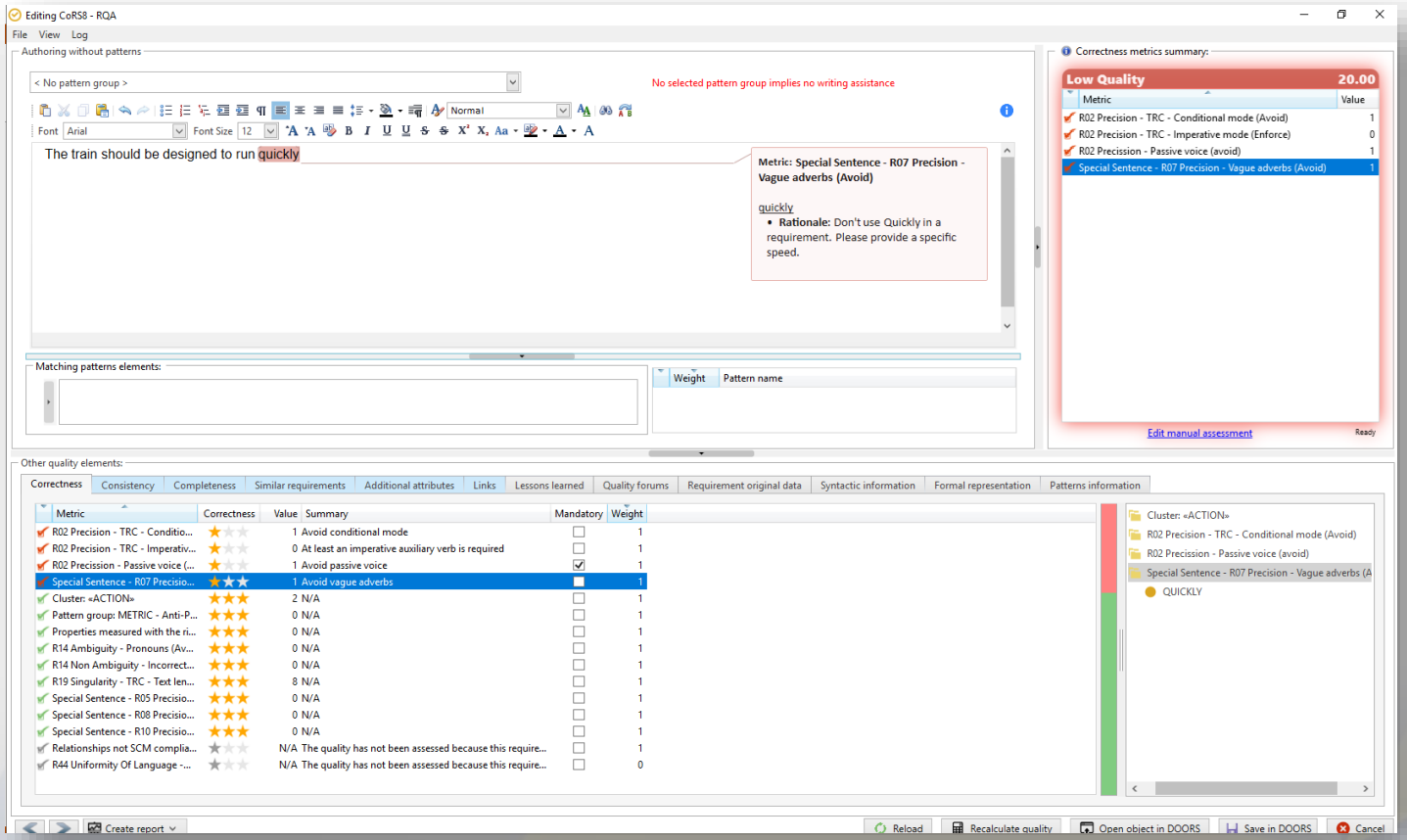
RQA provides tailored analysis and **configurable assessments**, represented in a centralized system quality scoreboard, provide a quick understanding of the current quality status, and quality evolution of a project.



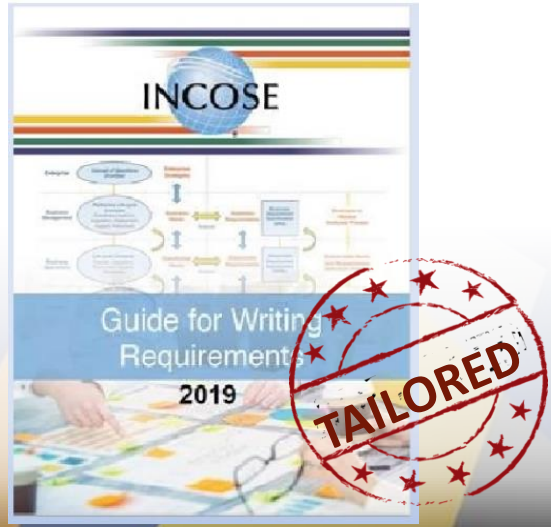
RQA (in combination with **RAT - Authoring Tool**) reduces the defect rate and boost the early detection, thus reducing cost and increasing overall quality.

Authoring with the INCOSE GfWR recommendations and rules

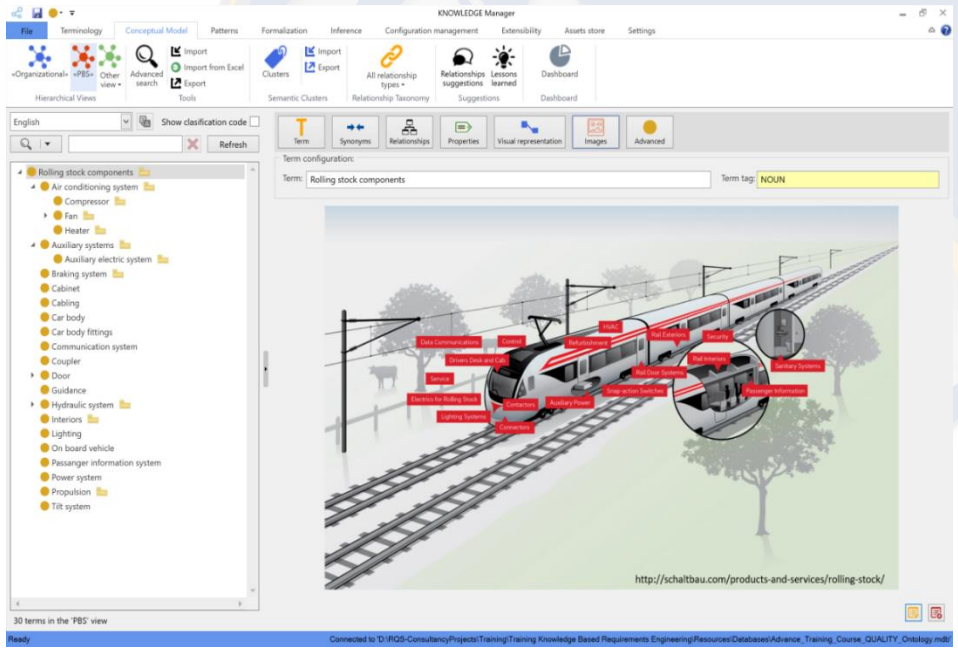
- Using the **SES Rich Authoring Tool (RAT)**



Requirements Authoring with quality on the fly assessment using a configured set of rules.



KM - Knowledge Manager

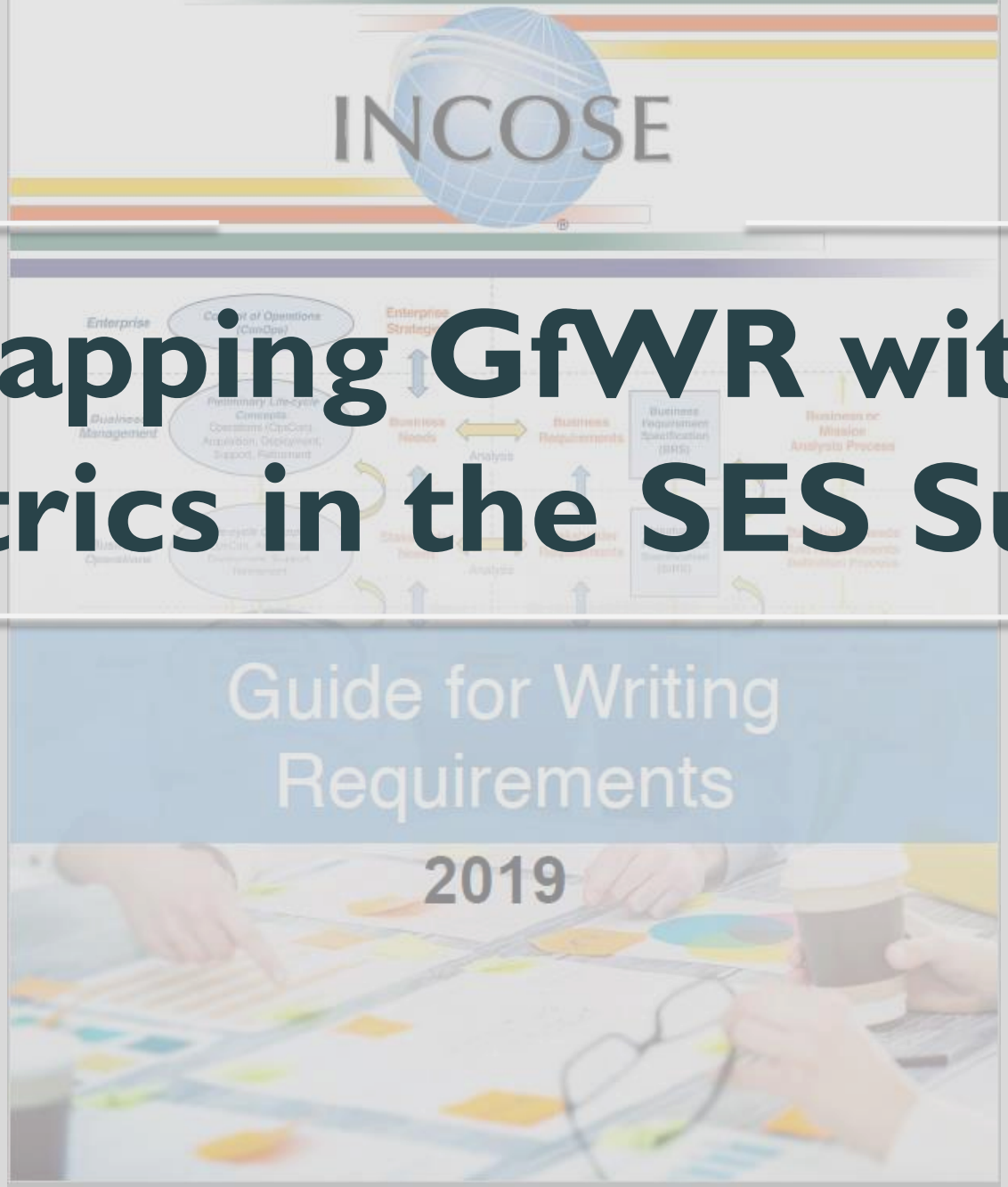


Manages **terminology** and **vocabulary**
Supports **breakdown structures**
Manages the **knowledge models** that better represent your project
Helps you to create the **patterns** used during authoring and control
Provides methods for automatic generation of Ontologies
Manages knowledge **evolution** over time
Manages and reuses knowledge **libraries**

Manages and Reuses the Knowledge of the Organization

Knowledge is one of the most valuable assets in your organization. The key driver to success in any system and software project is to reuse knowledge.
Knowledge should therefore be gathered from different sources, stored in secure repositories and accessed by the appointed personnel at the appropriate time.
KM - Knowledge Manager allows you to manage knowledge from the systems engineering point of view and to store valuable information from requirements, models, system architectures and other documents in a common System Knowledge Base

Mapping GfWR with metrics in the SES Suite



Mapping INCOSE 2019 rules per characteristic with TRC tool metrics

SCORE

37 out of 41 INCOSE rules are cover

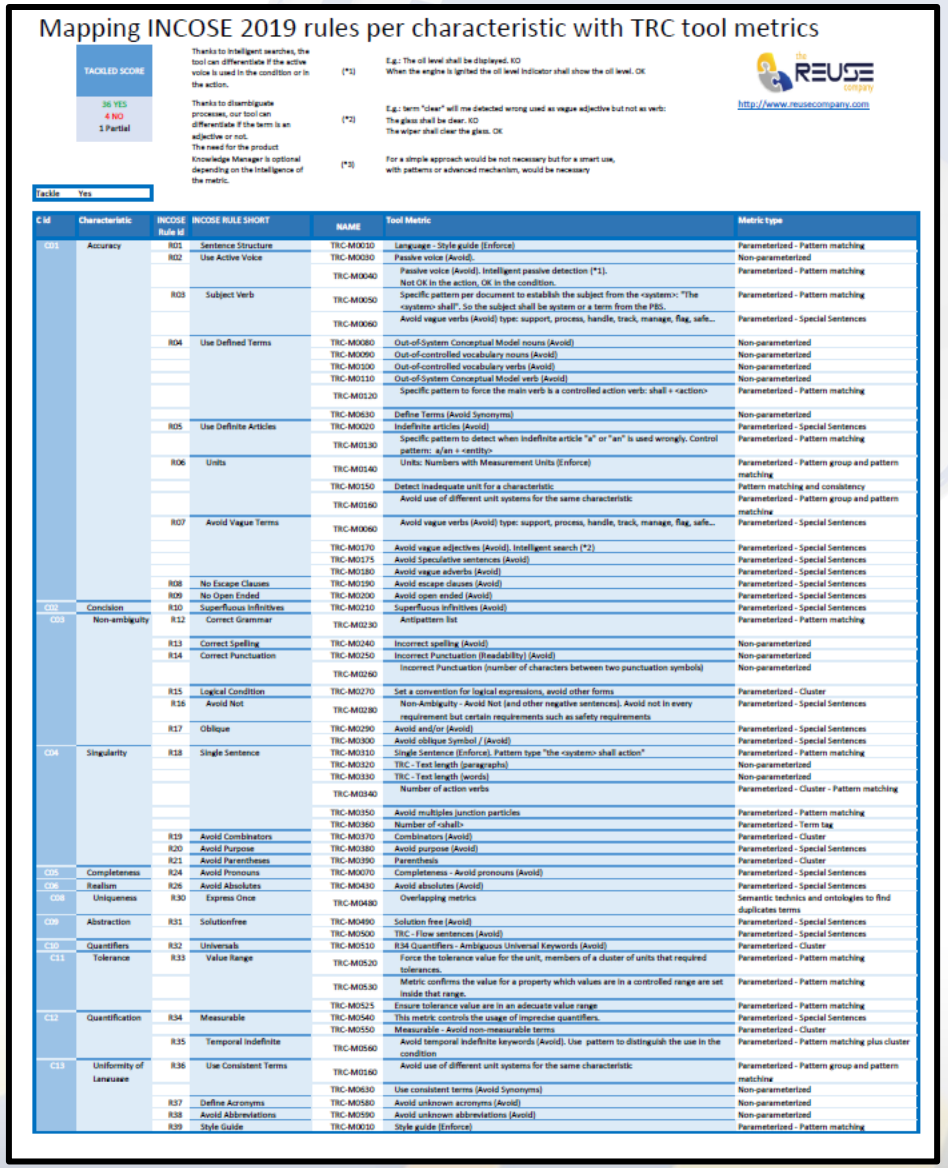
with 57 TRC metrics in this mapping.

Mapping types

One INCOSE rule / One TRC metric

One INCOSE rule / Several TRC Metrics

Direct approach / progressive / indirect or multiple



Mapping INCOSE 2019 rules per characteristic with TRC tool metrics

TACKLED SCORE

36 YES
4 NO
1 Partial

Thanks to intelligent searches, the tool can differentiate if the active voice is used in the condition or in the action.

(*1) E.g.: The oil level shall be displayed. KO
When the engine is ignited the oil level indicator shall show the oil level. OK

Thanks to disambiguate processes, our tool can differentiate if the term is an adjective or not.
The need for the product Knowledge Manager is optional depending on the intelligence of the metric.

(*2) E.g.: term "clear" will me detected wrong used as vague adjective but not as verb:
The glass shall be clear. KO
The wiper shall clear the glass. OK

(*3) For a simple approach would be not necessary but for a smart use, with patterns or advanced mechanism, would be necessary

the REUSE company
http://www.reusecompany.com

Tackle Yes

C id	Characteristic	INCOSE Rule id	INCOSE RULE SHORT	NAME	Tool Metric	Metric type
C01	Accuracy	R01	Sentence Structure	TRC-M0010	Language - Style guide (Enforce)	Parameterized - Pattern matching
		R02	Use Active Voice	TRC-M0030	Passive voice (Avoid).	Non-parameterized
				TRC-M0040	Passive voice (Avoid). Intelligent passive detection (*1). Not OK in the action, OK in the condition.	Parameterized - Pattern matching
		R03	Subject Verb	TRC-M0050	Specific pattern per document to establish the subject from the <system>: "The <system> shall". So the subject shall be system or a term from the PBS.	Parameterized - Pattern matching
				TRC-M0060	Avoid vague verbs (Avoid) type: support, process, handle, track, manage, flag, safe...	Parameterized - Special Sentences
		R04	Use Defined Terms	TRC-M0080	Out-of-System Conceptual Model nouns (Avoid)	Non-parameterized
				TRC-M0090	Out-of-controlled vocabulary nouns (Avoid)	Non-parameterized
				TRC-M0100	Out-of-controlled vocabulary verbs (Avoid)	Non-parameterized
				TRC-M0110	Out-of-System Conceptual Model verb (Avoid)	Non-parameterized
				TRC-M0120	Specific pattern to force the main verb is a controlled action verb: shall + <action>	Parameterized - Pattern matching
				TRC-M0630	Define Terms (Avoid Synonyms)	Non-parameterized
		R05	Use Definite Articles	TRC-M0020	Indefinite articles (Avoid)	Parameterized - Special Sentences
				TRC-M0130	Specific pattern to detect when indefinite article "a" or "an" is used wrongly. Control pattern: a/an + <entity>	Parameterized - Pattern matching
		R06	Units	TRC-M0140	Units: Numbers with Measurement Units (Enforce)	Parameterized - Pattern group and pattern matching
				TRC-M0150	Detect inadequate unit for a characteristic	Pattern matching and consistency
				TRC-M0160	Avoid use of different unit systems for the same characteristic	Parameterized - Pattern group and pattern matching
		R07	Avoid Vague Terms	TRC-M0060	Avoid vague verbs (Avoid) type: support, process, handle, track, manage, flag, safe...	Parameterized - Special Sentences
				TRC-M0170	Avoid vague adjectives (Avoid). Intelligent search (*2)	Parameterized - Special Sentences
				TRC-M0175	Avoid Speculative sentences (Avoid)	Parameterized - Special Sentences
				TRC-M0180	Avoid vague adverbs (Avoid)	Parameterized - Special Sentences
		R08	No Escape Clauses	TRC-M0190	Avoid escape clauses (Avoid)	Parameterized - Special Sentences
		R09	No Open Ended	TRC-M0200	Avoid open ended (Avoid)	Parameterized - Special Sentences
C02	Concision	R10	Superfluous Infinitives	TRC-M0210	Superfluous Infinitives (Avoid)	Parameterized - Special Sentences
C03	Non-ambiguity	R12	Correct Grammar	TRC-M0230	Antipattern list	Parameterized - Pattern matching
		R13	Correct Spelling	TRC-M0240	Incorrect spelling (Avoid)	Non-parameterized
		R14	Correct Punctuation	TRC-M0250	Incorrect Punctuation (Readability) (Avoid)	Non-parameterized
				TRC-M0260	Incorrect Punctuation (number of characters between two punctuation symbols)	Non-parameterized
		R15	Logical Condition	TRC-M0270	Set a convention for logical expressions, avoid other forms	Parameterized - Cluster
		R16	Avoid Not	TRC-M0280	Non-Ambiguity - Avoid Not (and other negative sentences). Avoid not in every requirement but certain requirements such as safety requirements	Parameterized - Special Sentences
		R17	Oblique	TRC-M0290	Avoid and/or (Avoid)	Parameterized - Special Sentences
				TRC-M0300	Avoid oblique Symbol / (Avoid)	Parameterized - Special Sentences
C04	Singularity	R18	Single Sentence	TRC-M0310	Single Sentence (Enforce). Pattern type "the <system> shall action"	Parameterized - Pattern matching
				TRC-M0320	TRC - Text length (paragraphs)	Non-parameterized
				TRC-M0330	TRC - Text length (words)	Non-parameterized
				TRC-M0340	Number of action verbs	Parameterized - Cluster - Pattern matching
				TRC-M0350	Avoid multiples junction particles	Parameterized - Pattern matching
				TRC-M0360	Number of <shall>	Parameterized - Term tag
		R19	Avoid Combinators	TRC-M0370	Combinators (Avoid)	Parameterized - Cluster
		R20	Avoid Purpose	TRC-M0380	Avoid purpose (Avoid)	Parameterized - Special Sentences
		R21	Avoid Parentheses	TRC-M0390	Parenthesis	Parameterized - Cluster
C05	Completeness	R24	Avoid Pronouns	TRC-M0070	Completeness - Avoid pronouns (Avoid)	Parameterized - Special Sentences
C06	Realism	R26	Avoid Absolutes	TRC-M0430	Avoid absolutes (Avoid)	Parameterized - Special Sentences
C08	Uniqueness	R30	Express Once	TRC-M0480	Overlapping metrics	Semantic technics and ontologies to find duplicates terms
C09	Abstraction	R31	Solutionfree	TRC-M0490	Solution free (Avoid)	Parameterized - Special Sentences
				TRC-M0500	TRC - Flow sentences (Avoid)	Parameterized - Special Sentences
C10	Quantifiers	R32	Universals	TRC-M0510	R34 Quantifiers - Ambiguous Universal Keywords (Avoid)	Parameterized - Cluster
C11	Tolerance	R33	Value Range	TRC-M0520	Force the tolerance value for the unit, members of a cluster of units that required tolerances.	Parameterized - Pattern matching
				TRC-M0530	Metric confirms the value for a property which values are in a controlled range are set inside that range.	Parameterized - Pattern matching
				TRC-M0525	Ensure tolerance value are in an adecuate value range	Parameterized - Pattern matching
C12	Quantification	R34	Measurable	TRC-M0540	This metric controls the usage of imprecise quantifiers.	Parameterized - Special Sentences
				TRC-M0550	Measurable - Avoid non-measurable terms	Parameterized - Cluster
		R35	Temporal Indefinite	TRC-M0560	Avoid temporal indefinite keywords (Avoid). Use pattern to distinguish the use in the condition	Parameterized - Pattern matching plus cluster
C13	Uniformity of Lenguage	R36	Use Consistent Terms	TRC-M0160	Avoid use of different unit systems for the same characteristic	Parameterized - Pattern group and pattern matching
				TRC-M0630	Use consistent terms (Avoid Synonyms)	Non-parameterized
		R37	Define Acronyms	TRC-M0580	Avoid unknown acronyms (Avoid)	Non-parameterized
		R38	Avoid Abbreviations	TRC-M0590	Avoid unknown abbreviations (Avoid)	Non-parameterized
		R39	Style Guide	TRC-M0010	Style guide (Enforce)	Parameterized - Pattern matching

https://share.hsforms.com/1-Cjrr0N1RBOiCd258J0jdQ2lpn5

Mapping INCOSE 2019 rules per characteristic with TRC tool metrics

Mapping approach types examples:

Direct approach

R13	Correct Spelling	TRC-M0240	Incorrect spelling (Avoid)

Progressive

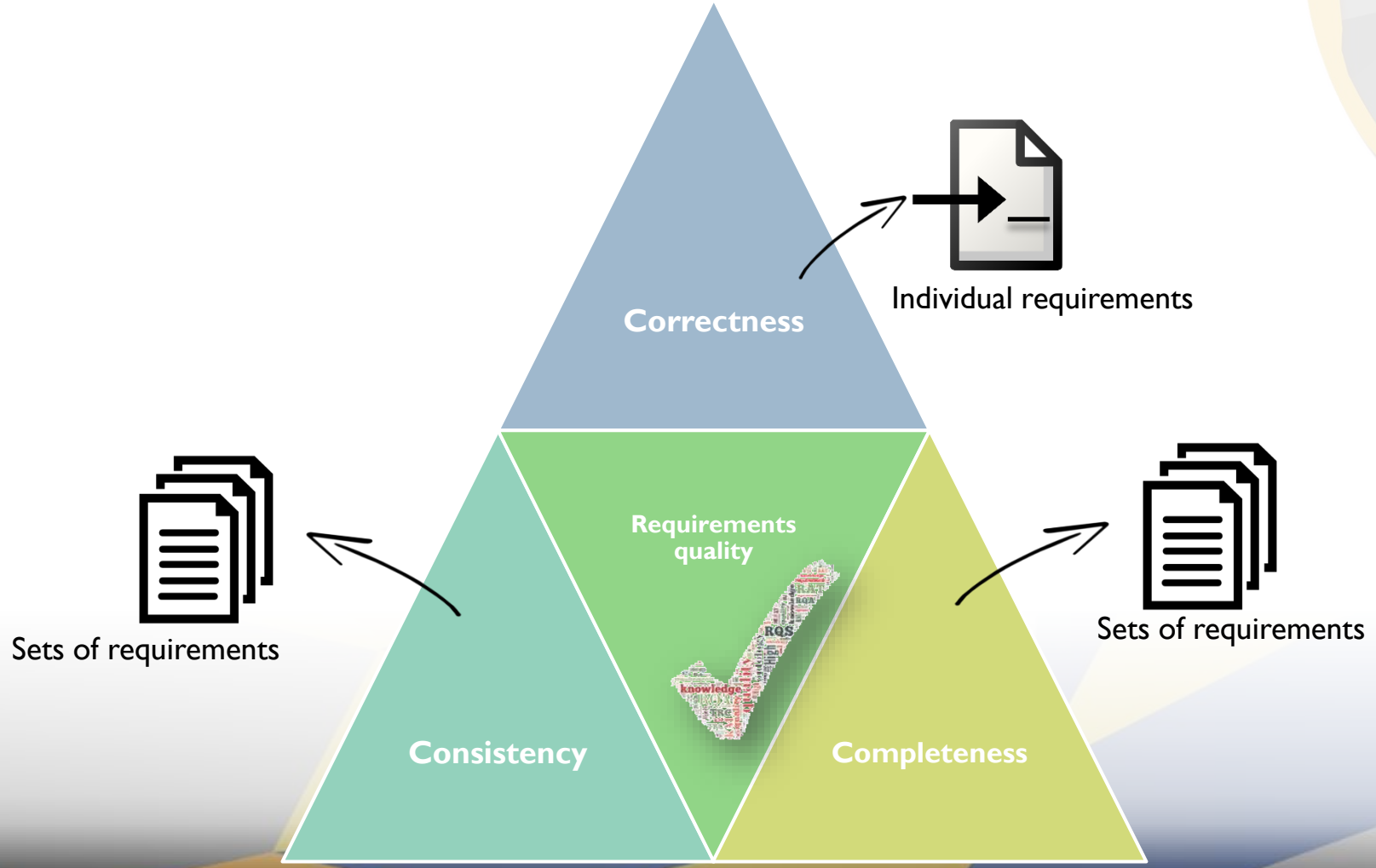
R02	Use Active Voice	TRC-M0030	Passive voice (Avoid).
		TRC-M0040	Passive voice (Avoid). Intelligent passive detection (*1). Not OK in the action, OK in the condition.

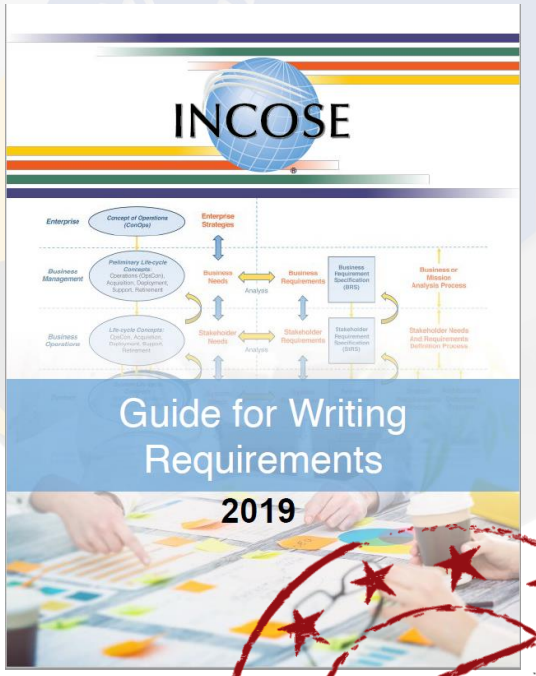
Indirect or multiple

R07	Avoid Vague Terms	TRC-M0060	Avoid vague verbs (Avoid) type: support, process, handle, track, manage, flag, safe…
		TRC-M0170	Avoid vague adjectives (Avoid). Intelligent search (*2)
		TRC-M0175	Avoid Speculative sentences (Avoid)
		TRC-M0180	Avoid vague adverbs (Avoid)
R18	Single Sentence	TRC-M0310	Single Sentence (Enforce). Pattern type "the <system> shall action"
		TRC-M0320	TRC - Text length (paragraphs)
		TRC-M0330	TRC - Text length (words)
		TRC-M0340	Number of action verbs
		TRC-M0350	Avoid multiples junction particles
		TRC-M0360	Number of <shall>

Requirements quality metrics: CCC Approach

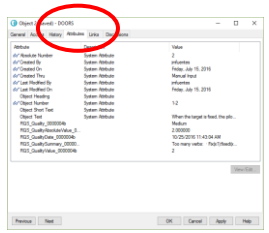
- CCC – Correctness, Consistency and Completeness

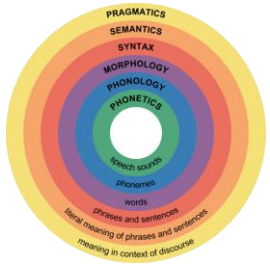


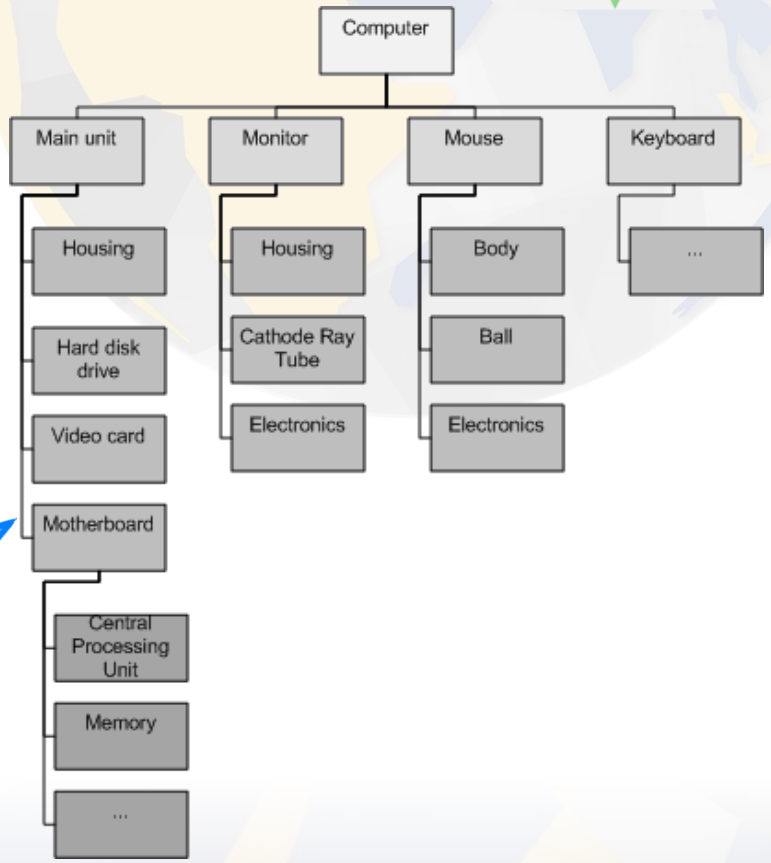


Examples of requirements metrics: Correctness

- Metrics based on information coming from the RMS:
  - Attributes, links, versions…
- Metrics based on lists of terms:
  - Forbidden: ambiguous, pronouns…
  - Restricted: negations…
  - Mandatory: 'shall', 'will', 'should'…
- Metrics based on linguistic algorithms:
  - Text length, misspelling, readability….
  - Detection of passive voice, imperative tense…
- Metrics based on the conformance with models:
  - Concepts in your requirements coming from PBS, FBS…
- Metrics based on patterns:
  - Compliance with different types of requirements patterns
  - Detection of specific structures within the requirements



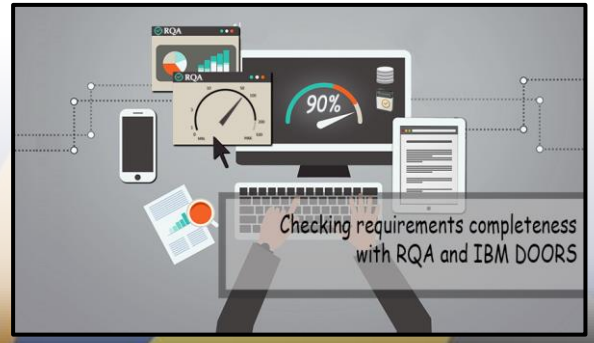




Examples of requirements metrics: Completeness

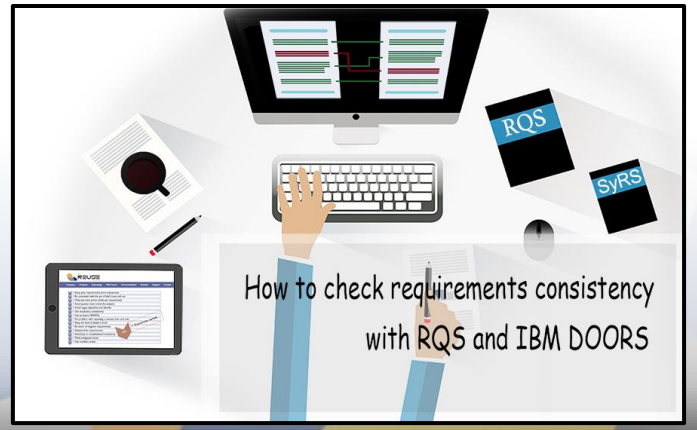
- Completeness at specification/project level:
  - Are all the expected requirements types involved in your specifications?
  - Are all the key concepts (from the ontology or from other models, e.g. blocks, states, signals, properties…) properly covered?
    - Does the whole set of requirements documents include requirements for all the elements of the system according to a block diagram (architecture)?
    - Does the spec. include requirements mentioning all the signals?
    - Does the spec. include requirements describing the behavior of the system elements in any of their possible states and modes?
  - Are your requirements properly linked? At the different levels?
  - 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 at requirement level:
  - Does every requirement include all the agreed parts (condition, subject…):

    following patterns
  - Are you stating the values for the mentioned properties with tolerances: 12V±0.5V



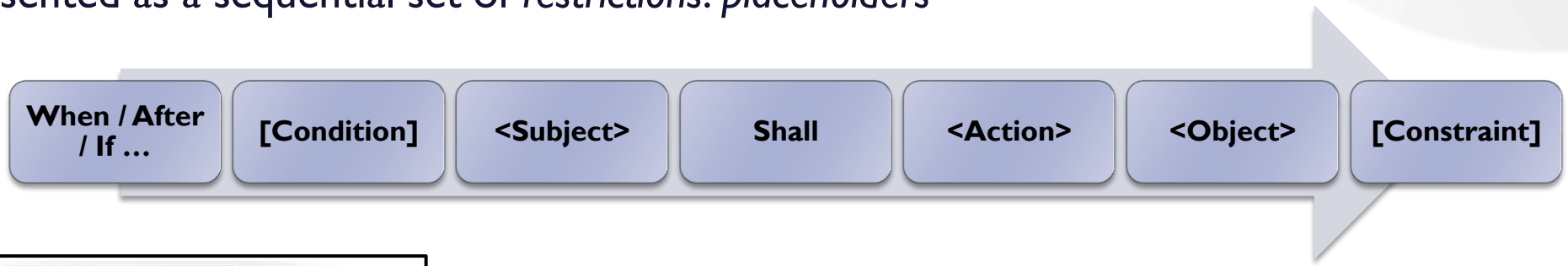
Examples of requirements metrics: Consistency

- Are your requirements consistent with each other?
- Are your requirements consistent with the models of your projects?
- Do you have duplicated requirements in your specifications?
- Are the values for the mentioned signals within the expected ranges?
- Are you using the proper measurement units in your requirements?
- Are all the properties property allocated along the system decomposition?
- Are your requirements describing wrong transitions in a statechart?



Patterns

- Represents the structures every *correct* requirement should meet
- Different types of requirements → different patterns (templates)
- Customizable for every domain, customer and content of each requirements document
- Libraries with sets of patterns
- Represented as a sequential set of *restrictions*: *placeholders*



Patterns

When / After / If …	[Condition]	<Subject>	Shall	<Action>	<Object>	[Constraint]

4.1.2 R2 - /ACCURACY/USEACTIVEVOICE

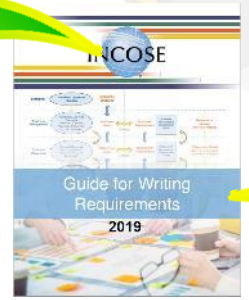
Use the active voice in the main sentence structure of the need or requirement statement with the responsible entity clearly identified as the subject of the sentence.

Elaboration:

The active voice requires that the entity performing the action is the subject of the sentence. This is important in writing needs and requirements since the onus for satisfying the requirement is on the subject, not the object of the statement. If the entity responsible for the action is not identified explicitly, it is unclear who or what should perform the action making verification of that requirement very difficult. Including the entity in the subject also
refers to the appropriate level consistent with the entity name (s

Often when the phrase "shall be" is used, the statement is in the

GUIDE



4.4.2 R19 - /SINGULARITY/AVOIDCOMBINATORS

Avoid combinators.

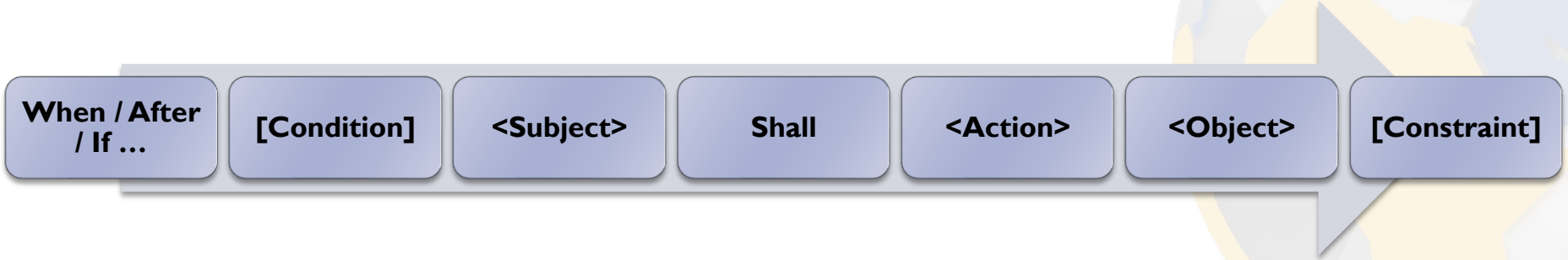
Elaboration:

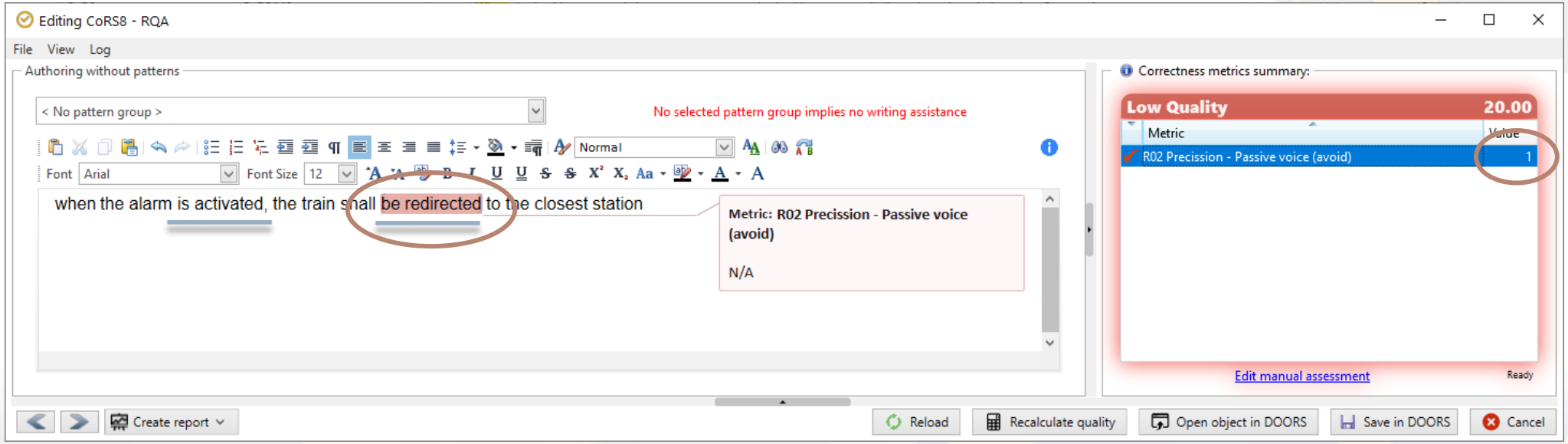
Combinators are words that join clauses, such as "and", "or", "then", "unless", "but", "as well as", "but also", "however", "whether", "meanwhile", "whereas", "on the other hand", and "otherwise." Their presence in a requirement usually indicates that multiple requirements should be written.

Exception: AND, OR, NOT can be used in need and requirement statements as logical conditions and qualifiers as stated in R15.

See also R16 and R17.

Patterns





Mapping INCOSE rules with our RQA metrics

Advanced semantic techniques

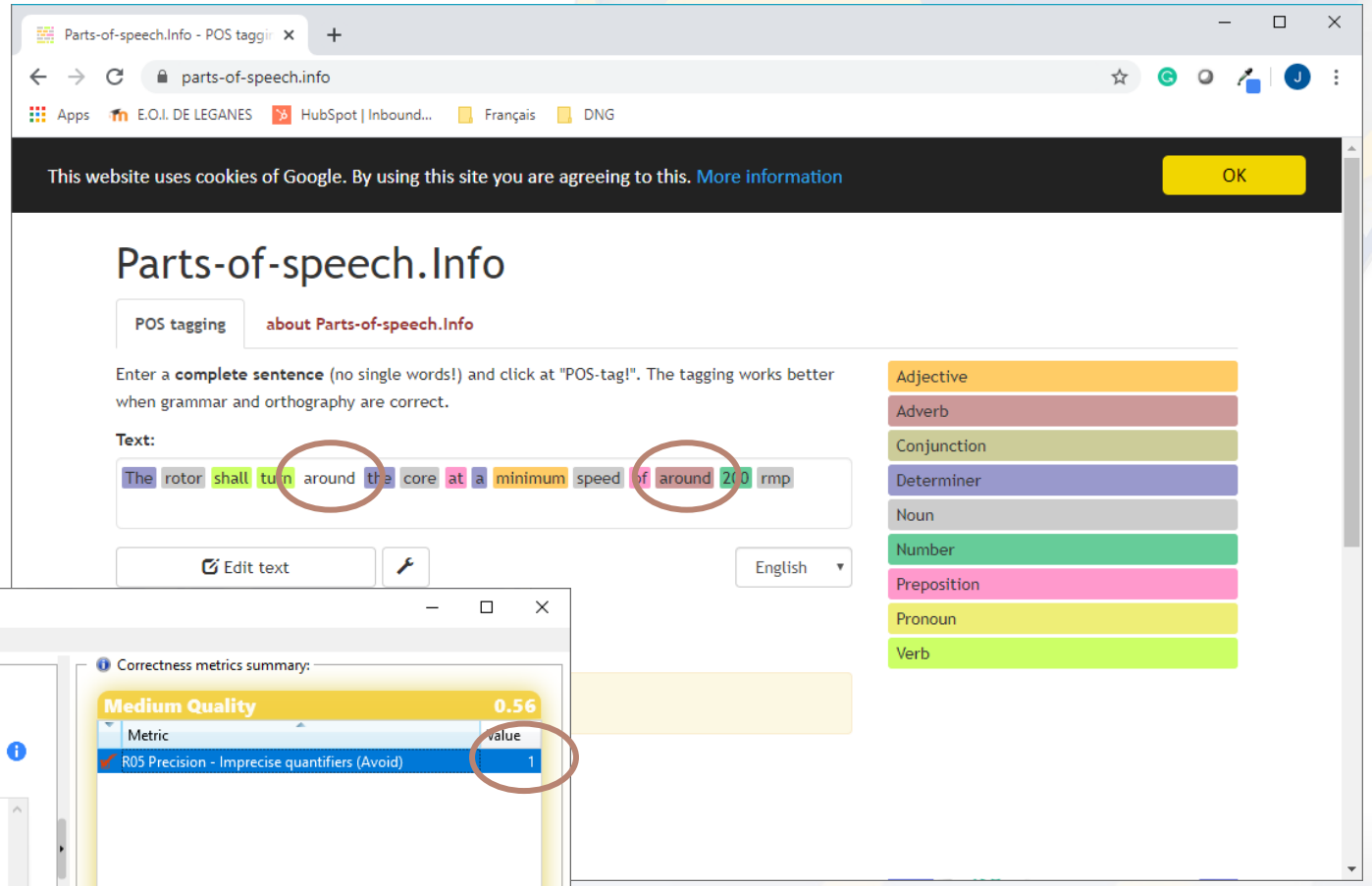
System Requirements Spec.

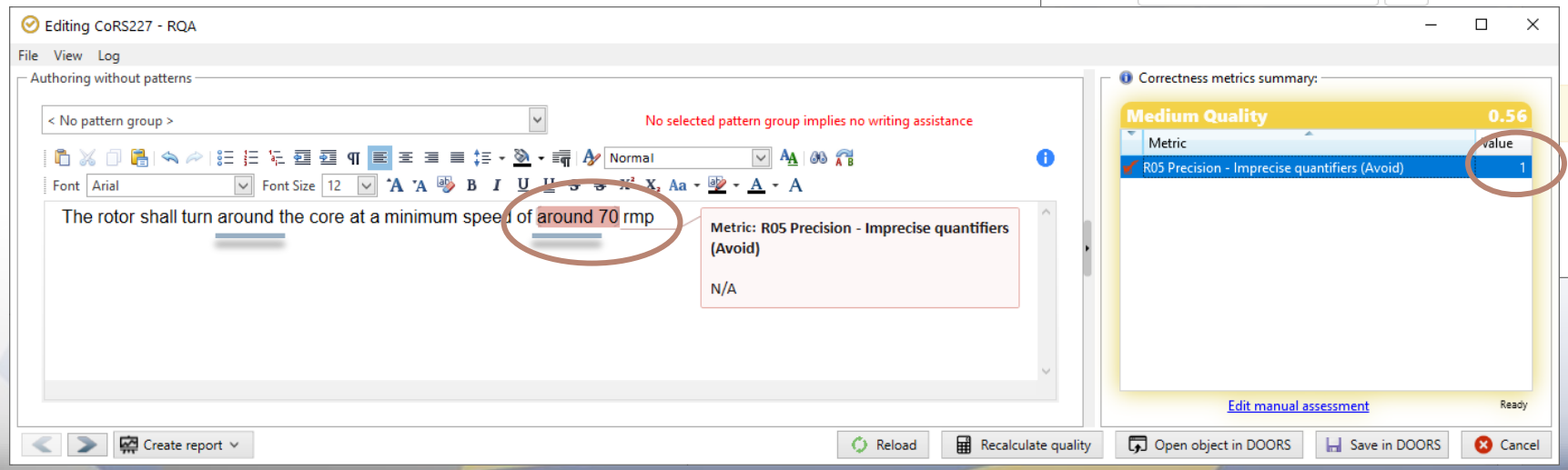
…

SyR-088: xxx

SyR-089: The rotor shall turn around the core at a minimum speed of around 70 rmp

SyR-090: xxx





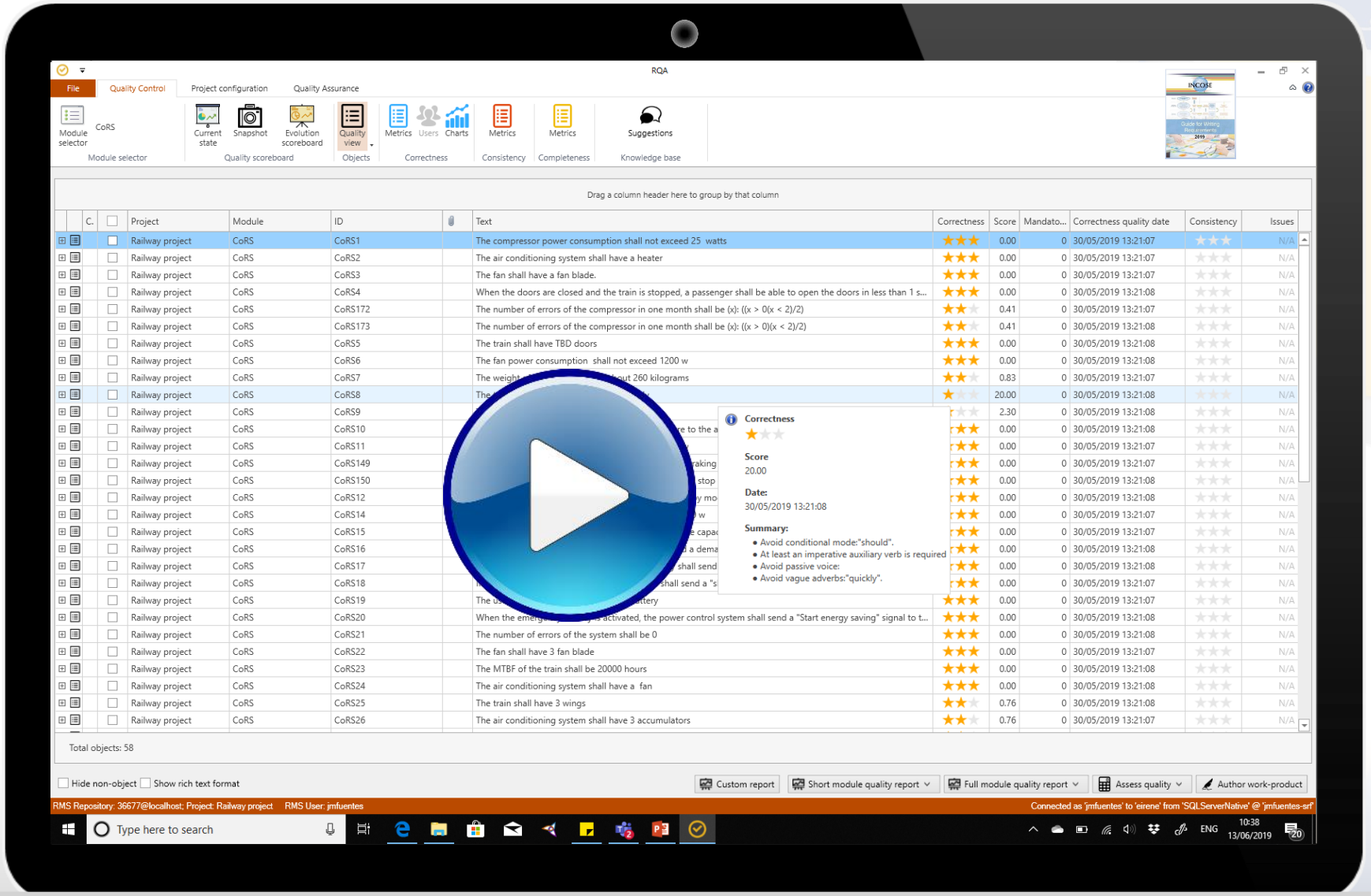
Mapping INCOSE rules with our RQA metrics

C id	Characteristic	INCOSE Rule id	INCOSE RULE SHORT	NAME	Tool Metric	Metric type
C01	Accuracy	R01	Sentence Structure	TRC-M0010	Language - Style guide (Enforce)	Parameterized - Pattern matching
		R02	Use Active Voice	TRC-M0030	Passive voice (Avoid).	Non-parameterized
				TRC-M0040	Passive voice (Avoid). Intelligent passive detection (*1). Not OK in the action, OK in the condition.	Parameterized - Pattern matching
		R03	Subject Verb	TRC-M0050	Specific pattern per document to establish the subject from the <system>: "The <system> shall". So the subject shall be system or a term from the PBS.	Parameterized - Pattern matching
				TRC-M0060	Avoid vague verbs (Avoid) type: support, process, handle, track, manage, flag, safe...	Parameterized - Special Sentences
		R04	Use Defined Terms	TRC-M0080	Out-of-System Conceptual Model nouns (Avoid)	Non-parameterized
				TRC-M0090	Out-of-controlled vocabulary nouns (Avoid)	Non-parameterized
				TRC-M0100	Out-of-controlled vocabulary verbs (Avoid)	Non-parameterized
				TRC-M0110	Out-of-System Conceptual Model verb (Avoid)	Non-parameterized
				TRC-M0120	Specific pattern to force the main verb is a controlled action verb: shall + <action>	Parameterized - Pattern matching
				TRC-M0630	Define Terms (Avoid Synonyms)	Non-parameterized
		R05	Use Definite Articles	TRC-M0020	Indefinite articles (Avoid)	Parameterized - Special Sentences
				TRC-M0130	Specific pattern to detect when indefinite article "a" or "an" is used wrongly. Control pattern: a/an + <entity>	Parameterized - Pattern matching
		R06	Units	TRC-M0140	Units: Numbers with Measurement Units (Enforce)	Parameterized - Pattern group and pattern matching
				TRC-M0150	Detect inadequate unit for a characteristic	Pattern matching and consistency
				TRC-M0160	Avoid use of different unit systems for the same characteristic	Parameterized - Pattern group and pattern matching
		R07	Avoid Vague Terms	TRC-M0060	Avoid vague verbs (Avoid) type: support, process, handle, track, manage, flag, safe...	Parameterized - Special Sentences
				TRC-M0170	Avoid vague adjectives (Avoid). Intelligent search (*2)	Parameterized - Special Sentences
				TRC-M0175	Avoid Speculative sentences (Avoid)	Parameterized - Special Sentences
				TRC-M0180	Avoid vague adverbs (Avoid)	Parameterized - Special Sentences
		R08	No Escape Clauses	TRC-M0190	Avoid escape clauses (Avoid)	Parameterized - Special Sentences
		R09	No Open Ended	TRC-M0200	Avoid open ended (Avoid)	Parameterized - Special Sentences

Mapping INCOSE rules with our RQA metrics

C03	Non-ambiguity	R12	Correct Grammar	TRC-M0230	Antipattern list	Parameterized - Pattern matching
		R13	Correct Spelling	TRC-M0240	Incorrect spelling (Avoid)	Non-parameterized
		R14	Correct Punctuation	TRC-M0250	Incorrect Punctuation (Readability) (Avoid)	Non-parameterized
				TRC-M0260	Incorrect Punctuation (number of characters between two punctuation symbols)	Non-parameterized
		R15	Logical Condition	TRC-M0270	Set a convention for logical expressions, avoid other forms	Parameterized - Cluster
		R16	Avoid Not	TRC-M0280	Non-Ambiguity - Avoid Not (and other negative sentences). Avoid not in every requirement but certain requirements such as safety requirements	Parameterized - Special Sentences
		R17	Oblique	TRC-M0290	Avoid and/or (Avoid)	Parameterized - Special Sentences
				TRC-M0300	Avoid oblique Symbol / (Avoid)	Parameterized - Special Sentences
C04	Singularity	R18	Single Sentence	TRC-M0310	Single Sentence (Enforce). Pattern type "the <system> shall action"	Parameterized - Pattern matching
				TRC-M0320	TRC - Text length (paragraphs)	Non-parameterized
				TRC-M0330	TRC - Text length (words)	Non-parameterized
				TRC-M0340	Number of action verbs	Parameterized - Cluster - Pattern matching
				TRC-M0350	Avoid multiples junction particles	Parameterized - Pattern matching
				TRC-M0360	Number of <shall>	Parameterized - Term tag
		R19	Avoid Combinators	TRC-M0370	Combinators (Avoid)	Parameterized - Cluster
		R20	Avoid Purpose	TRC-M0380	Avoid purpose (Avoid)	Parameterized - Special Sentences
		R21	Avoid Parentheses	TRC-M0390	Parenthesis	Parameterized - Cluster
C05	Completeness	R24	Avoid Pronouns	TRC-M0070	Completeness - Avoid pronouns (Avoid)	Parameterized - Special Sentences
C06	Realism	R26	Avoid Absolutes	TRC-M0430	Avoid absolutes (Avoid)	Parameterized - Special Sentences
C08	Uniqueness	R30	Express Once	TRC-M0480	Overlapping metrics	Semantic technics and ontologies to find duplicates terms
C09	Abstraction	R31	Solutionfree	TRC-M0490	Solution free (Avoid)	Parameterized - Special Sentences
				TRC-M0500	TRC - Flow sentences (Avoid)	Parameterized - Special Sentences
C10	Quantifiers	R32	Universals	TRC-M0510	R34 Quantifiers - Ambiguous Universal Keywords (Avoid)	Parameterized - Cluster
C11	Tolerance	R33	Value Range	TRC-M0520	Force the tolerance value for the unit, members of a cluster of units that required tolerances.	Parameterized - Pattern matching
				TRC-M0530	Metric confirms the value for a property which values are in a controlled range are set inside that range.	Parameterized - Pattern matching
				TRC-M0525	Ensure tolerance value are in an adecuate value range	Parameterized - Pattern matching
C12	Quantification	R34	Measurable	TRC-M0540	This metric controls the usage of imprecise quantifiers.	Parameterized - Special Sentences
				TRC-M0550	Measurable - Avoid non-measurable terms	Parameterized - Cluster
		R35	Temporal Indefinite	TRC-M0560	Avoid temporal indefinite keywords (Avoid). Use pattern to distinguish the use in the condition	Parameterized - Pattern matching plus cluster
C13	Uniformity of Language	R36	Use Consistent Terms	TRC-M0160	Avoid use of different unit systems for the same characteristic	Parameterized - Pattern group and pattern matching
				TRC-M0630	Use consistent terms (Avoid Synonyms)	Non-parameterized
		R37	Define Acronyms	TRC-M0580	Avoid unknown acronyms (Avoid)	Non-parameterized
		R38	Avoid Abbreviations	TRC-M0590	Avoid unknown abbreviations (Avoid)	Non-parameterized
		R39	Style Guide	TRC-M0010	Style guide (Enforce)	Parameterized - Pattern matching

TRC WEBINARS 2020




Q&A

Next webinar

- **The Financial Cost of Bad Quality Requirements - a Surprising Discovery Using the COSYSMO Model**

- **Dates:**
  - April 15th and 16th
- **Description:**
  - In this webinar we will run a development project using requirements with high vs. low degree of requirements understanding and what the economic benefits would be when using the INCOSE guide of writing requirements "for real" with our state-of-the-art The Systems Engineering Suite, offering you the tools, processes and skills to properly tackle the daily issues of complex systems. This with the benefits of increased quality, saved development effort and money.


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