



# Webinar

**Host:** José Fuentes

## Content

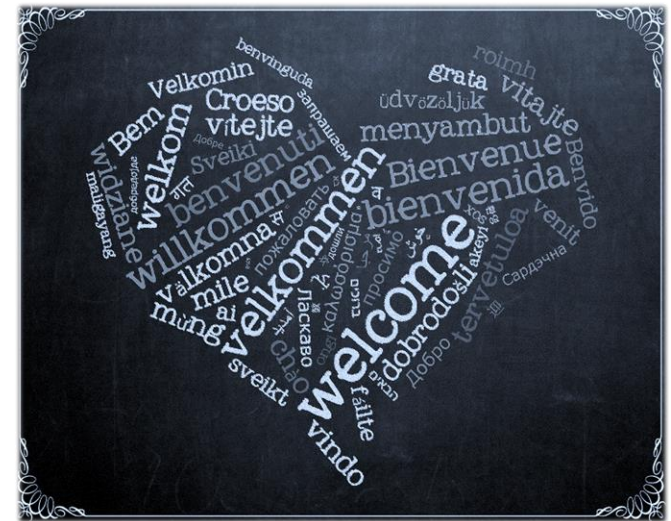
- Introduction to TRC and RQS
- What is a requirement pattern
- Patterns for writing requirements
- Patterns and quality metrics
- Patterns for knowledge elicitation
- Patterns for requirements elicitation
- Live demo
- Q&A



# Introduction



@ReuseCompany



## The REUSE Company (TRC)

### Knowledge Centric Systems Engineering

The REUSE Company is specialized in the application of **Semantic Analysis Technologies** to a wide range of industries (Aerospace, Defense, Automotive, Railway, Energy...)

Our main focus is on System/Software **Traceability, Reuse and Quality**. The integration of tools and technology from The REUSE Company facilitates the representation, analysis and exploitation of knowledge allowing for a knowledge-centric systems engineering approach.

Our mission is to promote system/software and knowledge reuse within any organization, by offering processes, methods, tools and services that make it possible. We offer technology that is fully integrated within the organization's production chain.

## Innovative technologies applied to Systems Engineering

### TRC main Customers

#### Aerospace and Defense



#### Automotive



#### Energy



#### Consulting



#### Banking



#### Health care



#### Other industries



## TRC - Our competences



T<sub>(he)</sub> R<sub>(euse)</sub> Q<sub>(ompany)y</sub>

**Trace + Retrieval + Quality**

**Towards systematic Reuse**

By means of : **Repositories** containing **Ontologies and Assets**



## RQS – Requirements Quality Suite

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



- **Requirements Quality Analyzer (RQA):** to setup, check and manage the quality of a requirements specification
- **Requirement 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...



*patterns*



## Requirements quality metrics: knowledge needs

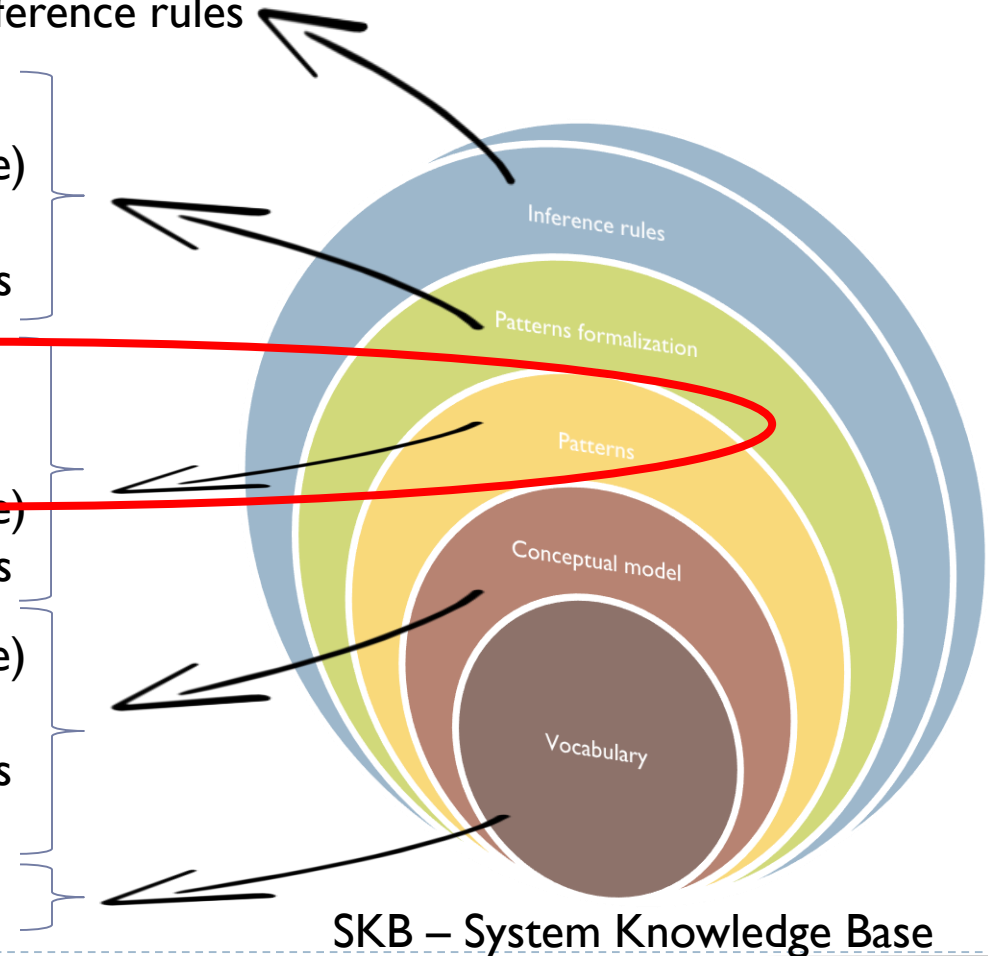
All RQS Metrics are mapped as Inference rules

Consistency metrics (most)  
Completeness metrics (some)  
Correctness metrics (few)  
Semantic retrieval capabilities

Writing assistance (RAT)  
Correctness metrics (few)  
Completeness metrics (some)  
Semantic retrieval capabilities

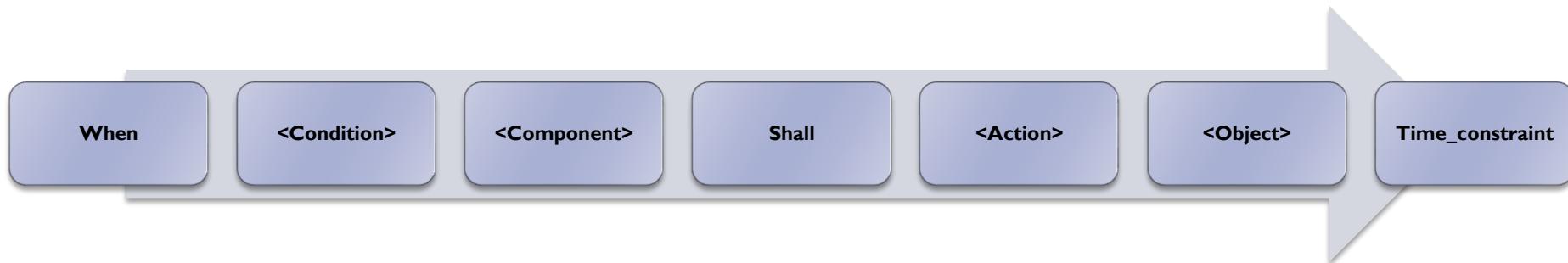
Completeness metrics (some)  
Correctness metrics (few)  
Semantic retrieval capabilities  
Writing assistance

CCC Metrics (most)



## Requirements patterns

- Represented as a sequential set of *restrictions: placeholders* (aka *slots*)
- Grammar for a specific language
- Used for:
  - Requirements writing
  - Create new quality metrics
  - Improve accuracy of other quality metrics
  - Knowledge elicitation
  - Requirements elicitation



## Requirements patterns: slots (content)

- Term – Obligation to have a Normalized Term at a particular position of the sentence
  - Its format representation is first character of the term in capital and the rest in lowercase.
- Term tag (Syntax)– Obligation to have a Normalized Term of a particular term tag at a particular position of the sentence
  - The format representation is to write it in uppercase.
- Semantic cluster – Obligation to have a Normalized Term of a particular semantic cluster at a particular position of the sentence
  - Its format representation is to write it is like the Terms but delimited with the '«' '»' symbols.

Wheel

NOUN

«System  
Element»

## Requirements patterns: slots (content)

- Term tag (Syntax) + Semantic cluster - Obligation to have a Normalized Term of a particular term tag AND a particular semantic cluster at a particular position of the sentence
  - Its format representation is the combination of term tags and semantic clusters separated by the | symbol.
- Sub-pattern – Obligation of a part of the document to match a sub-pattern
  - Its format representation is the first character of the pattern name in uppercase and the rest in lowercase and it's delimited by the square brackets.
- Part of Speech (POS) – Obligation to have a Normalized Term where the user input is a POS at a particular position of the sentence
  - Its format representation is the text delimited by quotes.
  - E.g.: “is”
  - Is compared against the original Input text and the normalized input text

NOUN +  
«SIGNAL»

[Condition]



## Requirements patterns: slots configuration

Select the new slot restriction to be added:

Select your desired slot restriction to be added:

☐ Text or Part Of Speech:  
Text:

☐ Term:  
Term:     
Example:

☒ Term Tag:  
Term tag:     
Example term:

☒ Cluster:  
Cluster:     
Example term:

☐ Pattern:  
Pattern group:     
Pattern:

AND

ADD a new slot

MOVE slots

«LY Adeverbs»

ADVERB

VERB

or

A couple of

automatically

Automatically

selected

Select

+

VERB

+

+

fly

Fly

VERB

ACTION

Verbiform: Ground

Person: Invariant

Hide Advanced Setup

Advanced slot setup:

☐ Optional

☐ Wildcard

☐ Relationship Type Restriction:

☒ Can formalize as occurrence

Excluded Terms

Can

Do

Must

Shall

Will

Normalization rules

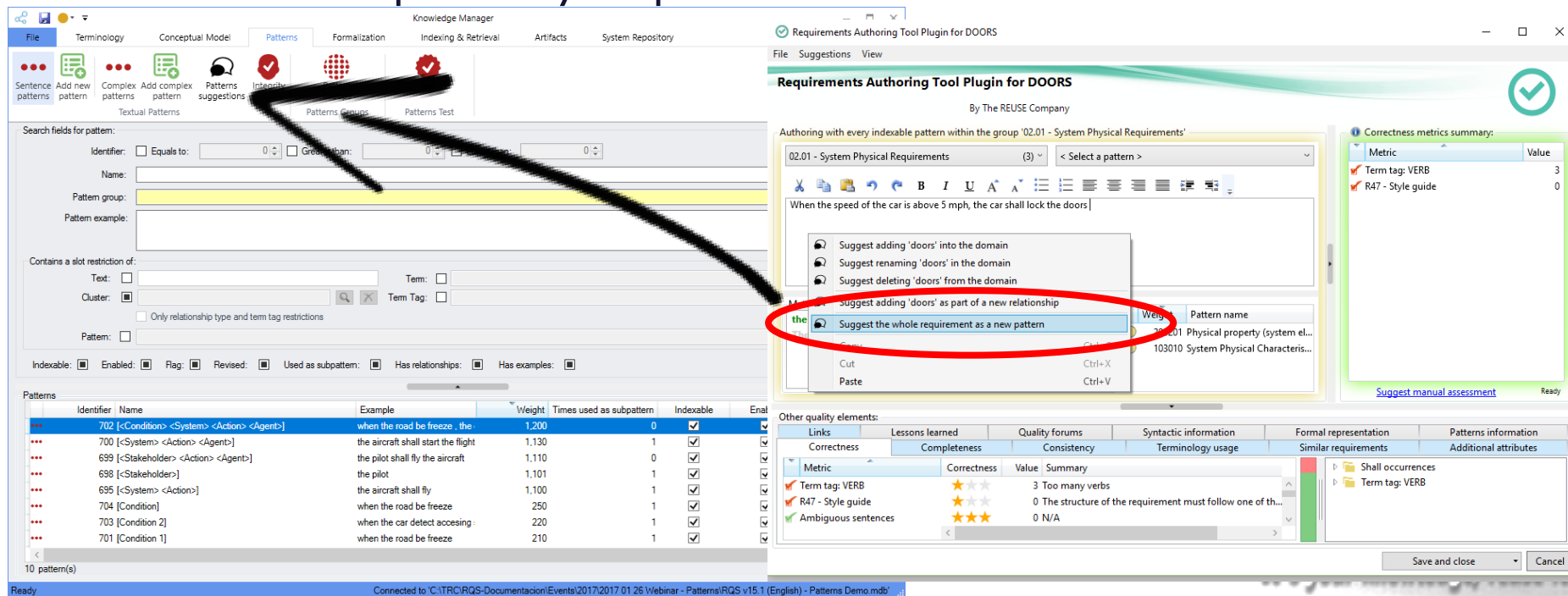
INFINITIVE (OR ENG. NOUN) FROM ...

INFINITIVE FROM PARTICIPLE/PRE...



## Requirements patterns

- Some included OOTB with RQS Server installation
- Other included in specific RQS Libraries
- Managed in Knowledge Manager
- Managed by Knowledge Architect...
- ... but also requested by Requirements Authors



The screenshot displays the 'Requirements Authoring Tool Plugin for DOORS' interface. The 'Patterns' tab is active, showing a list of patterns with columns for Identifier, Name, Example, Weight, Times used as subpattern, Indexable, and Enabled. A context menu is open over a requirement, with the option 'Suggest adding 'doors' as a new pattern' highlighted. A red circle is drawn around this option. A black arrow points from the 'Patterns' tab in the top navigation bar to the context menu.

**Patterns List:**

Identifier	Name	Example	Weight	Times used as subpattern	Indexable	Enabled
702	[<Condition> <System> <Action> <Agent>]	when the road be freeze, the	1,200	0	✓	✓
700	[<System> <Action> <Agent>]	the aircraft shall start the flight	1,130	1	✓	✓
699	[<Stakeholder> <Action> <Agent>]	the pilot shall fly the aircraft	1,110	0	✓	✓
698	[<Stakeholder>]	the pilot	1,101	1	✓	✓
695	[<System> <Action>]	the aircraft shall fly	1,100	1	✓	✓
704	[Condition]	when the road be freeze	250	1	✓	✓
703	[Condition 2]	when the car detect accessing	220	1	✓	✓
701	[Condition 1]	when the road be freeze	210	1	✓	✓

**Context Menu Options:**

- Suggest adding 'doors' into the domain
- Suggest renaming 'doors' in the domain
- Suggest deleting 'doors' from the domain
- Suggest adding 'doors' as part of a new relationship
- Suggest the whole requirement as a new pattern**

**Correctness metrics summary:**

Metric	Value
Term tag: VERB	3
R47 - Style guide	0

**Other quality elements:**

Metric	Correctness	Value	Summary
Term tag: VERB	★ ★ ★	3	Too many verbs
R47 - Style guide	★ ★ ★	0	The structure of the requirement must follow one of th...
Ambiguous sentences	★ ★ ★	0	N/A

# Requirements

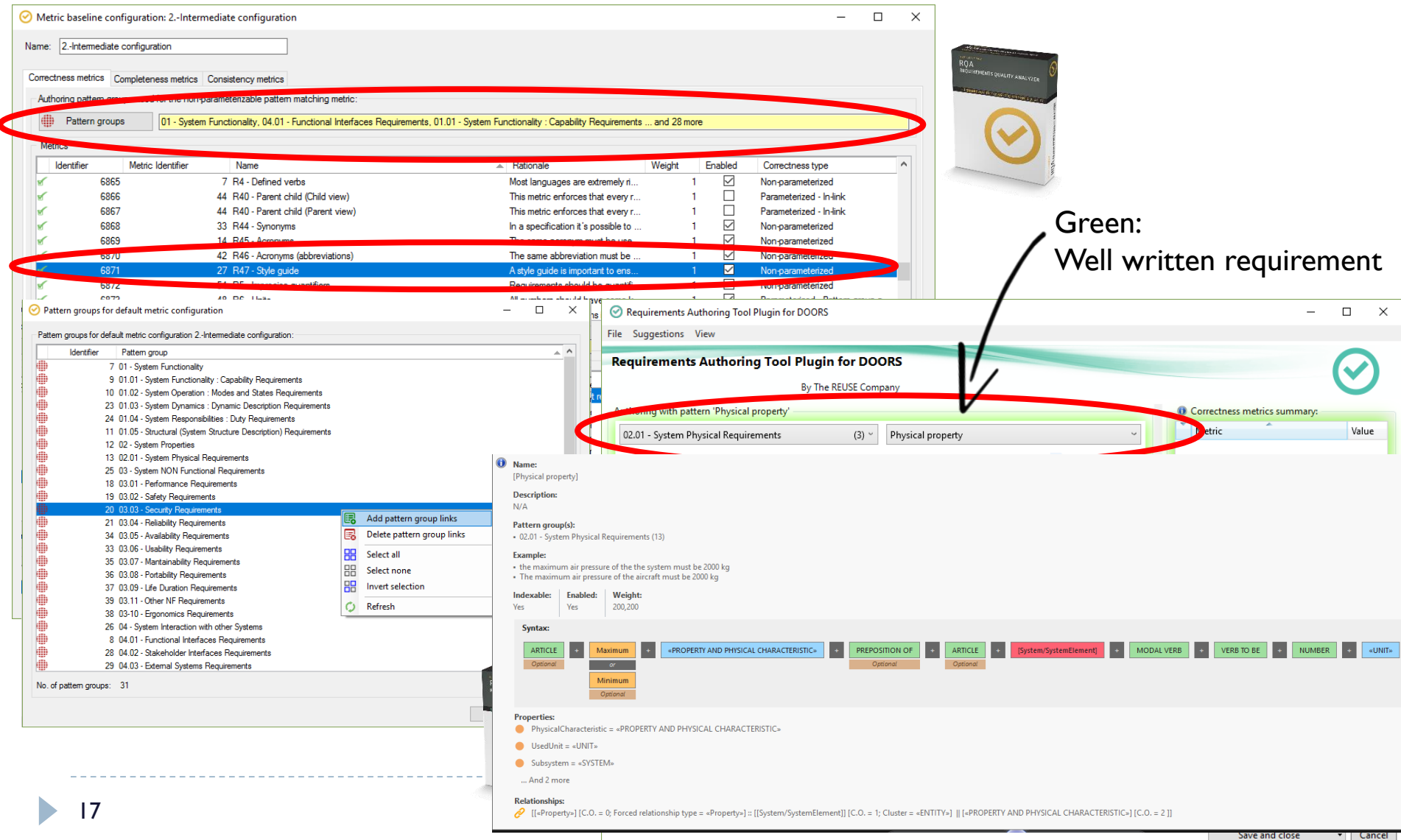
## Writing



## Requirements patterns for authoring

- 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 (safety, acquisition, ISO26262, EARS...)

## Requirements patterns for authoring



**Metric baseline configuration: 2.-Intermediate configuration**

Name: 2.-Intermediate configuration

Correctness metrics | Completeness metrics | Consistency metrics

Authoring pattern groups: 01 - System Functionality, 04.01 - Functional Interfaces Requirements, 01.01 - System Functionality : Capability Requirements ... and 28 more

Identifier	Metric Identifier	Name	Rationale	Weight	Enabled	Correctness type
6865	7	R4 - Defined verbs	Most languages are extremely ri...	1	<input checked="" type="checkbox"/>	Non-parameterized
6866	44	R40 - Parent child (Child view)	This metric enforces that every r...	1	<input type="checkbox"/>	Parameterized - In-link
6867	44	R40 - Parent child (Parent view)	This metric enforces that every r...	1	<input type="checkbox"/>	Parameterized - In-link
6868	33	R44 - Synonyms	In a specification it's possible to ...	1	<input checked="" type="checkbox"/>	Non-parameterized
6869	14	R45 - Acronyms	The same acronym must be use...	1	<input checked="" type="checkbox"/>	Non-parameterized
6870	42	R46 - Acronyms (abbreviations)	The same abbreviation must be ...	1	<input checked="" type="checkbox"/>	Non-parameterized
6871	27	R47 - Style guide	A style guide is important to ens...	1	<input checked="" type="checkbox"/>	Non-parameterized
6872	54	R5 - Inclusion in interface	Requirements should be avail...	1	<input type="checkbox"/>	Non-parameterized
6873	48	R6 - Link...	AB...	1	<input type="checkbox"/>	Non-parameterized

**Pattern groups for default metric configuration**

Identifier	Pattern group
7	01 - System Functionality
9	01.01 - System Functionality : Capability Requirements
10	01.02 - System Operation : Modes and States Requirements
23	01.03 - System Dynamics : Dynamic Description Requirements
24	01.04 - System Responsibilities : Duty Requirements
11	01.05 - Structural (System Structure Description) Requirements
12	02 - System Properties
13	02.01 - System Physical Requirements
25	03 - System NON Functional Requirements
18	03.01 - Performance Requirements
19	03.02 - Safety Requirements
20	03.03 - Security Requirements
21	03.04 - Reliability Requirements
34	03.05 - Availability Requirements
33	03.06 - Usability Requirements
35	03.07 - Maintainability Requirements
36	03.08 - Portability Requirements
37	03.09 - Life Duration Requirements
39	03.11 - Other NF Requirements
38	03.10 - Ergonomics Requirements
26	04 - System Interaction with other Systems
8	04.01 - Functional Interfaces Requirements
28	04.02 - Stakeholder Interfaces Requirements
29	04.03 - External Systems Requirements

No. of pattern groups: 31

**Requirements Authoring Tool Plugin for DOORS**

File | Suggestions | View

By The REUSE Company

Working with pattern 'Physical property'

02.01 - System Physical Requirements (3) | Physical property

Correctness metrics summary:

Metric	Value
02.01 - System Physical Requirements	Physical property

**Name:** [Physical property]

**Description:** N/A

**Pattern group(s):** 02.01 - System Physical Requirements (13)

**Example:**

- the maximum air pressure of the the system must be 2000 kg
- The maximum air pressure of the aircraft must be 2000 kg

**Indexable:** Yes | **Enabled:** Yes | **Weight:** 200,200

**Syntax:**

Optional: ARTICLE + Maximum + «PROPERTY AND PHYSICAL CHARACTERISTIC» + PREPOSITION OF + ARTICLE + [System/SystemElement] + MODAL VERB + VERB TO BE + NUMBER + «UNIT»

**Properties:**

- PhysicalCharacteristic = «PROPERTY AND PHYSICAL CHARACTERISTIC»
- UsedUnit = «UNIT»
- Subsystem = «SYSTEM»
- ... And 2 more

**Relationships:**

[[«Property»] [C.O. = 0; Forced relationship type = «Property»] : [[System/SystemElement]] [C.O. = 1; Cluster = «ENTITY»] || [«PROPERTY AND PHYSICAL CHARACTERISTIC»] [C.O. = 2]]

Save and close | Cancel

**Green:**  
Well written requirement

## Requirements patterns for authoring

- How to create a set of patterns:







**Patterns and**  
*quality metrics?*

## Requirements patterns and quality metrics

### ➤ Three mechanisms as *parameterized* metrics in RQA

Metric baseline configuration: 2.-Intermediate configuration

Name: 2.-Intermediate configuration

Correctness metrics | Completeness metrics | Consistency metrics

Authoring pattern groups used for the non-parameterizable pattern matching metric:

Pattern groups: 01 - System Functionality, 04.01 - Functional Interfaces Requirements, 01.01 - System Functionality - Capability Requirements ... and 28 more

Metrics

Identifier	Metric Identifier	Name	Rationale	Weight	Enabled	Correctness type
✓ 6865	7	R4 - Defined verbs	Most languages are extremely ri...	1	<input checked="" type="checkbox"/>	Non-parameterized
✓ 6866	44	R40 - Parent child (Child view)	This metric enforces that every r...	1	<input type="checkbox"/>	Parameterized - In-link
✓ 6867	44	R40 - Parent child (Parent view)	This metric enforces that every r...	1	<input type="checkbox"/>	Parameterized - In-link
✓ 6868	33	Add new parameterized metric				
✓ 6869	1	Edit metric				
✓ 6870	4	Delete parameterized metric(s)				
✓ 6871	2	Enabled				
✓ 6872	5	Pattern groups used for the non-parameterizable pattern matching metric				
✓ 6873	4	Import new metrics based on new special sentences				
✓ 6874	5	Copy to clipboard				

No. of metrics: 75 Enabled: 55

Selected metric ranges

Lower limit	Upper limit	Manda
0	1	False
1	∞	False

No. of ranges: 2

Accept Cancel

Knowledge, reuse it.

## Requirements patterns and quality metrics

- Three mechanisms as **parameterized** metrics in RQA
- Pattern matching and Pattern group matching metrics
- Example: all requirements with type *Safety* must be compliant with a pattern in the *System Safety Requirement* pattern group

Parameterized pattern metric

Metric information:

Name: Pattern: [System Safety Requirement]

Rationale:

Weight: 1

Enabled: ☒

Apply only on requirements holding this expression over an attribute:

Attribute: Type

Type: String

Operator: =

Value: Safety

Pattern parameters:

Pattern: [System Safety Requirement]

☒ Count repetitions

Metric baseline configuration: 2.-Intermediate configuration

Name: 2.-Intermediate configuration

Correctness metrics | Completeness metrics | Consistency metrics

Authoring pattern groups used for the non-parameterizable pattern matching metric:

Pattern groups: 01 - System Functionality, 04.01 - Functional Interfaces Requirements, 01.01 - System Functionality : Capability Requirements ... and 28 more

Metrics

Identifier	Metric Identifier	Name	Rationale	Weight	Enabled	Correctness type
6834	24	Out-links	A requirement must be traced, a...	1	<input type="checkbox"/>	Non-parameterized
6835	45	Out-links: Verifies	This metric enforces that every ...	1	<input type="checkbox"/>	Parameterized - Out-link
6836	26	Out-of-domain concepts	Nouns that are not part of the o...	1	<input checked="" type="checkbox"/>	Non-parameterized
6837	25	Out-of-domain verbs	Verbs that are not part of the on...	1	<input checked="" type="checkbox"/>	Non-parameterized
6838	93	Out-of-SCM terms	Verbs that are not part of the SC...	1	<input type="checkbox"/>	Non-parameterized
6839	43	Part terms	Depending on the specification l...	1	<input type="checkbox"/>	Non-parameterized
6840	47	Pattern: [METRIC: Imprecise prepositions]		1	<input checked="" type="checkbox"/>	Parameterized - Pattern matching
6841	47	Pattern: [Passive detector]		1	<input checked="" type="checkbox"/>	Parameterized - Pattern matching
19492	47	Pattern: [System Safety Requirement]		1	<input checked="" type="checkbox"/>	Parameterized - Pattern matching
6844	53	R1 - Indefinite articles	This rule define the definite artic...	1	<input checked="" type="checkbox"/>	Non-parameterized

No. of metrics: 76 Enabled: 56

Selected metric ranges

Lower limit	Upper limit	Mandatory	Quality level	Summary	Description
0	1	False	Low	Invalid requirement stru...	The requirement is labeled as Safety, however, ...
1	+	False	High		

No. of ranges: 2

High



Medium

Low

0 1 ∞

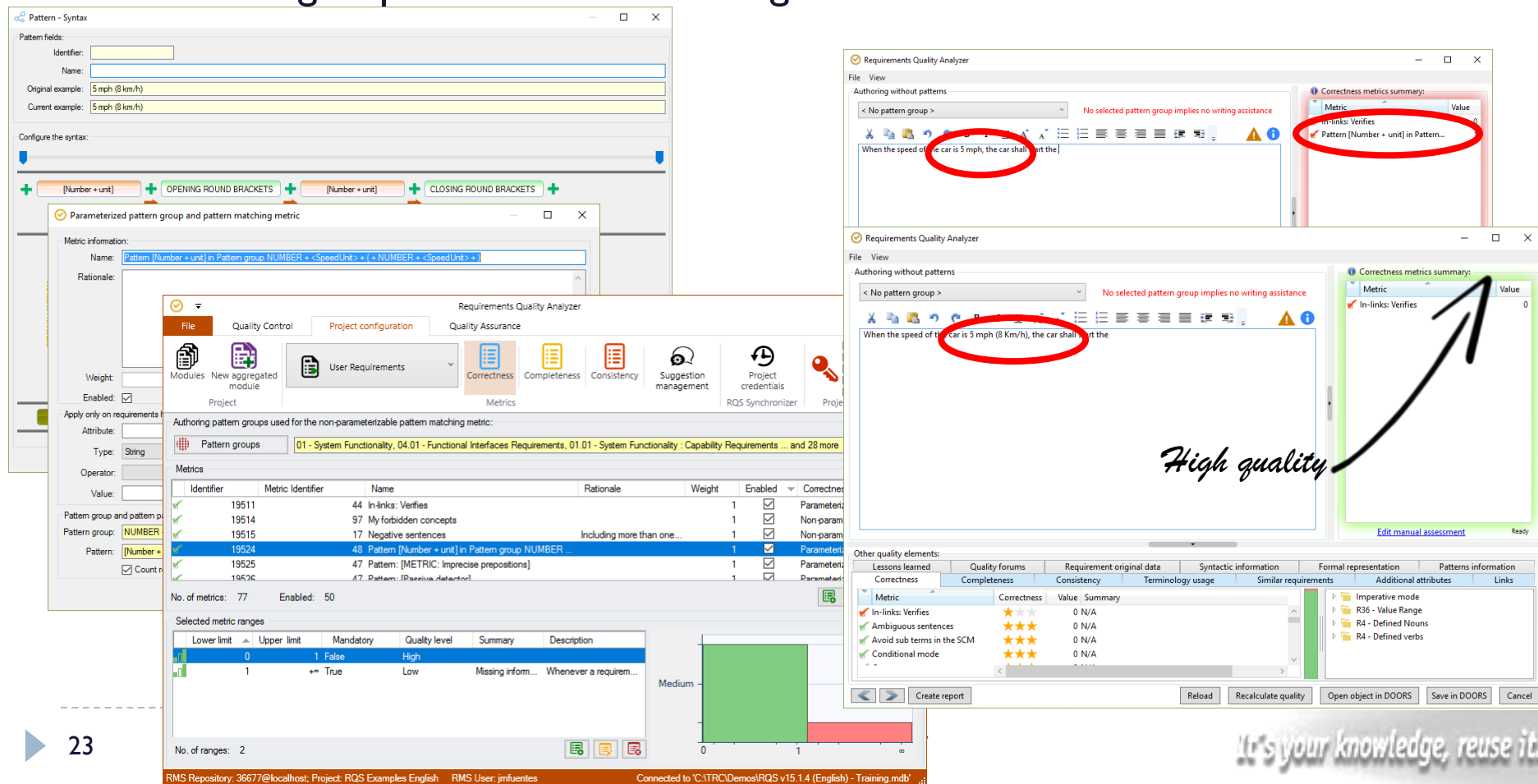
Accept Cancel

## Requirements patterns and quality metrics

- Three mechanisms as ***parameterized*** metrics in RQA
- “Pattern group and Pattern matching” metric
- An example is worth a thousand words:
  - Every time a requirement is using *mph*, it should also use *Km/h*
  - “When the speed of the car is below 5 *mph*, the car shall lock...” 
  - “When the speed of the car is below 5 *mph* (8 *km/h*), the car shall lock...” 
- Solution:
  - A patter to represent: NUMBER + <SpeedUnit>
  - Another patter for: NUMBER + <SpeedUnit> + ( + NUMBER + <SpeedUnit> + )
  - Both in the same pattern group, but the second one with more weight
  - A metric to detect that whenever any of the patterns in the group is discovered, the short one is not welcomed

## Requirements patterns and quality metrics

- Three mechanisms as **parameterized** metrics in RQA
- “Pattern group and Pattern matching” metric



The image displays the Requirements Quality Analyzer (RQA) interface, showing the configuration of a pattern group and the results of a quality analysis.

**Pattern - Syntax** window:

- Identifier: [Empty]
- Name: [Empty]
- Original example: 5 mph (8 km/h)
- Current example: 5 mph (8 km/h)
- Configure the syntax: [Empty]
- Pattern fields: [Number + unit], OPENING ROUND BRACKETS, [Number + unit], CLOSING ROUND BRACKETS

**Parameterized pattern group and pattern matching metric** window:

- Metric information: Name: Pattern [Number + unit] in Pattern group NUMBER + <SpeedUnit> + <[NUMBER + <SpeedUnit>]
- Rationale: [Empty]

**Requirements Quality Analyzer** window:

- File View
- Authoring without patterns: No selected pattern group implies no writing assistance
- Correctness metrics summary: Metric: In-links: Verifies, Value: 0
- Pattern group: 01 - System Functionality, 04.01 - Functional Interfaces Requirements, 01.01 - System Functionality: Capability Requirements ... and 28 more
- Metrics table:

Identifier	Metric Identifier	Name	Rationale	Weight	Enabled	Correctness
19511	44	In-links: Verifies		1	<input checked="" type="checkbox"/>	Parameterized
19514	97	My forbidden concepts		1	<input checked="" type="checkbox"/>	Non-parameterized
19515	17	Negative sentences	Including more than one...	1	<input checked="" type="checkbox"/>	Non-parameterized
19524	48	Pattern [Number + unit] in Pattern group NUMBER		1	<input checked="" type="checkbox"/>	Parameterized
19525	47	Pattern: [METRIC: Imprecise prepositions]		1	<input checked="" type="checkbox"/>	Parameterized
19526	47	Pattern: [Imprecise prepositions]		1	<input checked="" type="checkbox"/>	Parameterized

No. of metrics: 77 Enabled: 50

**Selected metric ranges**

Lower limit	Upper limit	Mandatory	Quality level	Summary	Description
0	1	False	High		
1	+	True	Low	Missing inform...	Whenever a require...

No. of ranges: 2

**Other quality elements**

Metric	Correctness	Value	Summary
In-links: Verifies	★★★★	0	N/A
Ambiguous sentences	★★★★	0	N/A
Avoid sub terms in the SCM	★★★★	0	N/A
Conditional mode	★★★★	0	N/A

**Formal representation**

- Imperative mode
- R36 - Value Range
- R4 - Defined Nouns
- R4 - Defined verbs

**Patterns information**

- Links

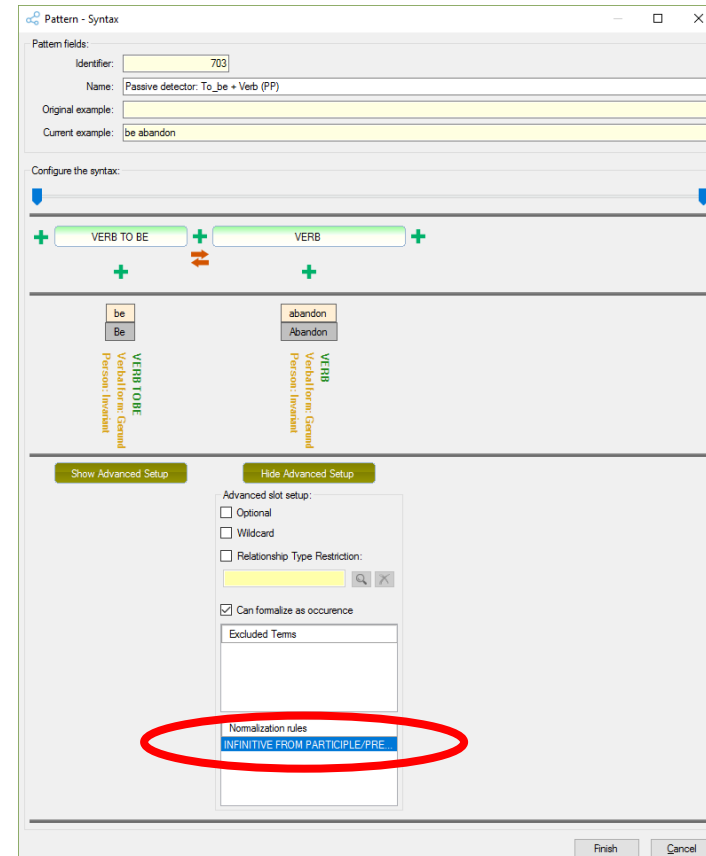
**High quality** (Handwritten note with arrow pointing to the 'In-links: Verifies' metric)

**It's your knowledge, reuse it.**



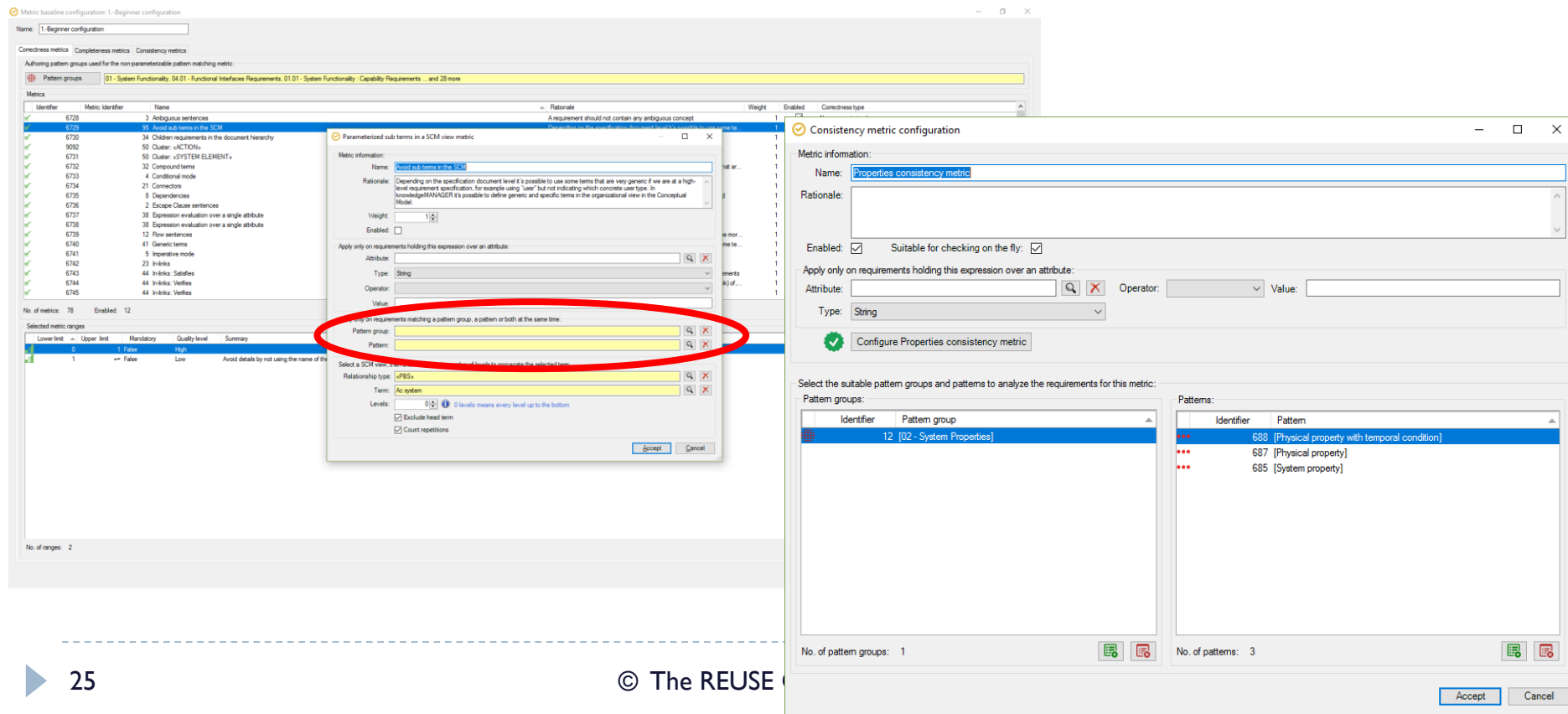
## Patterns to contextualize other metrics

- Example: passive voice detector
- Leading to false positives:
  - “When the car **is started**, the ac system shall send...” ❌
- When replacing by a pattern with more context:
  - Shall + To be + Verb (PP)
- No false positives:
  - “When the car **is started**, the ac system shall send...” ✓
  - “When the car **is started**, the ACTIVE signal **shall be sent**” ❌
- Similar examples solved with patterns:
  - “Part1 shall turn **around** Part2 at a minimum speed of 200 rpm” ✓
  - “The weight of Part1 shall be **around** 2 Kg” ❌



## Patterns to filter out requirements not affected by a metric

- Not all the metrics shall affect to every requirement in a document
- Every *parameterized* metrics can include a pattern in its config screen
- Those requirements not matching the patterns will not be affected



The screenshot displays the REUSE tool's configuration interface for requirements quality analysis. It shows a list of metrics on the left, a central configuration window for a specific metric, and a right-hand panel for pattern selection.

**Metric Configuration (Left):** A table lists various metrics. The 'Ambiguous sentences' metric is highlighted. Below the table, a 'Selected metric ranges' section shows a table with columns for 'Lower limit', 'Upper limit', 'Mandatory', 'Quality level', and 'Summary'.

**Parameterized sub terms in a SCM view metric (Center):** This dialog box allows configuring a specific metric. It includes fields for 'Metric name', 'Rationale', 'Weight', 'Enabled', and 'Apply only on requirements holding this expression over an attribute'. A red circle highlights the 'Pattern' field, which is currently empty.

**Consistency metric configuration (Right):** This panel shows the configuration for the 'Properties consistency metric'. It includes fields for 'Metric information', 'Name', 'Rationale', 'Enabled', 'Suitable for checking on the fly', 'Apply only on requirements holding this expression over an attribute', 'Attribute', 'Type', 'Operator', and 'Value'. A 'Configure Properties consistency metric' button is visible.

**Pattern Selection (Bottom Right):** This section shows the selection of suitable pattern groups and patterns for the metric. It includes a table for 'Pattern groups' and a table for 'Patterns'.

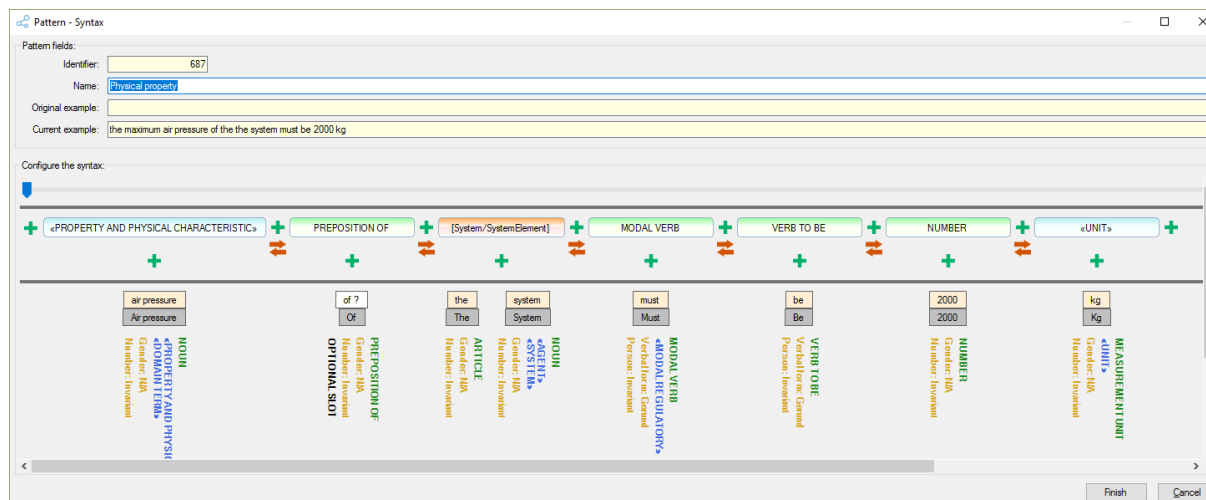
Identifier	Pattern group
12	[D2 - System Properties]

Identifier	Pattern
688	[Physical property with temporal condition]
687	[Physical property]
685	[System property]

At the bottom of the interface, there are buttons for 'Accept' and 'Cancel'.

## Patterns to extract information for completeness and consistency

- Consistency and completeness metrics are mostly based on relationships (semantic graphs) or properties extracted from requirements
- Pattern formalization is the key point to rule the way this information is extracted



- SyR076: “The capacity of the fuel tank shall be 50 gallons”
  - → capacity(‘fuel tank’, 50 gallons) → To be checked against the SKB managed in KM

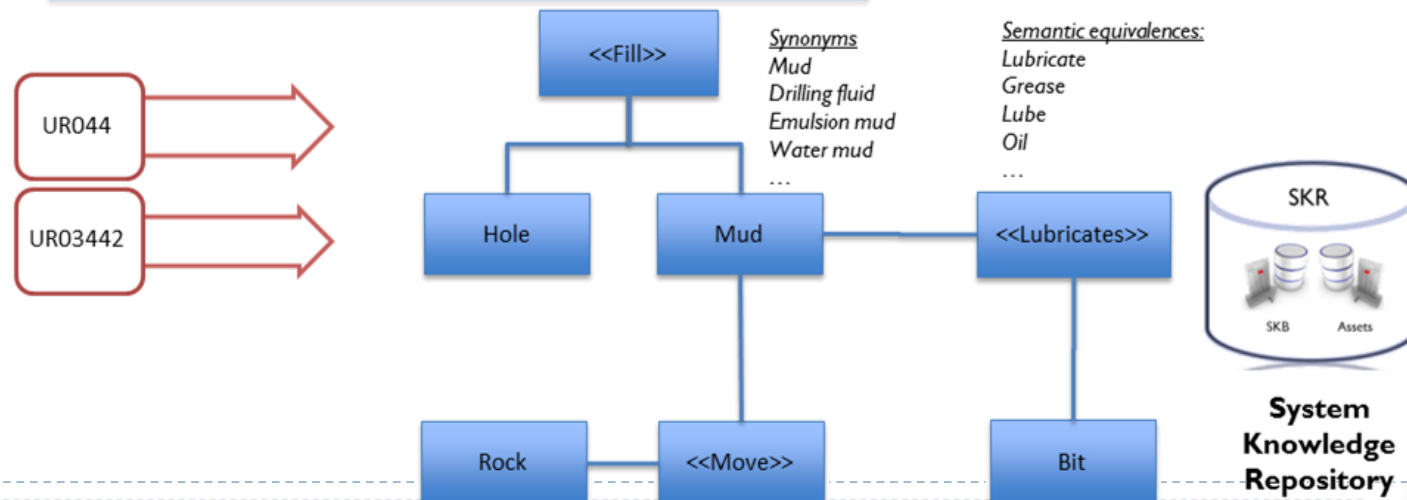
## Patterns for requirements retrieval

### ➤ Metrics and functionality based on semantic search:

- Overlapping requirements
- Missing links
- Requirements retrieval features

UR044 : The hole made **drilling** must be filled with **mud** it **lubricates** the bit and help move the broken rock out of the way.

UR03442 : The **drilling fluid** used to move the broken rock out of the way, must be used to **grease** the bit while **making the hole**.



A close-up photograph of a bumblebee on a yellow flower. The bee is positioned in the center, facing the viewer, with its head and thorax visible. The flower's petals are bright yellow and slightly blurred in the background. The text 'Collecting knowledge' is overlaid on the image, with 'Collecting' in a large, bold, black sans-serif font and 'knowledge' in a smaller, black, cursive script font. The text is enclosed within a thin black rectangular border.

**Collecting**  
knowledge



## Knowledge collection: document analysis (archeology)

- KM+RQA can use patterns to classify some of these common terms:
  - System names
  - Actions

### Step 1

Pattern - Syntax

Pattern fields:

Identifier: 696

Name: ELICITATION: NOUN + Shall

Original example:

Current example: car shall

Configure the syntax:

+ NOUN + MODAL VERB +

+ +

car shall

Car Shall

NOUN MODAL VERB

Gender: MA

Number: Invariant

Verbal form: Gerund

Person: Invariant

Show Advanced Setup

Show Advanced Setup

Finish Cancel

Pattern - Syntax

Pattern fields:

Identifier: 683

Name: VOCABULARY EXTRACTION: <ACTION>

Original example:

Current example: shall fly

Configure the syntax:

+ MODAL VERB + VERB +

+ or +

+ NOUN +

shall fly

Shall Fly

MODAL VERB VERB

Verbal form: Gerund

Person: Invariant

Show Advanced Setup

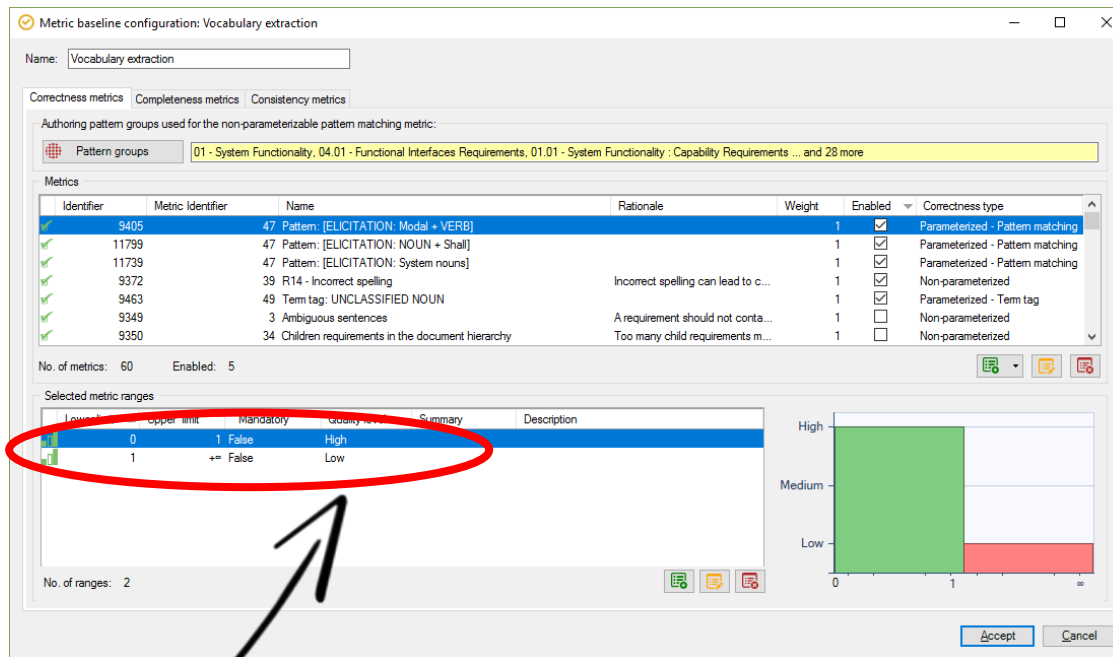
Show Advanced Setup

Finish Cancel



## Knowledge collection: document analysis (archeology)

- KM+RQA can use patterns to classify some of these common terms:
  - System names
  - Actions



Metric baseline configuration: Vocabulary extraction

Name: Vocabulary extraction

Correctness metrics | Completeness metrics | Consistency metrics

Authoring pattern groups used for the non-parameterizable pattern matching metric:

Pattern groups: 01 - System Functionality, 04.01 - Functional Interfaces Requirements, 01.01 - System Functionality : Capability Requirements ... and 28 more

Identifier	Metric Identifier	Name	Rationale	Weight	Enabled	Correctness type
9405	47	Pattern: [ELICITATION: Modal + VERB]		1	<input checked="" type="checkbox"/>	Parameterized - Pattern matching
11799	47	Pattern: [ELICITATION: NOUN + Shall]		1	<input checked="" type="checkbox"/>	Parameterized - Pattern matching
11739	47	Pattern: [ELICITATION: System nouns]		1	<input checked="" type="checkbox"/>	Parameterized - Pattern matching
9372	39	R14 - Incorrect spelling	Incorrect spelling can lead to c...	1	<input checked="" type="checkbox"/>	Non-parameterized
9463	49	Term tag: UNCLASSIFIED NOUN		1	<input checked="" type="checkbox"/>	Parameterized - Term tag
9349	3	Ambiguous sentences	A requirement should not conta...	1	<input type="checkbox"/>	Non-parameterized
9350	34	Children requirements in the document hierarchy	Too many child requirements m...	1	<input type="checkbox"/>	Non-parameterized

No. of metrics: 60    Enabled: 5

Selected metric ranges

Lower	Upper	Mandatory	Summary	Description
0	1	False	High	
1	+	False	Low	

No. of ranges: 2

High  
Medium  
Low

0 1

Accept Cancel

Step 2

*This is not quality checking anymore*



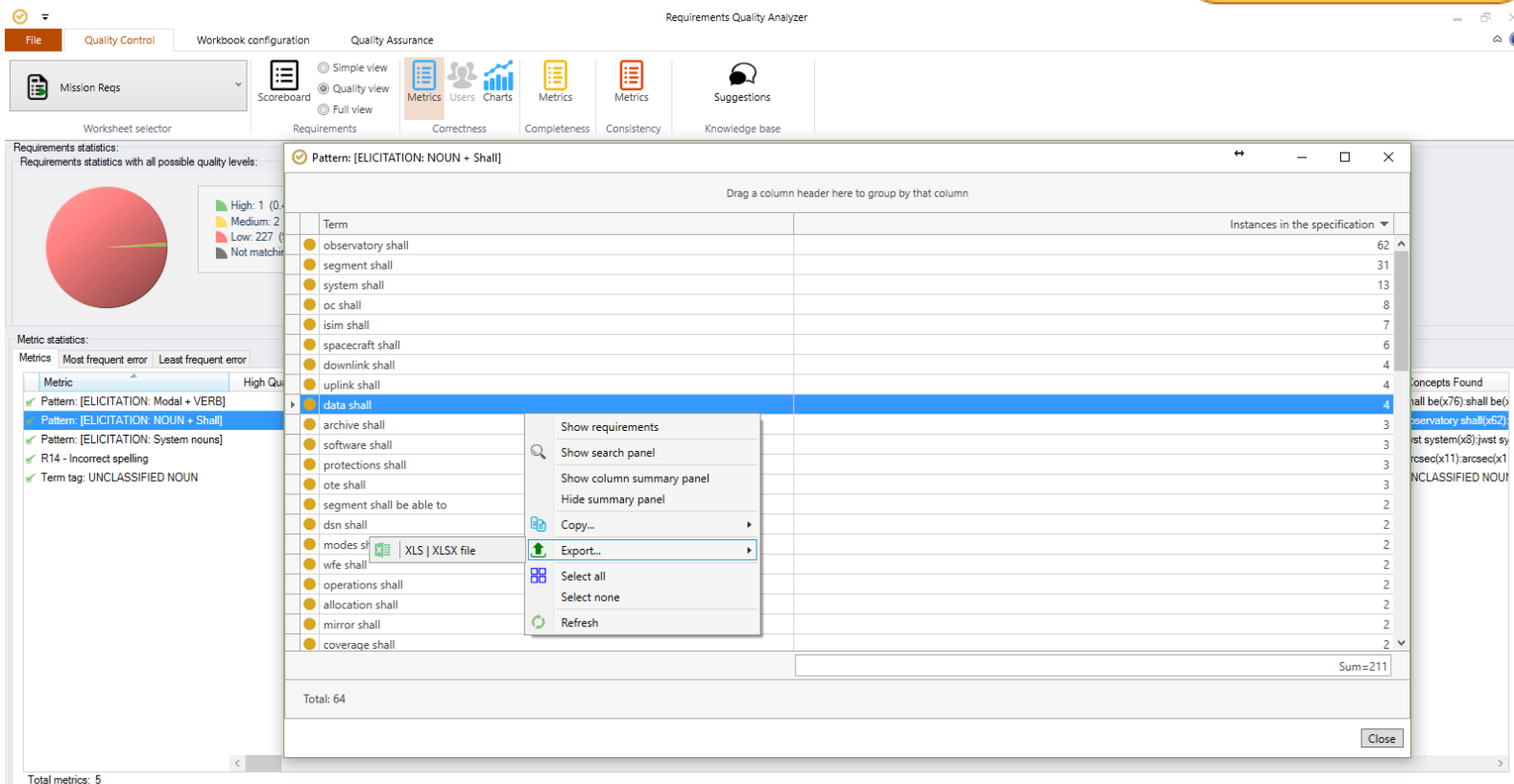
## Knowledge collection: document analysis (archeology)

➤ KM+RQA can use patterns to classify some of these common terms:

➤ System names

➤ Actions

**Step 3**



re knowledge, reuse it.



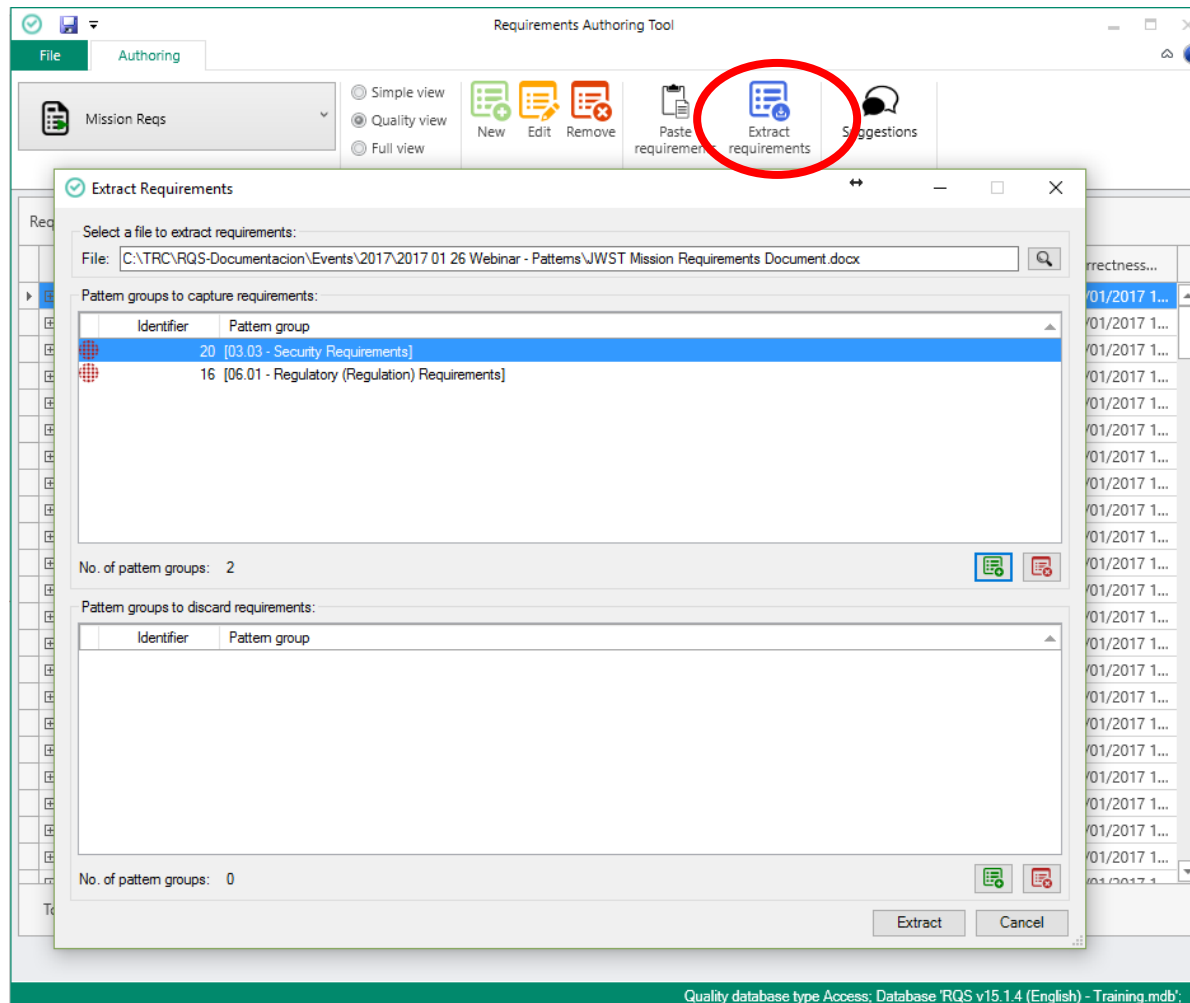
# Collecting *requirements*

## Collecting requirements

- Patterns can be used to identify requirements in textual unstructured sources
- The more unconstrained the patterns are, the more chances to catch new requirements
- One or more patterns are needed as input, together with a source file
- One or more patterns can be provided as exclusions/exceptions to the main list of patterns (normally, more constrained patterns than the main list)



## Collecting requirements





## Questions & Answers





Margarita Salas, 16 2<sup>nd</sup> Floor  
Innovation Center  
LEGATEC Technology Park  
28919 Leganés – Madrid  
SPAIN – EU



<http://www.reusecompany.com>



[@ReuseCompany](https://twitter.com/ReuseCompany)



Tel: (+34) 912 17 25 96  
Fax: (+34) 916 80 98 26



[contact@reusecompany.com](mailto:contact@reusecompany.com)