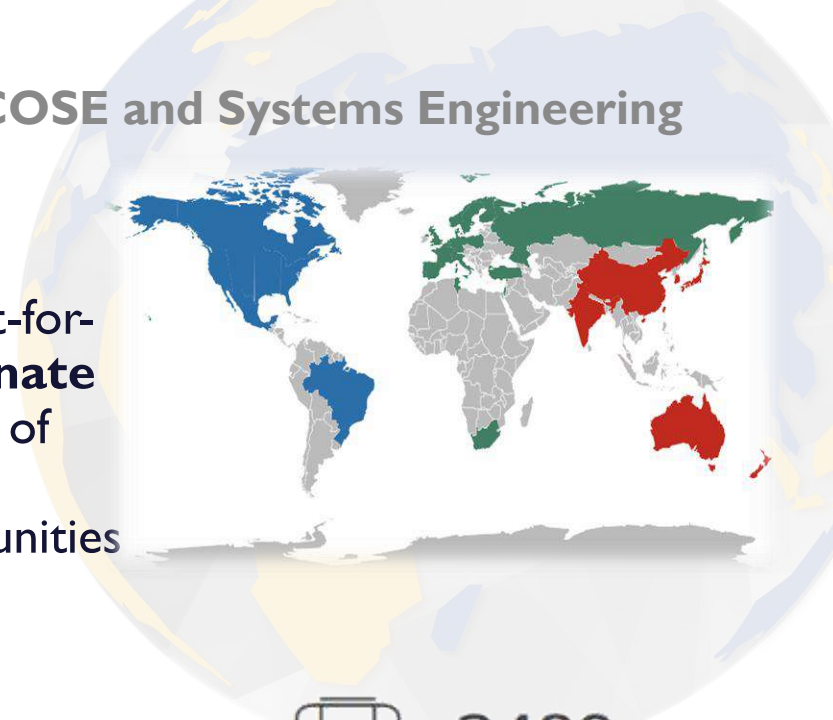


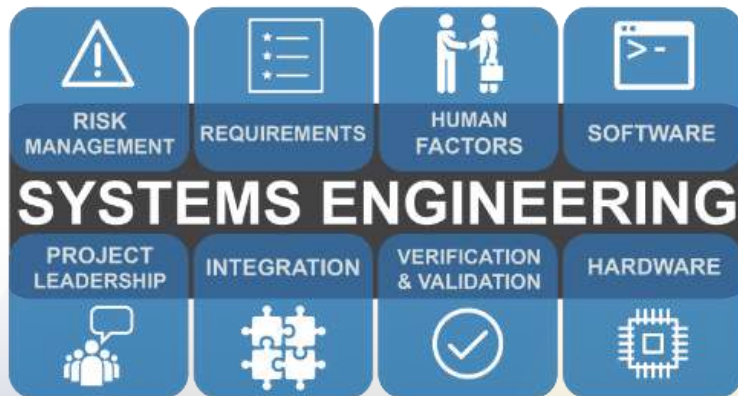
INCOSE Guide for Writing Requirements: Real-Time Quality Assessment of the INCOSE Rules

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.

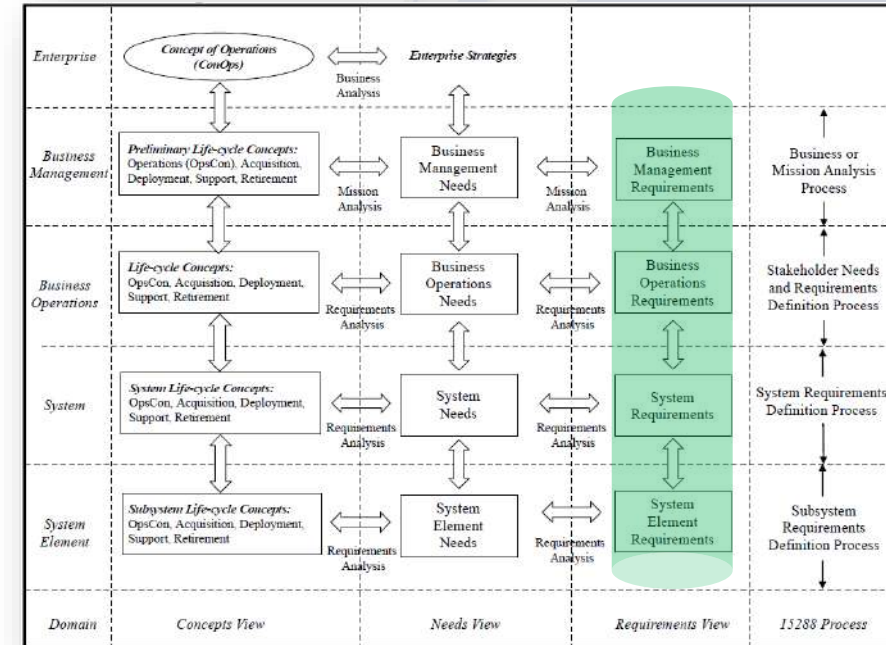
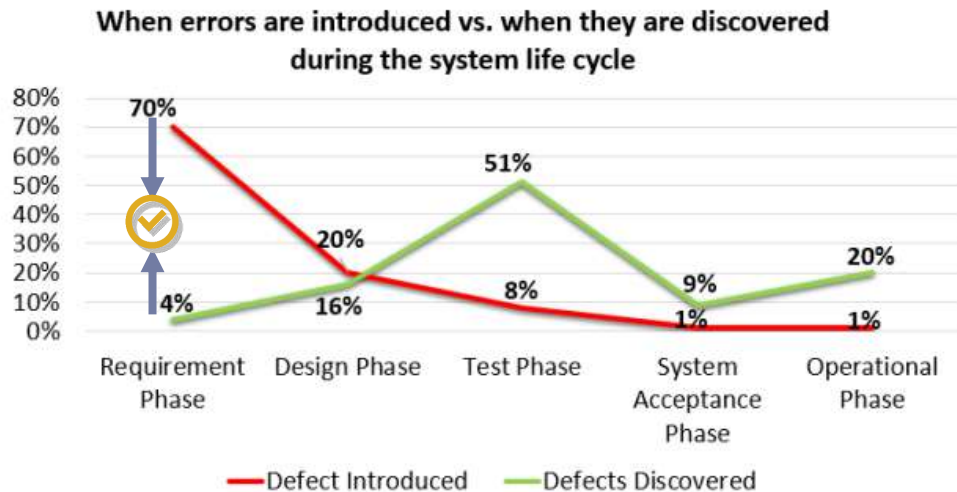


- > **Systems Engineering** is an **interdisciplinary approach** and means to enable the realization of **successful systems**. It focusses on defining customer **needs** and required **functionality** early in the development cycle, documenting **requirements**, and then proceeding with **design** synthesis 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

REQUIREMENTS are the reason for FAILURE



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??

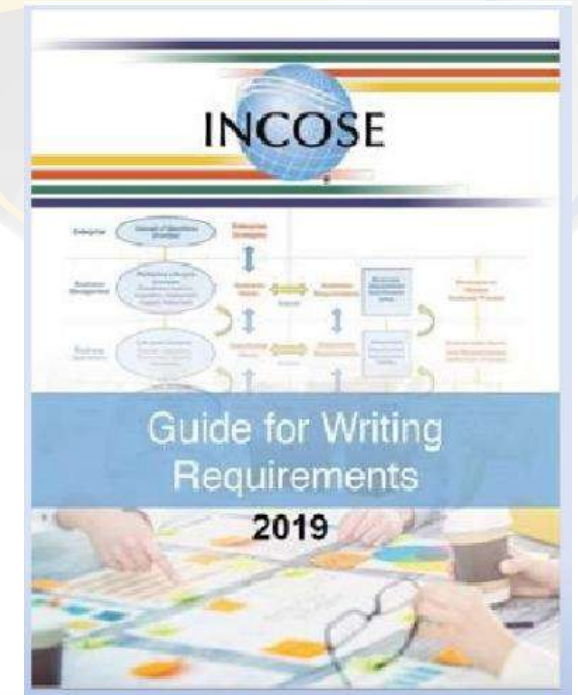
INCOSE Guide for Writing Requirements

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.

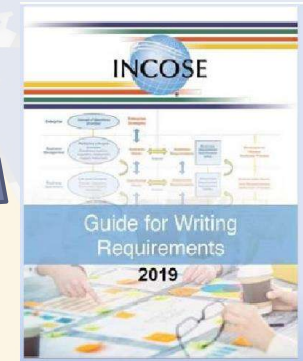


INCOSE GfWR

- **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.

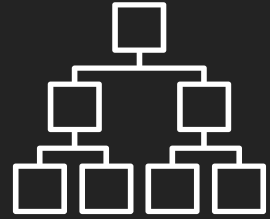
41 Rules / 14 Characteristics

Type	Rule Number	Rule name	CHARACTERISTICS OF NEED AND REQUIREMENT STATEMENTS							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
Accuracy	R01	Sentence Structure			1				1						
	R02	Use Active Voice			1				1						
	R03	Subject Verb		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						
Conciseness	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	
	R19	Avoid Combinators			1		1								
	R20	Avoid Purpose			1		1								
	R21	Avoid Parentheses			1		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				1		
Conditions	R27	Explicit			1			1							
Uniqueness	R28	Explicit Lists			1			1							
	R29	Classify			1			1			1	1	1		
Abstraction	R30	Express Once	1							1		1	1		
Solutionfree	R31	Solutionfree		1											
Quantifiers	R32	Universals			1			1	1						
Value Range	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	1
	R38	Avoid Abbreviations			1				1		1		1	1	1
	R39	Style Guide			1	1			1		1		1	1	1
Modularity	R40	Related Requirements							1		1		1	1	1
	R41	Structured							1		1		1	1	1



46 Attributes

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

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 GfVR)

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**.

Knowledge Ontology

05 Reasoning

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

04 Formalization

Representation of assets semantic through SRL – System Representation Language



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. Establish 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

Example of ontology

Domain specific

Common English

Vocabulary

Aircraft

A380

A350

System

Operate

Temperature

Environment

Pressure

shall

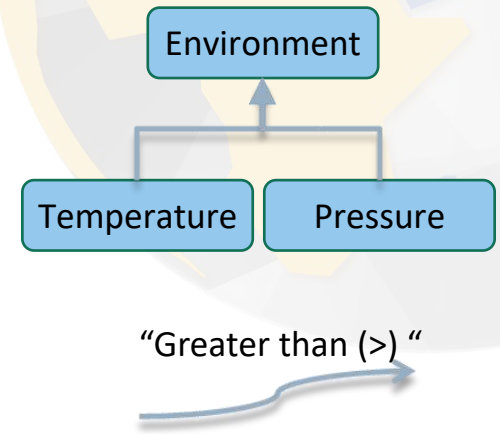
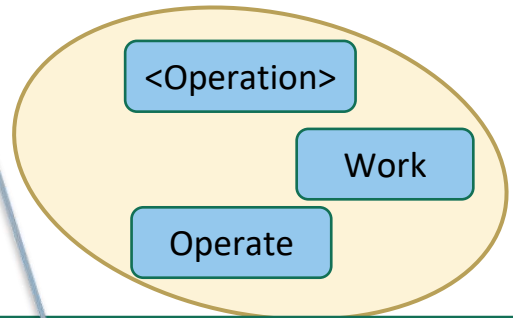
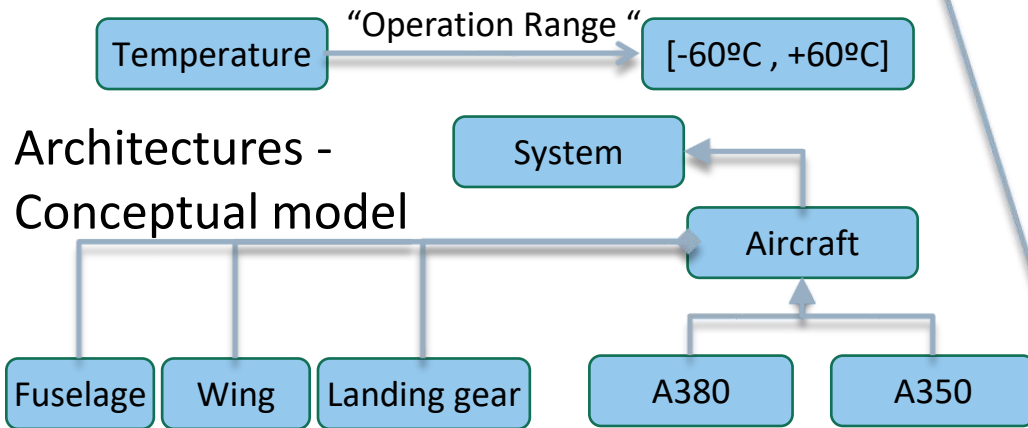
The

of

at

Lower

Architectures - Conceptual model



Patterns

System (*)

Shall

Operation (*)

At

«Minimum»

Environment (*)

Of

NUMBER

MEASUREMENT UNIT

Formalization

The A380 shall be able to operate at a minimum temperature of -70° C

Temperature

“Greater than (>)”

-70

°C

Reasoning

If NUMBER

Lower than (<) -60° °C

Or NUMBER

Greater than (>) +60° °C



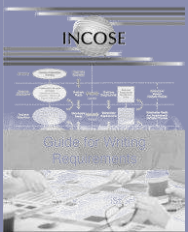
What is a Knowledge Library

- 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

Knowledge Libraries



INCOSE Guide for Writing Requirements

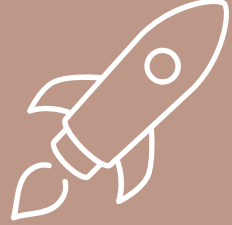
INCOSE
Quality rules for the analysis of textual requirements



EARS Patterns

EARS
Requirements patterns

ECSS and NASA
Glossary, patterns and rules




ECSS and NASA Libraries

Knowledge Base



ISO 26262 Library

ISO 26262
Glossary, patterns and rules



MASTER patterns

MASTER
Quality rules for requirements and requirements patterns



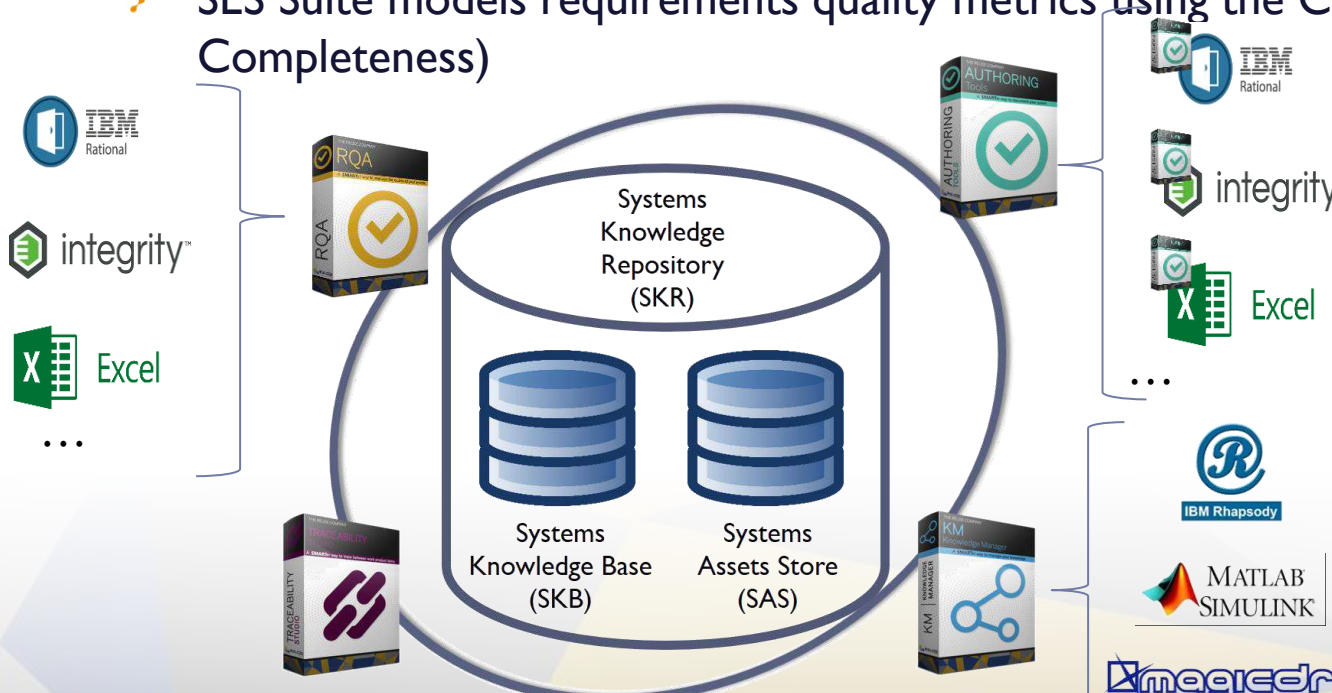
A blue plastic toolbox is shown open, revealing a collection of hand tools. On the left side, there are two pairs of pliers (one with red handles, one with black), several screwdrivers with different colored handles (red, green, blue), and a pair of scissors with white handles. On the right side, there are two adjustable wrenches, a hammer with a wooden handle, a utility knife with a green handle, and a green and yellow tape measure. A white rectangular box with a black border is overlaid in the center of the toolbox, containing the text "The Systems Engineering Suite".

The Systems Engineering Suite



The TRC Systems Engineering Suite

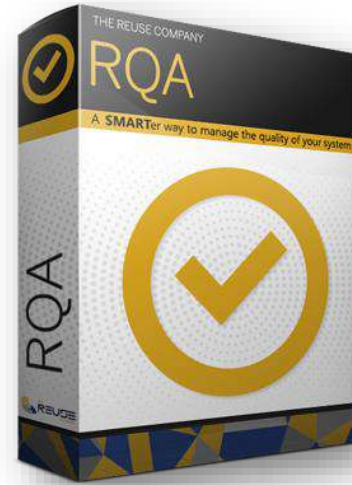
- The Systems Engineering Suite intends to tackle requirements quality management by offering a set of tools and processes
- Automatic measurement of requirements quality metric
- Support to Requirements Authoring
- SES Suite models requirements quality metrics using the CCC approach (Correctness, Consistency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



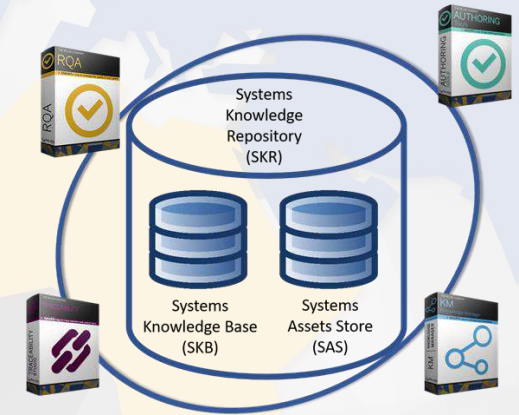
- **RQA:** to setup, check and manage the quality of a requirements specification
- **Rich Authoring Tool (RAT):** to assist authors while they are creating or editing requirements
- **Knowledge Manager (KM):** to manage knowledge around a requirements specification: dictionaries, glossaries, concept 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.



The Systems Engineering Suite



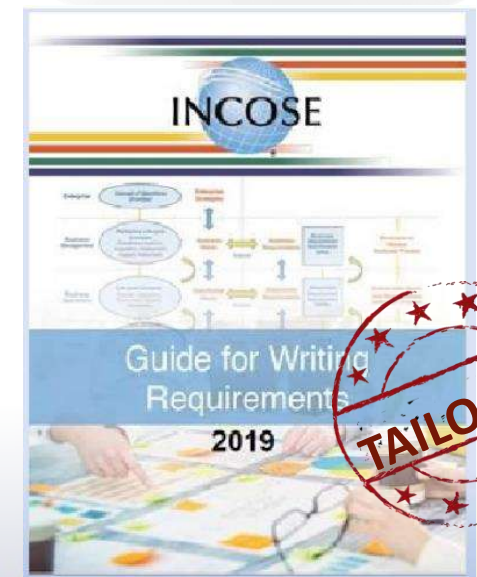
Quality Analysis of Requirements and all Kind of Engineering Items

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.



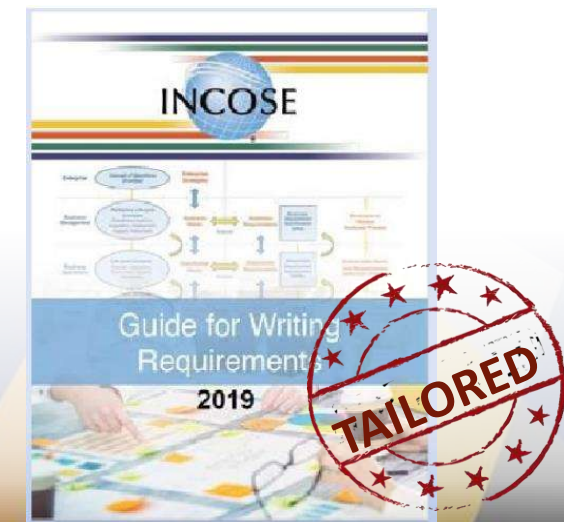
Authoring with the INCOSE GfWR recommendations and rules

Using the SES Rich Authoring Tool (RAT)

Metric	Correctness	Value	Summary	Mandatory	Weight
R02 Precision - TRC - Conditio...	★ ★ ★	1	Avoid conditional mode	<input type="checkbox"/>	1
R02 Precision - TRC - Imperativ...	★ ★ ★	0	At least an imperative auxiliary verb is required	<input type="checkbox"/>	1
R02 Precision - Passive voice (...)	★ ★ ★	1	Avoid passive voice	<input checked="" type="checkbox"/>	1
Special Sentence - R07 Precisi...	★ ★ ★	1	Avoid vague adverbs	<input type="checkbox"/>	1
Cluster «ACTION»	★ ★ ★	2	N/A	<input type="checkbox"/>	1
Pattern group: METRIC - Anti-P...	★ ★ ★	0	N/A	<input type="checkbox"/>	1
Properties measured with the n...	★ ★ ★	0	N/A	<input type="checkbox"/>	1
R14 Ambiguity - Pronouns (Av...	★ ★ ★	0	N/A	<input type="checkbox"/>	1
R14 Non Ambiguity - Incorrect...	★ ★ ★	0	N/A	<input type="checkbox"/>	1
R19 Singularity - TRC - Text len...	★ ★ ★	8	N/A	<input type="checkbox"/>	1
Special Sentence - R05 Precisi...	★ ★ ★	0	N/A	<input type="checkbox"/>	1
Special Sentence - R08 Precisi...	★ ★ ★	0	N/A	<input type="checkbox"/>	1
Special Sentence - R10 Precisi...	★ ★ ★	0	N/A	<input type="checkbox"/>	1
Relationships not SCM complia...	★ ★ ★	N/A	The quality has not been assessed because this require...	<input type="checkbox"/>	1
R44 Uniformity Of Language ...	★ ★ ★	N/A	The quality has not been assessed because this require...	<input type="checkbox"/>	0



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

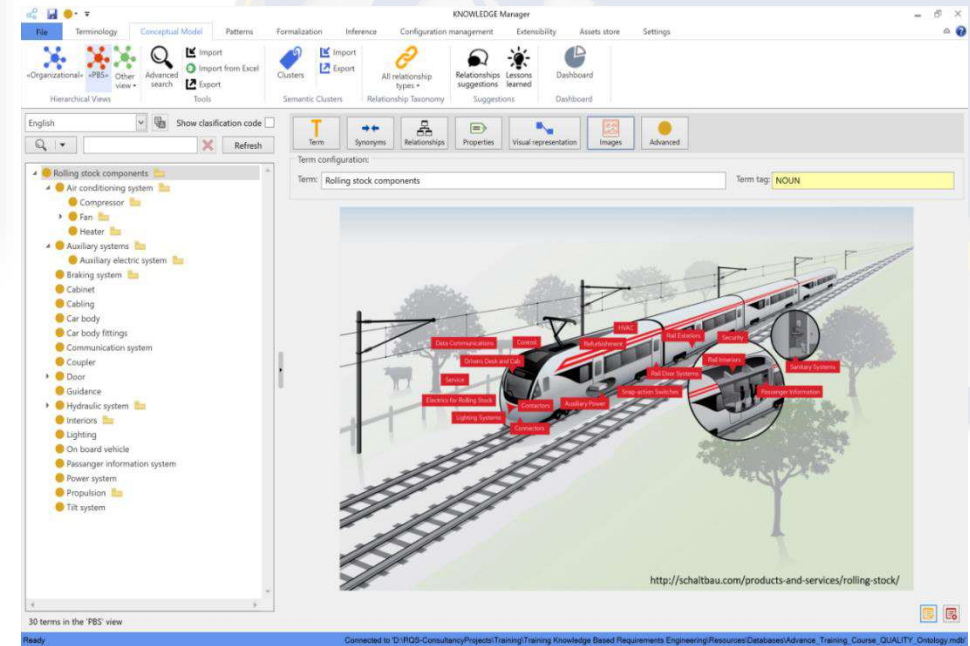
Knowledge is one of the most valuable assets in your organization. The key driver to success in any system and software project is to reuse knowledge.

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



The Systems Engineering Suite



INCOSE

Mapping GfWR with metrics in the SES Suite



Guide for Writing Requirements

2019



Mapping INCOSE 2019 rules per characteristic with TRC tool metrics

SCORE

37 out of 41 INCOSE rules are covered with 58 TRC metrics in this mapping.

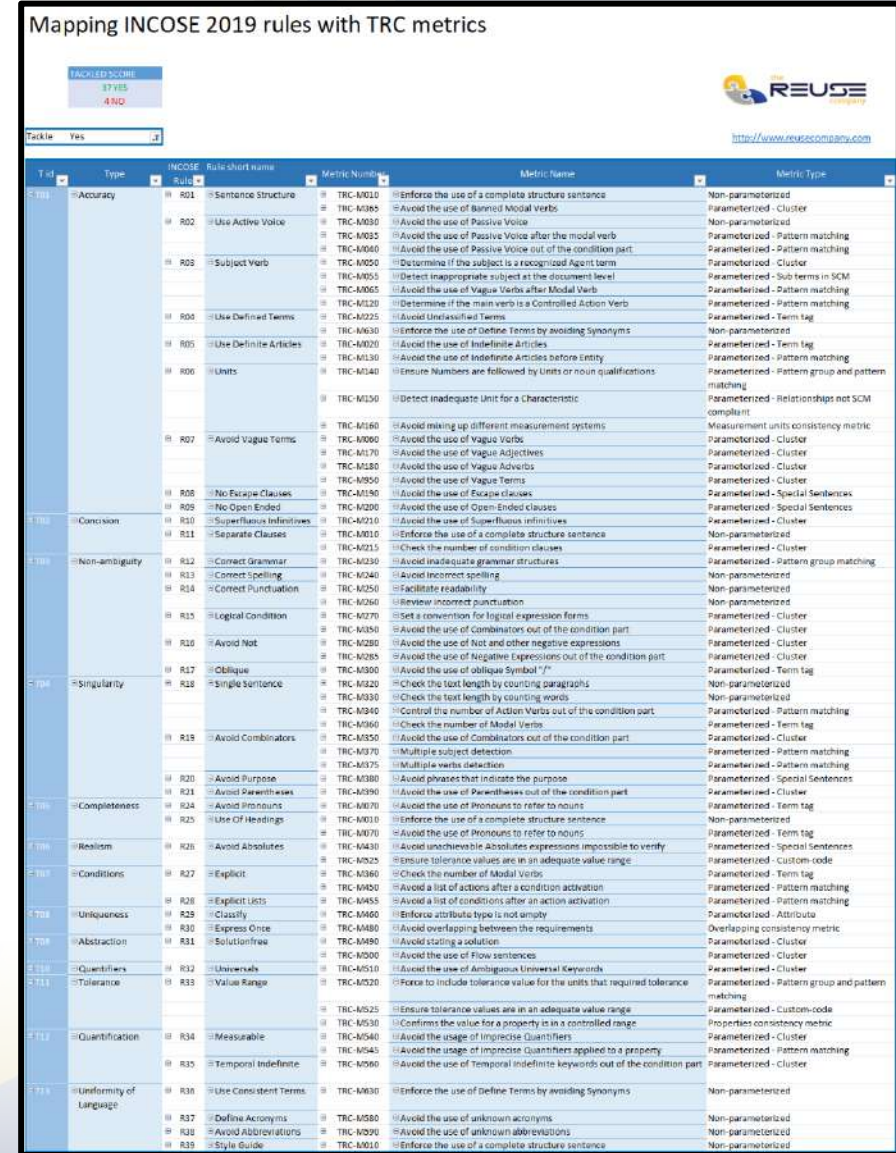
Mapping types

One INCOSE rule / One TRC metric

One INCOSE rule / Several TRC Metrics

Direct approach / progressive / indirect or multiple

Mapping INCOSE 2019 rules with TRC metrics



TRC Type	INCOSE Rule	Metric Number	Metric Name	Metric Type	
Accuracy	R01 Sentence Structure	TRC-M010	Enforce the use of a complete structure sentence	Non-parameterized	
	R02 Use Active Voice	TRC-M085	Avoid the use of banned Modal Verbs	Parameterized - Cluster	
	R03 Subject Verb	TRC-M030	Avoid the use of Passive Voice	Non-parameterized	
		TRC-M035	Avoid the use of Passive Voice after the modal verb	Parameterized - Pattern matching	
	R04 Use Defined Terms	TRC-M060	Avoid the use of Passive Voice out of the condition part	Parameterized - Pattern matching	
		TRC-M050	Determine if the subject is a recognized Agent term	Parameterized - Cluster	
	R05 Use Definite Articles	TRC-M055	Detect inappropriate subject at the document level	Parameterized - Sub terms in SCM	
		TRC-M065	Avoid the use of Vague Verbs after Modal Verb	Parameterized - Pattern matching	
	R06 Units	TRC-M120	Determine if the main verb is a Controlled Action Verb	Parameterized - Pattern matching	
		TRC-M225	Avoid Unrelated Terms	Parameterized - Term tag	
Concision	R07 Avoid Vague Terms	TRC-M630	Enforce the Use of Define Terms by avoiding Synonyms	Non-parameterized	
	R08 No Escape Clauses	TRC-M020	Avoid the use of indefinite Articles	Parameterized - Term tag	
		TRC-M130	Avoid the use of indefinite Articles before Entry	Parameterized - Pattern matching	
	R09 No Open Ended	TRC-M140	Ensure Numbers are followed by Units or noun qualifications	Parameterized - Pattern group and pattern matching	
		TRC-M150	Detect inadequate Unit for a Characteristic	Parameterized - Relationships not SCM compliant	
	R10 Superfluous Infinitives	TRC-M160	Avoid making up different measurement systems	Measurement units consistency metric	
		TRC-M090	Avoid the use of vague Verbs	Parameterized - Cluster	
	R11 Separate Clauses	TRC-M170	Avoid the use of vague Adjectives	Parameterized - Cluster	
		TRC-M180	Avoid the use of Vague Adverbs	Parameterized - Cluster	
	Non-ambiguity	R12 Correct Grammar	TRC-M900	Avoid the use of Vague Terms	Parameterized - Cluster
R13 Correct Spelling		TRC-M190	Avoid the use of Escape clauses	Parameterized - Special Sentences	
R14 Correct Punctuation		TRC-M200	Avoid the use of Open Ended clauses	Parameterized - Special Sentences	
R15 Logical Condition		TRC-M210	Avoid the use of Superfluous infinitives	Parameterized - Cluster	
		TRC-M010	Enforce the use of a complete structure sentence	Non-parameterized	
R16 Avoid Not		TRC-M215	Check the number of condition clauses	Parameterized - Cluster	
		TRC-M230	Avoid inadequate grammar structures	Parameterized - Pattern group matching	
R17 Idiomatic		TRC-M240	Avoid incorrect spelling	Non-parameterized	
		TRC-M250	Facilitate readability	Non-parameterized	
R18 Single Sentence		TRC-M260	Review incorrect punctuation	Non-parameterized	
	TRC-M270	Set a convention for logical expression forms	Parameterized - Cluster		
Singularity	R19 Avoid Combinators	TRC-M340	Avoid the use of combinators out of the condition part	Parameterized - Cluster	
	R20 Avoid Purpose	TRC-M280	Avoid the use of Not and other negative expressions	Parameterized - Cluster	
		TRC-M285	Avoid the use of negative Expressions out of the condition part	Parameterized - Cluster	
	R21 Avoid Parentheses	TRC-M300	Avoid the use of oblique Symbol "/"	Parameterized - Term tag	
		TRC-M320	Check the text length by counting paragraphs	Non-parameterized	
	R22 Avoid Redundant	TRC-M330	Check the text length by counting words	Non-parameterized	
		TRC-M340	Control the number of Action Verbs out of the condition part	Parameterized - Pattern matching	
	R23 Use Of Headings	TRC-M360	Check the number of Modal Verbs	Parameterized - Term tag	
		TRC-M350	Avoid the use of Combinators out of the condition part	Parameterized - Cluster	
	Completeness	R24 Avoid Absolutes	TRC-M370	Multiple subject detection	Parameterized - Pattern matching
R25 Use Of Headings		TRC-M375	Multiple verb detection	Parameterized - Pattern matching	
		TRC-M380	Avoid phrases that indicate the purpose	Parameterized - Special Sentences	
R26 Avoid Redundant		TRC-M390	Avoid the use of Parentheses out of the condition part	Parameterized - Cluster	
		TRC-M070	Avoid the use of Pronouns to refer to nouns	Parameterized - Term tag	
R27 Explicit		TRC-M010	Enforce the use of a complete structure sentence	Non-parameterized	
		TRC-M070	Avoid the use of Pronoun to refer to nouns	Parameterized - Term tag	
R28 Avoid Ambiguity		TRC-M410	Avoid unresolvable absolutes expressions impossible to verify	Parameterized - Special Sentences	
		TRC-M225	Ensure tolerance values are in an adequate value range	Parameterized - Custom-code	
Conditions		R29 Explicit Lists	TRC-M360	Check the number of Modal Verbs	Parameterized - Term tag
	R30 Classify	TRC-M400	Avoid a list of actions after a condition activation	Parameterized - Pattern matching	
	R31 Express Once	TRC-M455	Avoid a list of conditions after an action activation	Parameterized - Pattern matching	
	R32 Solutionflow	TRC-M460	Enforce attribute type is not empty	Parameterized - Attribute	
		TRC-M480	Avoid overlapping between the requirements	Overlapping consistency metric	
	R33 Universals	TRC-M490	Avoid starting a solution	Parameterized - Cluster	
		TRC-N000	Avoid the use of Flow sentences	Parameterized - Cluster	
	Quantification	R34 Measurable	TRC-M510	Avoid the use of Ambiguous Universal Keywords	Parameterized - Cluster
		R35 Temporal Indefinite	TRC-M520	Force to include tolerance value for the units that required tolerance	Parameterized - Pattern group and pattern matching
			TRC-M525	Ensure tolerance values are in an adequate value range	Parameterized - Custom-code
R36 Value Range		TRC-M530	Confirms the value for a property is in a controlled range	Properties consistency metric	
		TRC-M540	Avoid the usage of Imprecise Quantifiers	Parameterized - Cluster	
R37 Use Consistent Terms		TRC-M045	Avoid the usage of Imprecise Quantifiers applied to a property	Parameterized - Pattern matching	
		TRC-M560	Avoid the use of temporal indefinite keywords out of the condition part	Parameterized - Cluster	
Uniformity of Language		R38 Define Acronyms	TRC-M630	Enforce the use of Define Terms by avoiding Synonyms	Non-parameterized
		R39 Avoid Abbreviations	TRC-M580	Avoid the use of unknown acronyms	Non-parameterized
		R40 Style Guide	TRC-N090	Avoid the use of unknown abbreviations	Non-parameterized
	R41	TRC-M010	Enforce the use of a complete structure sentence	Non-parameterized	

Mapping INCOSE 2019 rules per characteristic with TRC tool metrics

Mapping approach types examples:

Direct approach

☐ R13	☐ Correct Spelling	☐ TRC-M240	☐ Avoid Incorrect spelling
-------	--------------------	------------	----------------------------

Progressive

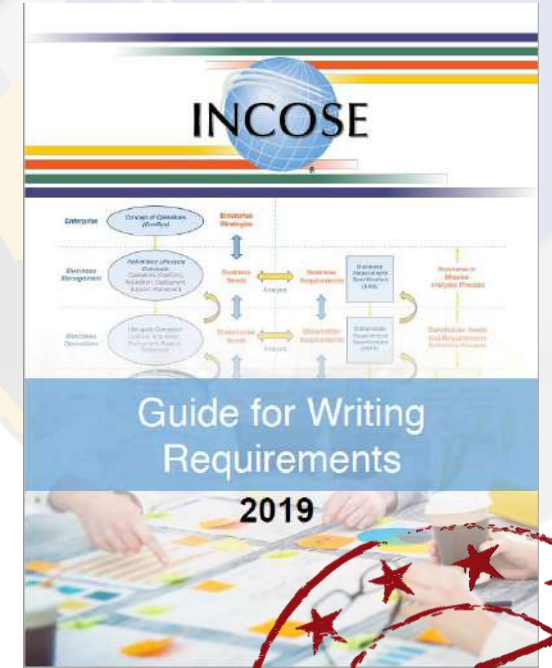
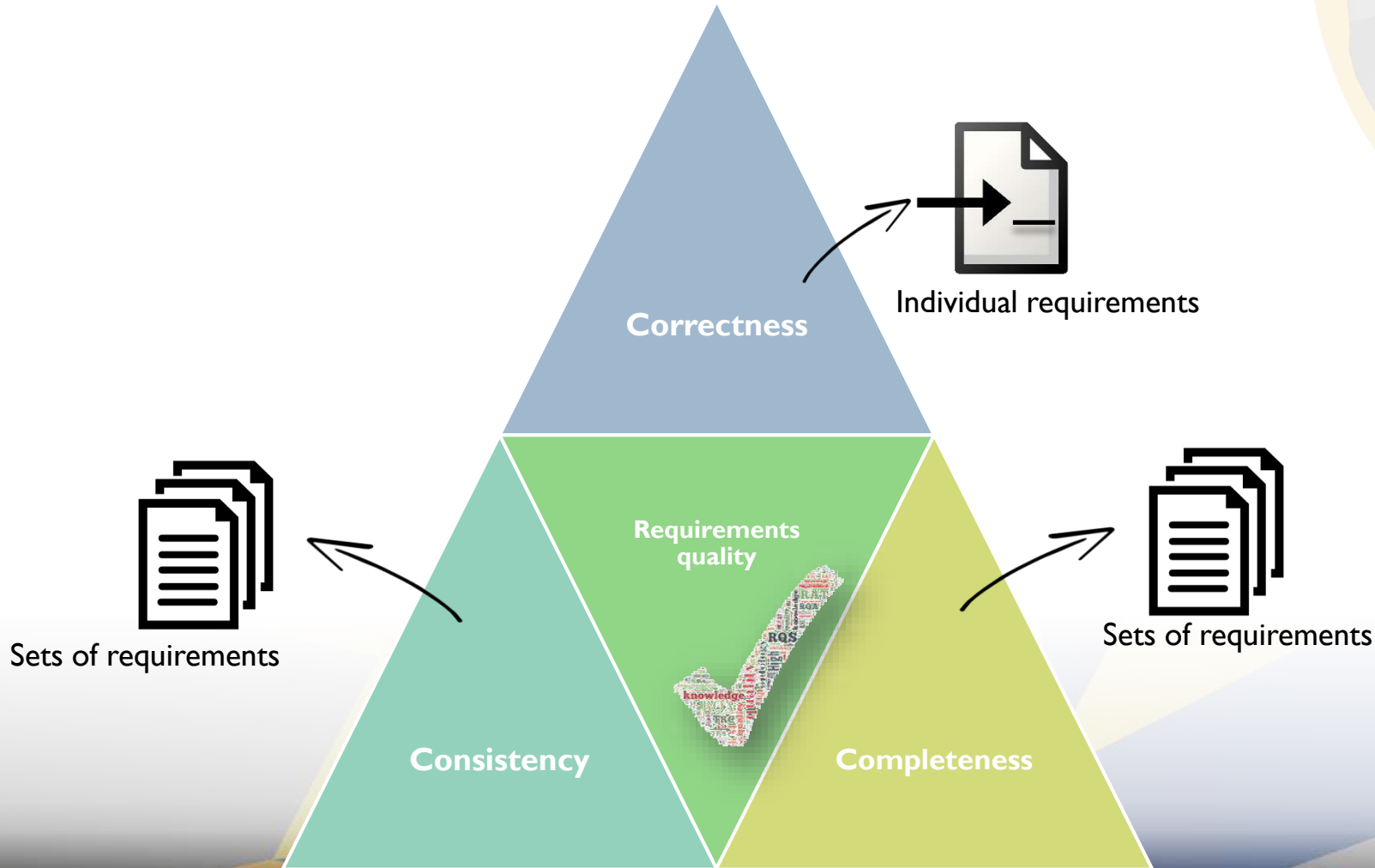
☐ R02	☐ Use Active Voice	☐ TRC-M030	☐ Avoid the use of Passive Voice
		☐ TRC-M035	☐ Avoid the use of Passive Voice after the modal verb
		☐ TRC-M040	☐ Avoid the use of Passive Voice out of the condition part

Indirect or multiple

☐ R07	☐ Avoid Vague Terms	☐ TRC-M060	☐ Avoid the use of Vague Verbs
		☐ TRC-M170	☐ Avoid the use of Vague Adjectives
		☐ TRC-M180	☐ Avoid the use of Vague Adverbs
		☐ TRC-M950	☐ Avoid the use of Vague Terms
☐ R18	☐ Single Sentence	☐ TRC-M320	☐ Check the text length by counting paragraphs
		☐ TRC-M330	☐ Check the text length by counting words
		☐ TRC-M340	☐ Control the number of Action Verbs out of the condition part
		☐ TRC-M360	☐ Check the number of Modal Verbs

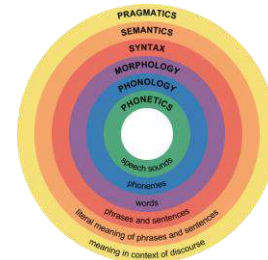
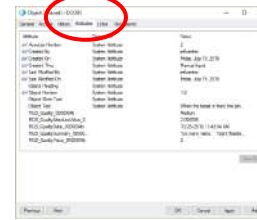
Requirements quality metrics: CCC Approach

- 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



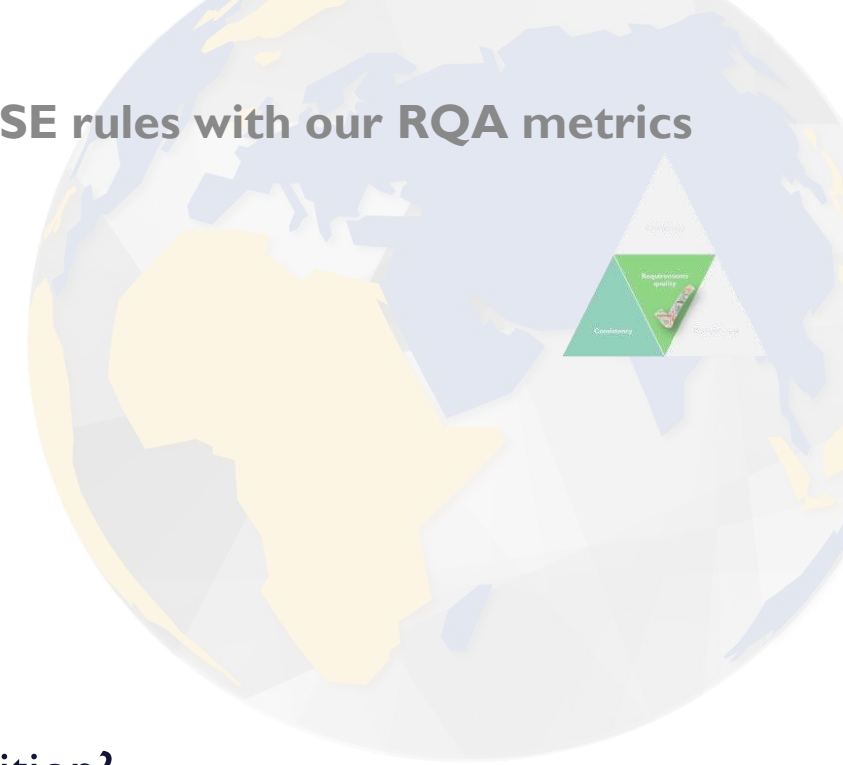
Examples of requirements metrics: Completeness

- 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: $12V \pm 0.5V$



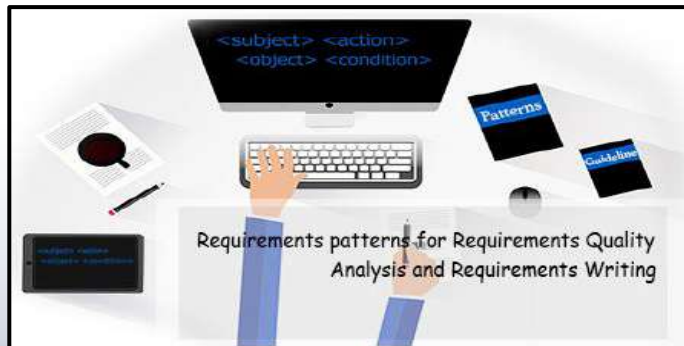
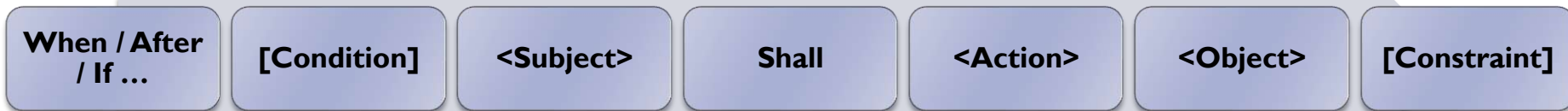
Examples of requirements metrics: Consistency

- 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



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 helps ensure the requirement 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

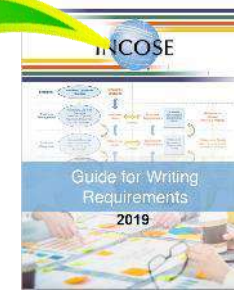
Avoid combinators.

Elaboration:

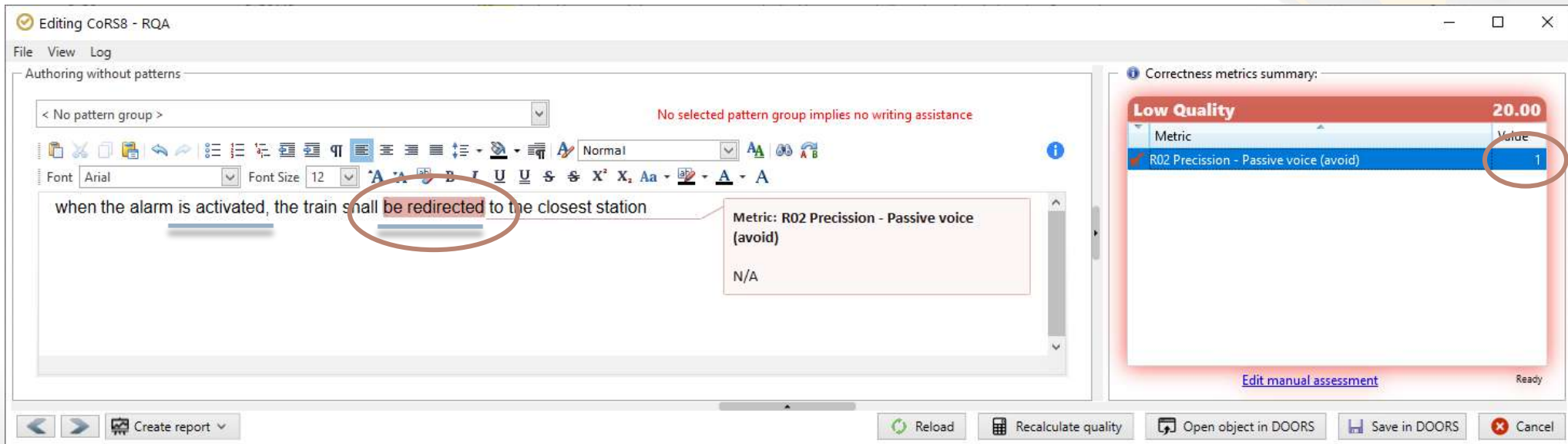
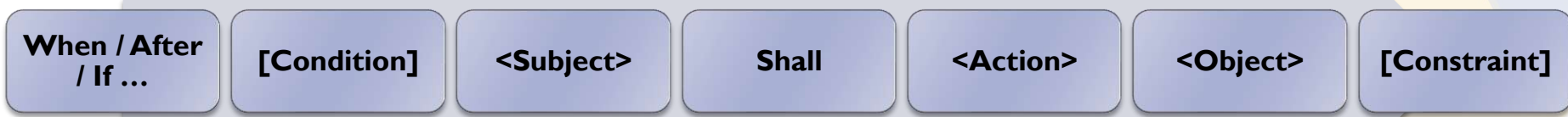
Combinators are words that join clauses, such as “and”, “or”, “then”, “unless”, “but”, “as well as”, “but also”, “however”, “whether”, “meanwhile”, “whereas”, “on the other hand”, and “otherwise.” Their presence in a requirement usually indicates that multiple requirements should be written.

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



Editing CoRS8 - RQA

File View Log

Authoring without patterns

< No pattern group >

No selected pattern group implies no writing assistance

Font Arial Font Size 12

when the alarm is activated, the train shall be redirected to the closest station

Metric: R02 Precision - Passive voice (avoid)
N/A

Correctness metrics summary: **Low Quality** 20.00

Metric	Value
R02 Precision - Passive voice (avoid)	1

Edit manual assessment Ready

Create report Reload Recalculate quality Open object in DOORS Save in DOORS Cancel

Advanced semantic techniques

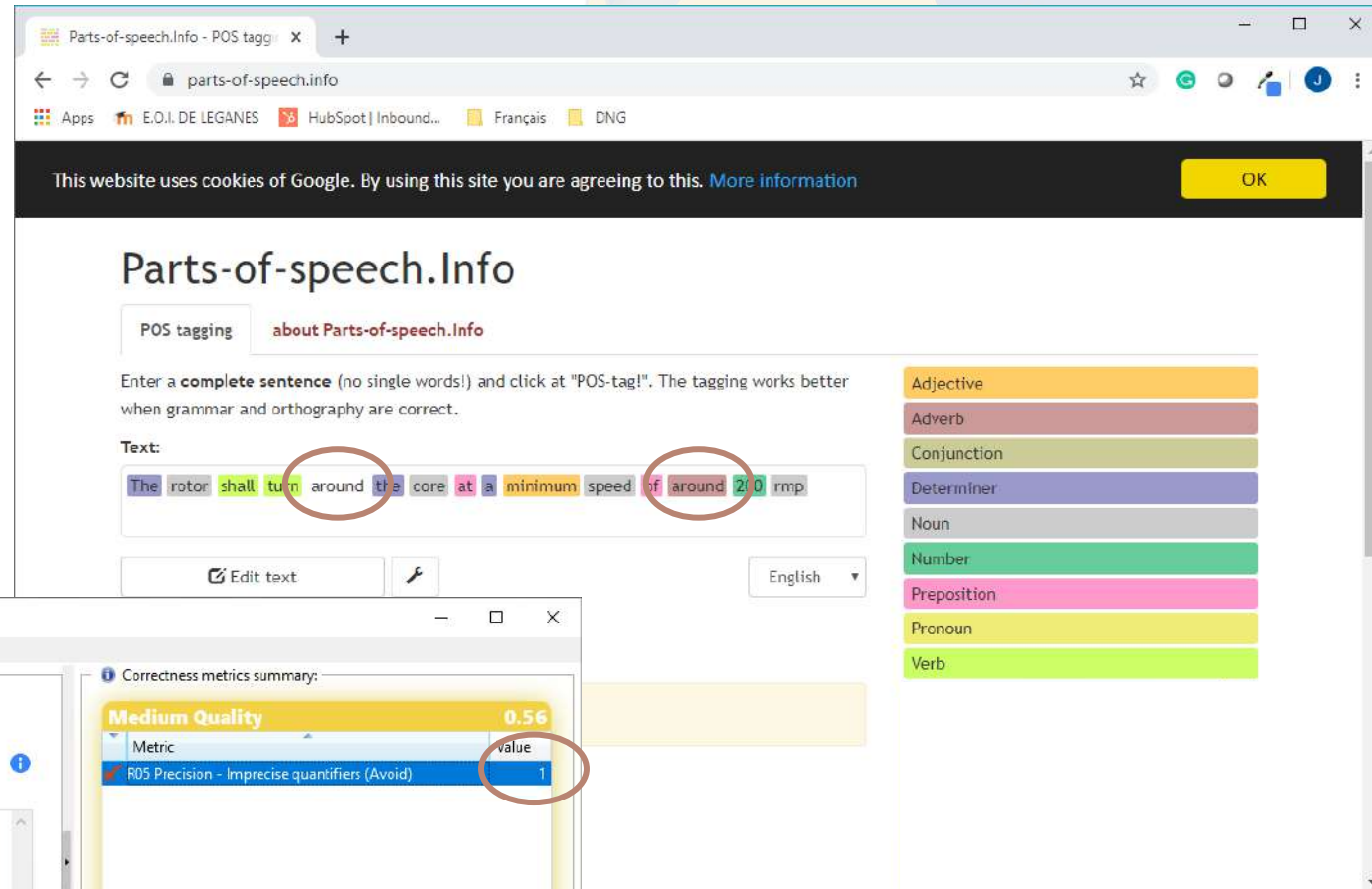
System Requirements Spec.

...

SyR-088: xxx

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

SyR-090: xxx



Parts-of-speech.info - POS tagging

parts-of-speech.info

This website uses cookies of Google. By using this site you are agreeing to this. [More information](#) OK

Parts-of-speech.info

POS tagging [about Parts-of-speech.info](#)

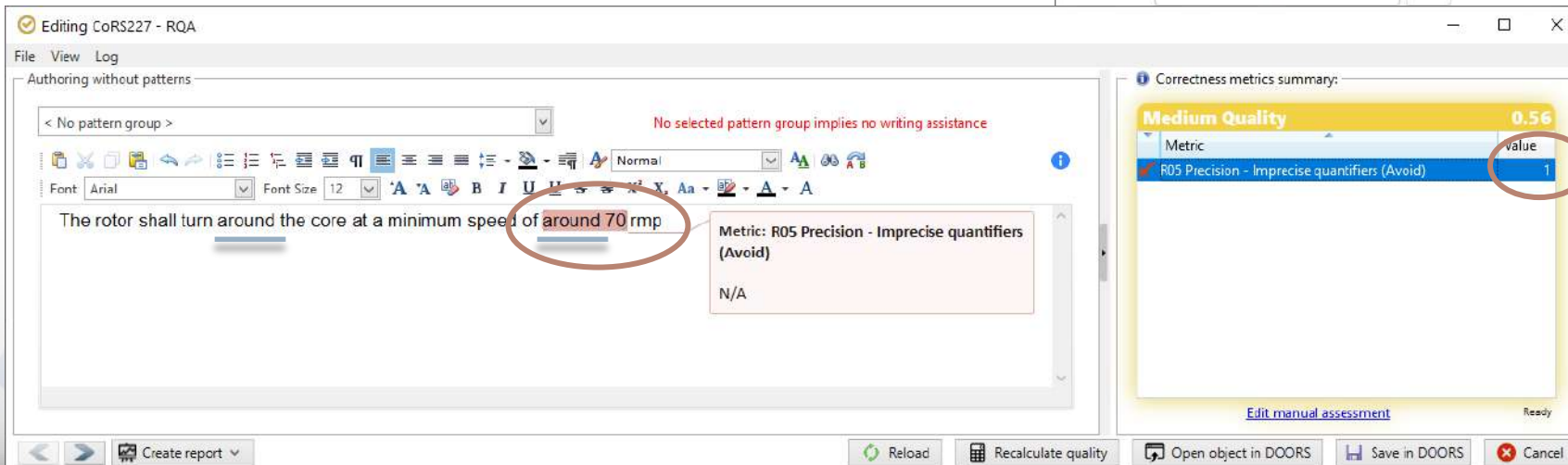
Enter a **complete sentence** (no single words!) and click at "POS-tag!". The tagging works better when grammar and orthography are correct.

Text:

The rotor shall turn around the core at a minimum speed of around 20 rmp

Edit text English

- Adjective
- Adverb
- Conjunction
- Determiner
- Noun
- Number
- Preposition
- Pronoun
- Verb



Editing CoRS227 - RQA

File View Log

Authoring without patterns

< No pattern group >

No selected pattern group implies no writing assistance

Font Arial Font Size 12

The rotor shall turn around the core at a minimum speed of around 70 rmp

Metric: R05 Precision - Imprecise quantifiers (Avoid)
N/A

Correctness metrics summary:

Medium Quality 0.56

Metric	value
✓ R05 Precision - Imprecise quantifiers (Avoid)	1

[Edit manual assessment](#) Ready

Reload Recalculate quality Open object in DOORS Save in DOORS Cancel



the

REUSE

company

