the REUSE company

Advanced requirements verification using parameterized metrics in RQS (correctness)

Webinar
Wednesday, March 22, 2017

Host: José Fuentes

Content

- Introduction to TRC and RQS
- Correctness analysis, the first dimension of the requirements quality
- Parameterized correctness metrics
- Custom-code correctness metrics
- Parameterized correctness metrics in RAT
- Live demo
- Q&A

Introduction

Introduction

José M. Fuentes

jose.fuentes@reusecompany.com

+34 912 17 25 96

@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(he) **R**(euse) **Q**(ompany)(y)

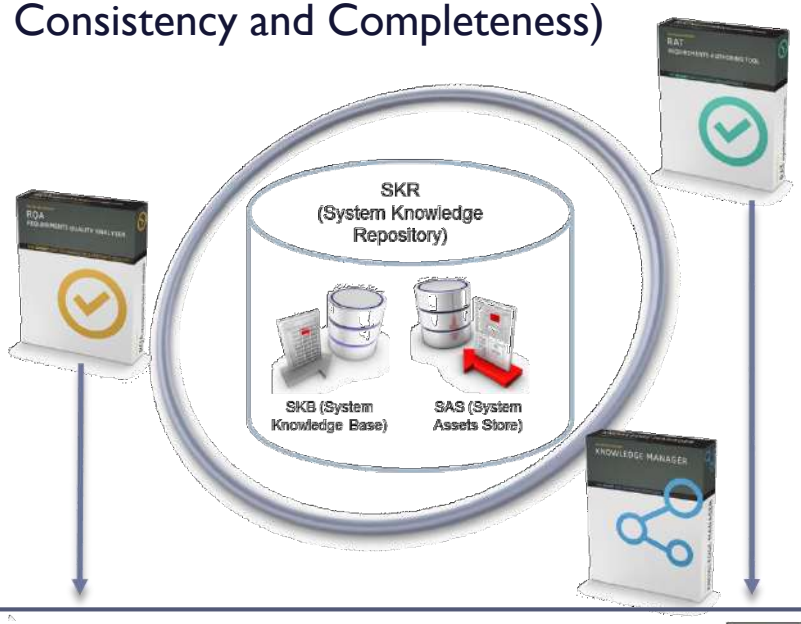
Trace + Retrieval + Quality

Towards systematic Reuse

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

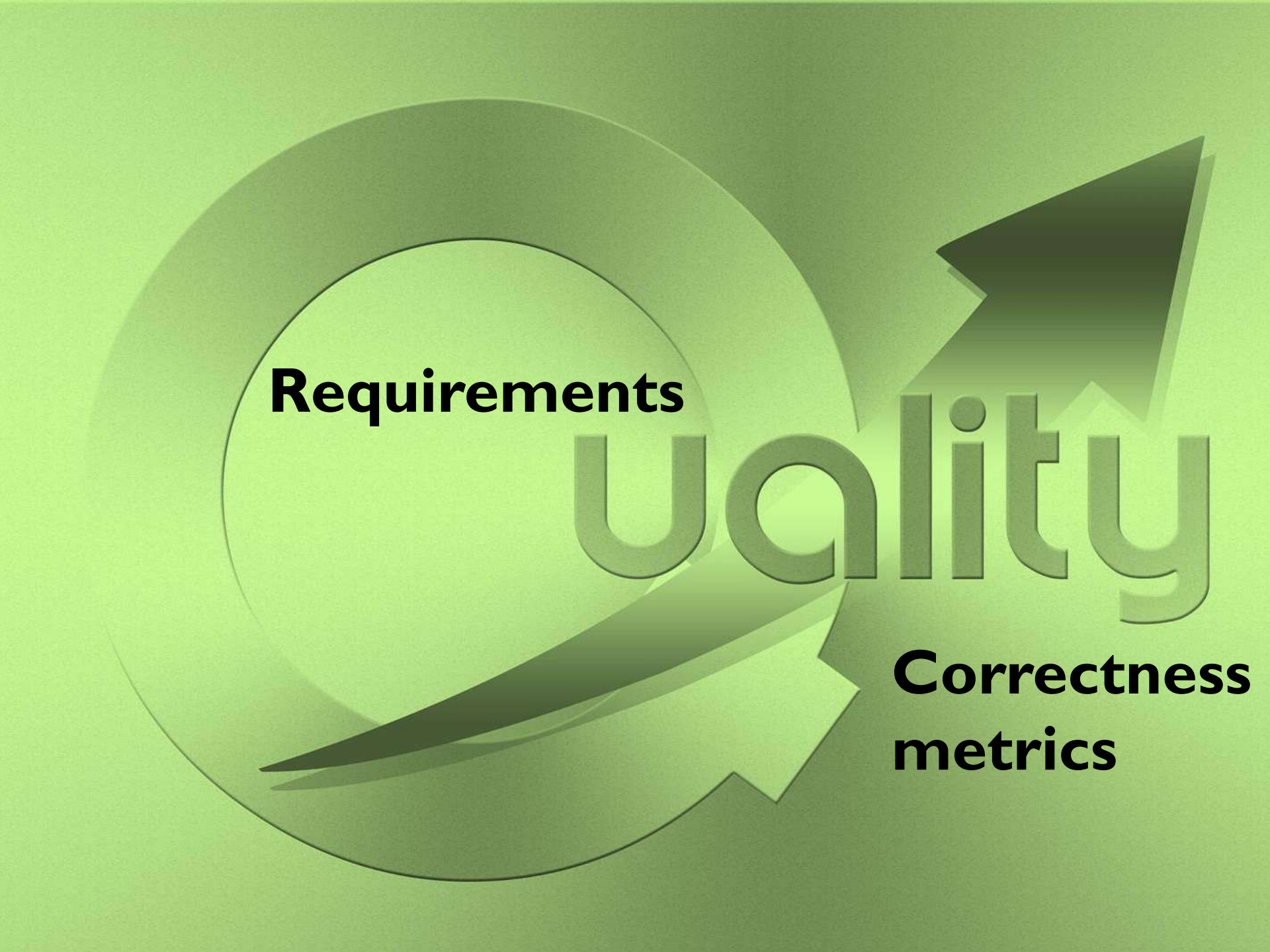
RQS – Requirements Quality Suite

- 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)



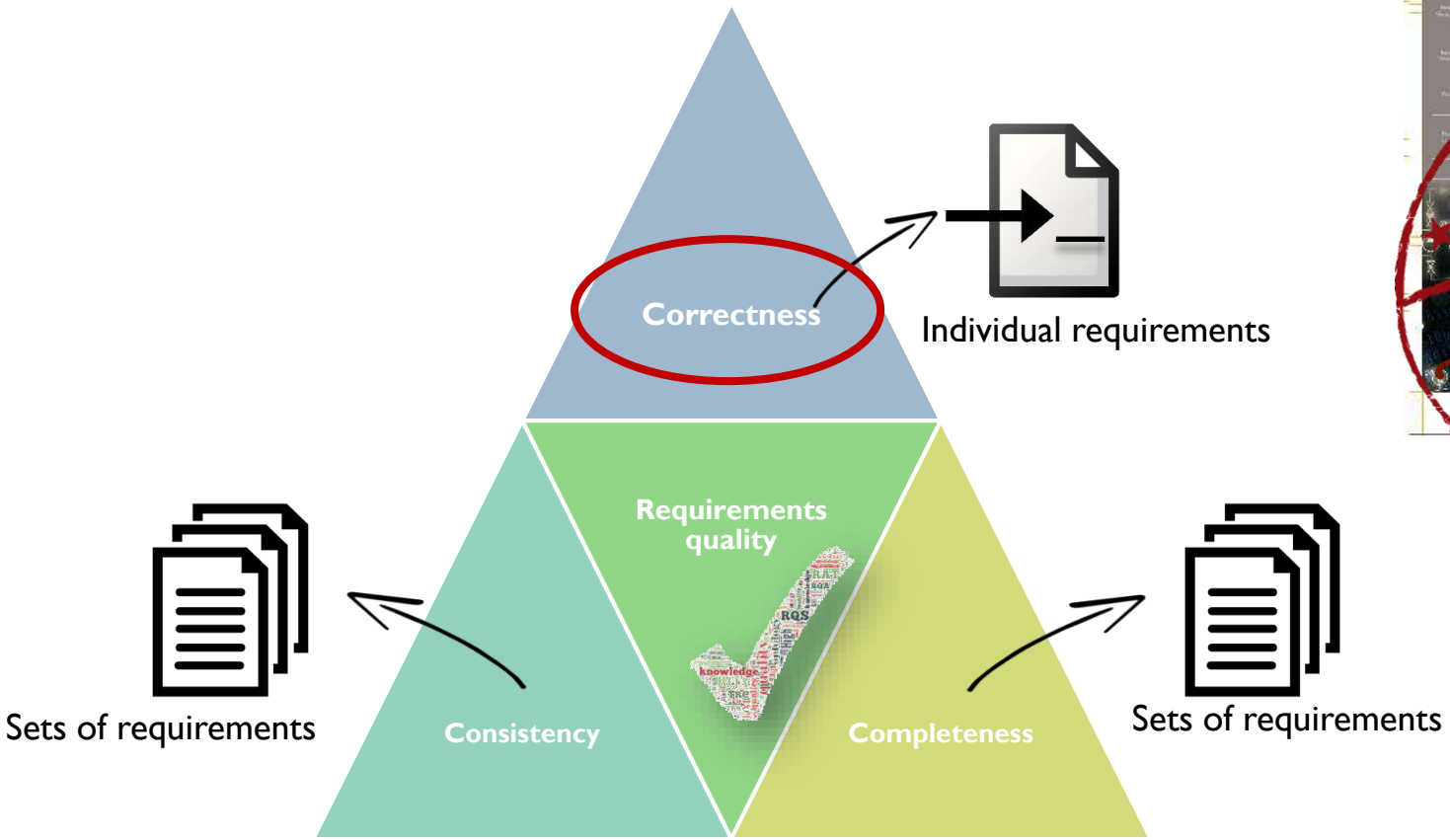
- **Requirements Quality Analyzer (RQA):** to setup, check and manage the quality of a requirements specification
- **Requirement Authoring Tool (RAT):** to assist authors while they are creating or editing requirements.
- **Knowledge Manager (KM):** to manage knowledge around a requirements specification: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

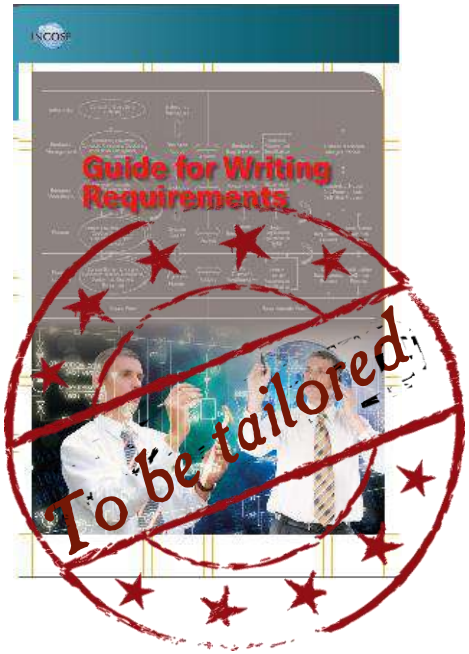
IBM PTC Integrity REQTIFY visure requirements OSLC Excel 2010 XML


Requirements
uality
Correctness
metrics

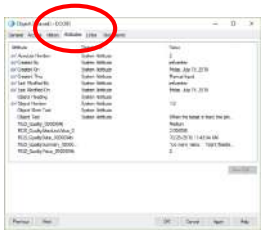
Requirements quality metrics: CCC Approach

- CCC – Correctness, Consistency and Completeness



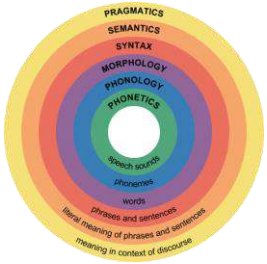


Examples of requirements metrics: Correctness



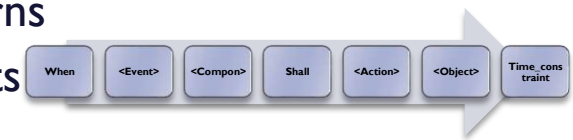
- Metrics based on information coming from the RMS:
  - Attributes, links, versions…
- Metrics based on lists of terms:
  - Forbidden: ambiguous…
  - Restricted: negations, pronouns…
  - Mandatory: 'shall'

- Metrics based on linguistic algorithms:
  - Text length, misspelling….
  - 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



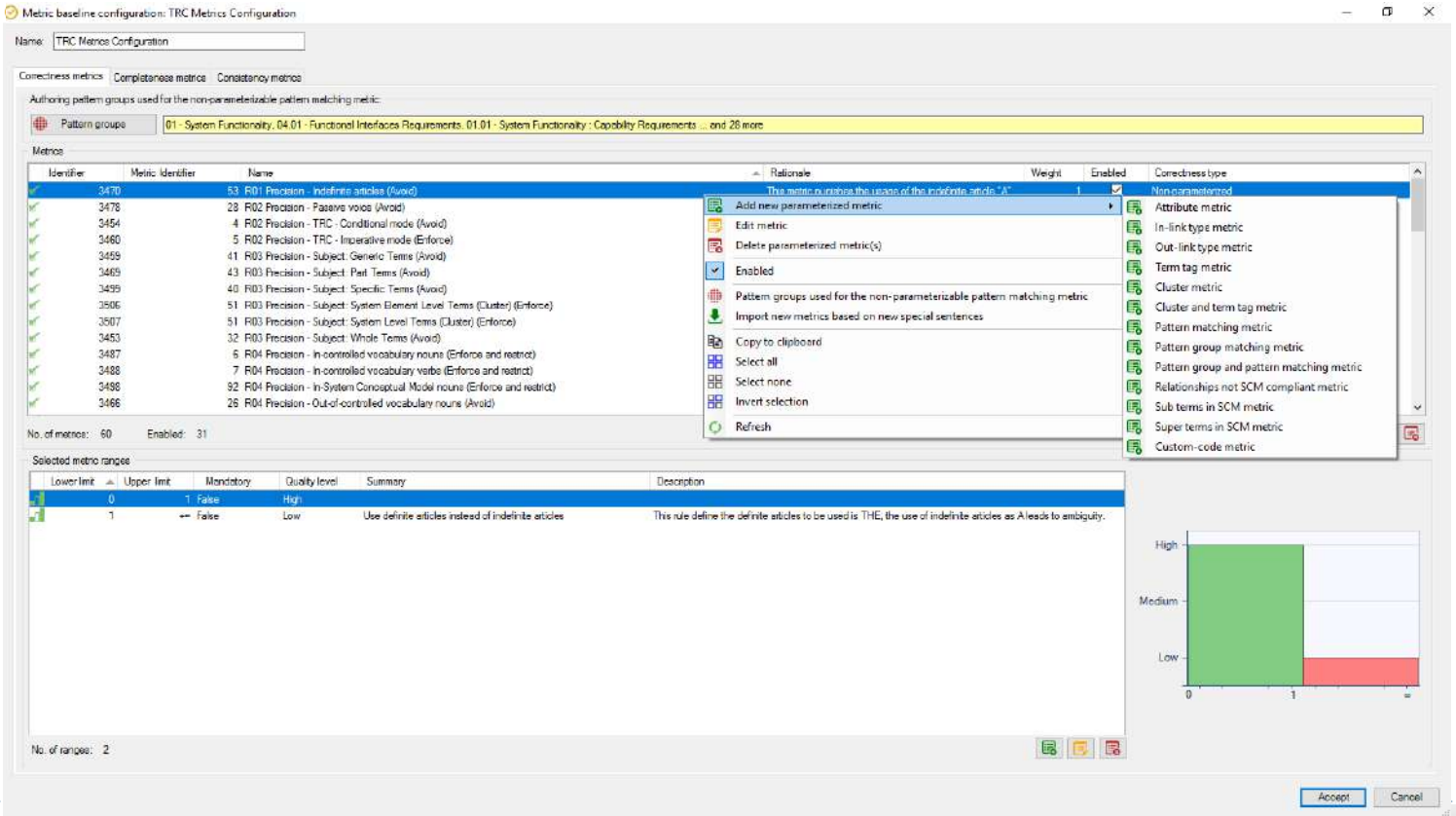
Correctness metrics

- RQS v15 includes, out-of-the-box +60 metrics
- Two new mechanisms to add more Correctness metrics:
  - **Parameterized metrics**: the behavior can be influenced by a parameter that can be different from one analysis to the other
  - **Custom-coded metrics**: you can create your own library of metrics
- One type of parameterized metric already exists from previous versions:
  - Metrics based on customizable lists of concepts
- What you get with this mechanism:
  - More powerful metrics
  - More flexibility
  - More accurate translation between your guidelines (paper-based) and RQS
  - Different parameterized metrics can be defined at the same time, even of the same type

Parameterized correctness metrics

Parameterized correctness metrics

- RQS includes a set of *built-in* metrics: out-of-the-box
- Plus the possibility of instantiating a number of parameterized metrics:

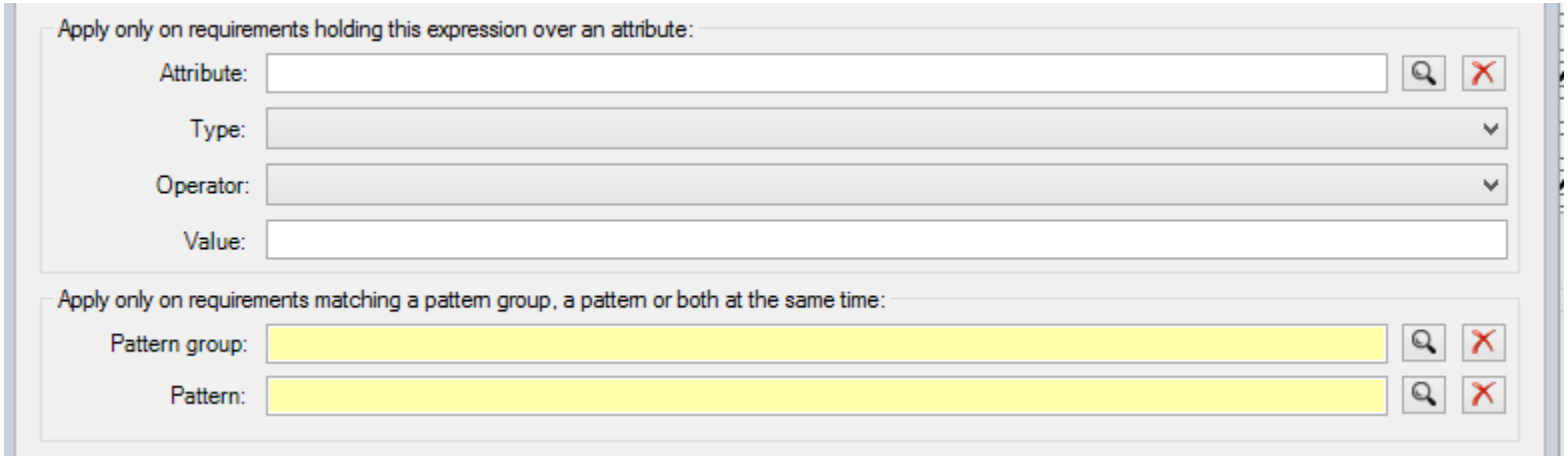


Parameterized correctness metrics

- As the rest of the metrics, the parameterized metrics:
  - Are seamlessly integrated with other metrics
  - Return a numerical value
  - Use quality functions to transform this value into a high-medium-low scale
  - Use weight to compute the overall quality level of the requirement

Parameterized correctness metrics

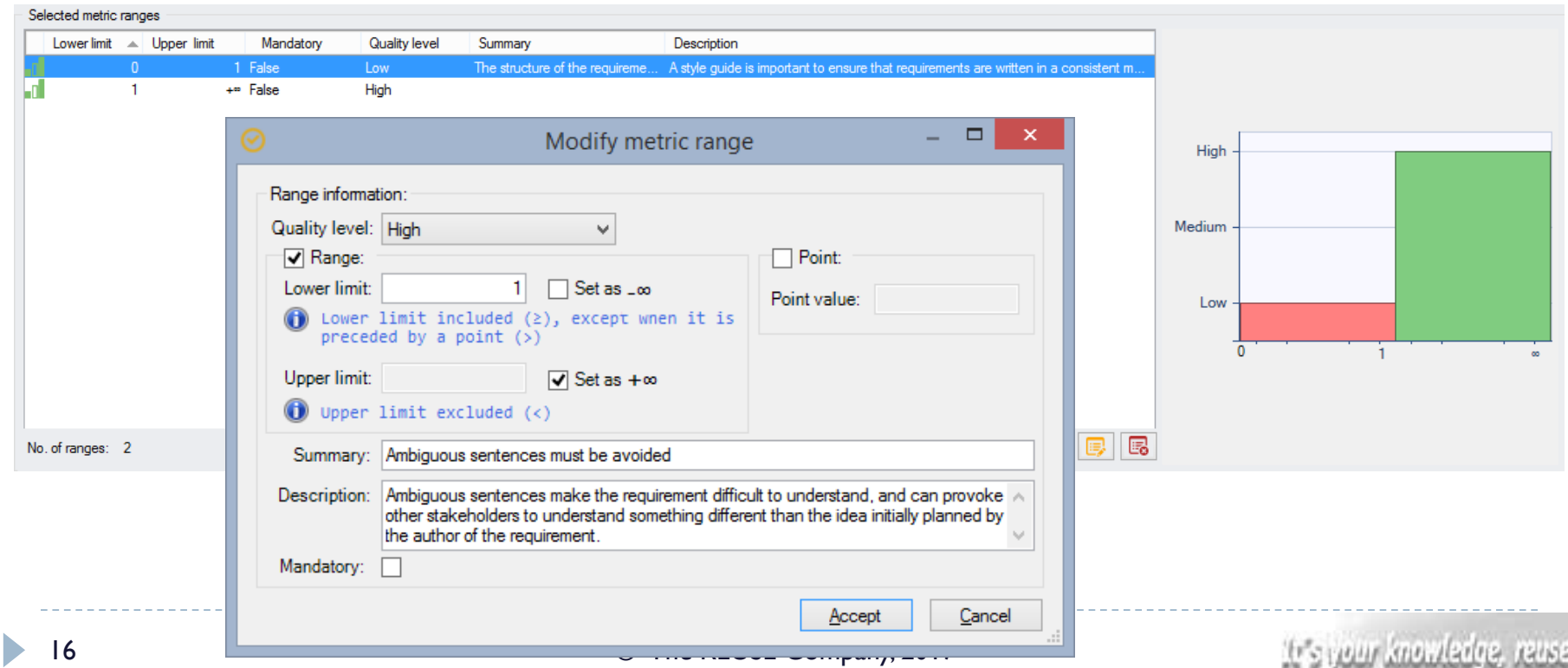
- Unlike the other OOTB metrics:
  - Can include an additional condition to filter out which requirements will not be assessed with a specific metric. Filtering is based on *attributes* or *patterns*



  - Some might not be available at metric baseline level, but rather at module specific level: e.g. attributes and links
  - Some are not available for some *not real RMS* (e.g. Excel, XML)

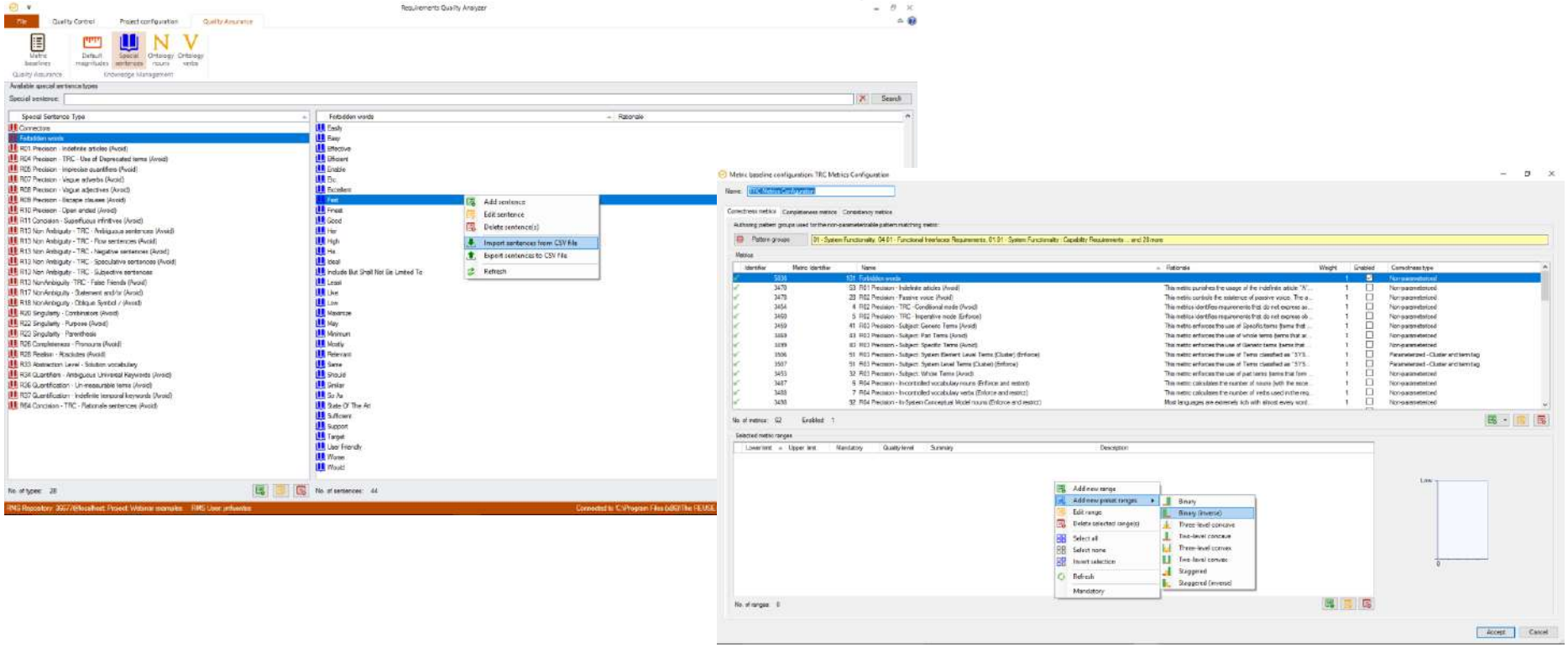
Parameterized correctness metrics. Returned value

- The quantitative result is computed thanks to a quality function, as in any other OOTB metric. E.g. for *Attribute expression matching*:
  - [0, 1) : the expression was not matched → Low quality
  - [1,∞) : the expression was matched → High quality



Special sentences

- Some lists are included as OOTB metrics
- These OOTB lists can be edited as needed
- New lists can also be created and managed, related to new metrics



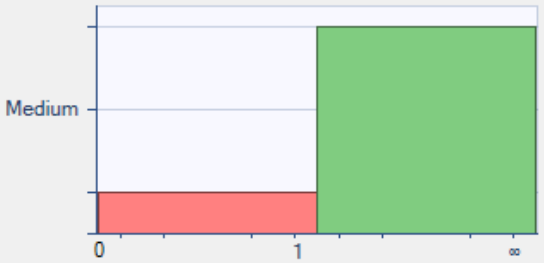
Parameterized Attribute metric

- Evaluation of the value of an attribute, looking for matching with an expression
  - Numbers: >, >=, <, <=, =
  - Strings: =, !=, *regular expression (link to an example based on reg expressions)*
  - Boolean: =, !=
- Possible return values:
  - Value = 1: the expression was matched → high quality
  - Value = 0: the expression was not matched → low quality
  - Value = 0: when the attribute doesn't exist

Selected metric ranges

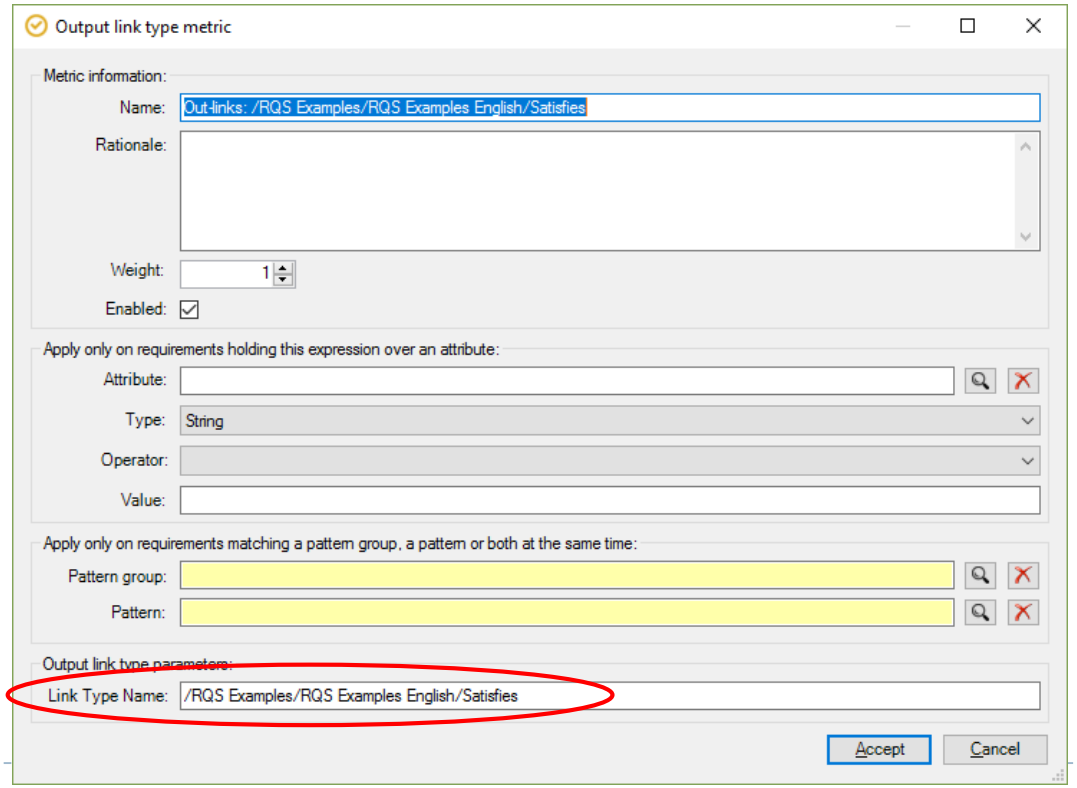
Lower limit	Upper limit	Mandatory	Quality level	Summary	Description
0	1	False	Low	Rationale attri...	Rationale attribute cannot be empty
1	+∞	False	High		

No. of ranges: 2



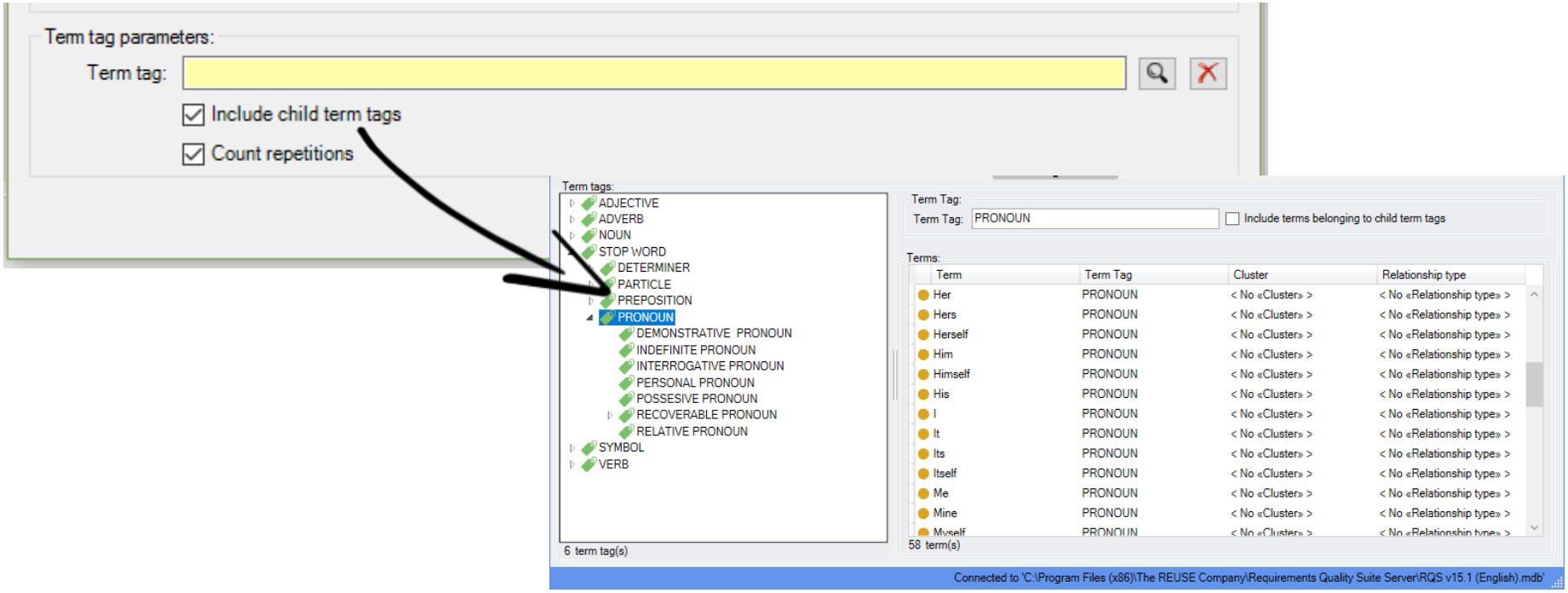
Parameterized in/out-link type metric

- Returns the number of links (*in* or *out*)
- Unlike the OOTB metric, the parameter represents a specific link module
  - N.B.: for DOORS, provide the **full path** of the Link Module



Parameterized term tag metric

- This metric counts how many terms of a given syntactic tag appear in a specific requirement
- Example: let's detect the number of pronouns to forgive the use of pronouns in a requirement



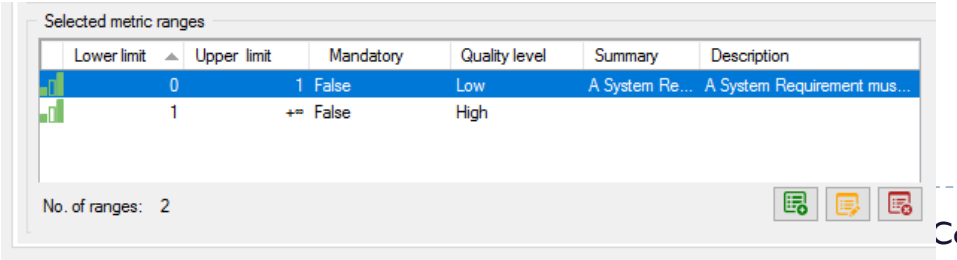
Parameterized term tag metric

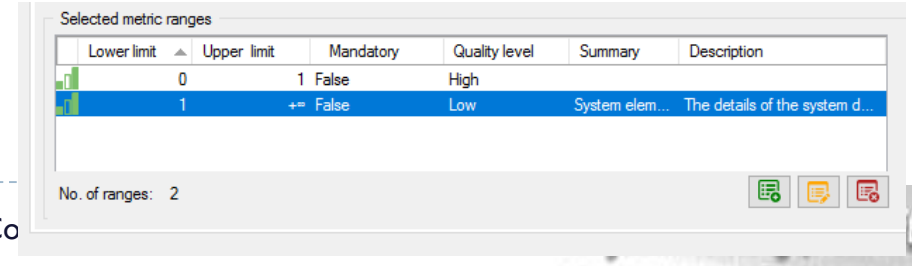
- Unlike the *special list based* metrics, this one uses semantic algorithms to disambiguate
- Example:
  - The IT Department shall generate a monthly report, it has to be sent to them

Parameterized semantic cluster metric

- Measures the number of terms, belonging to a specific semantic cluster (the parameter), that appears in a requirement
- Example: let's force every system requirement to involve –at least- the name of one system

- Depending on how the quality function is created, RQA can detect when a cluster element is expected but not found, or when it's not expected but found. In this example: OK for System, KO for System Elements



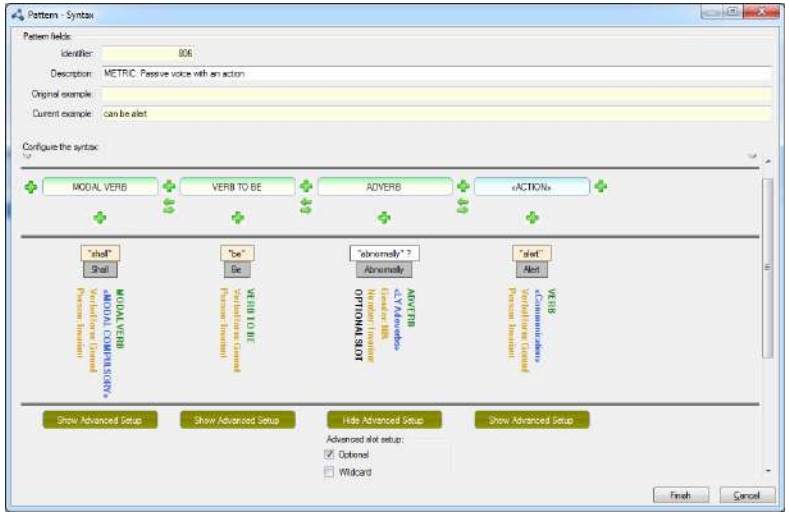


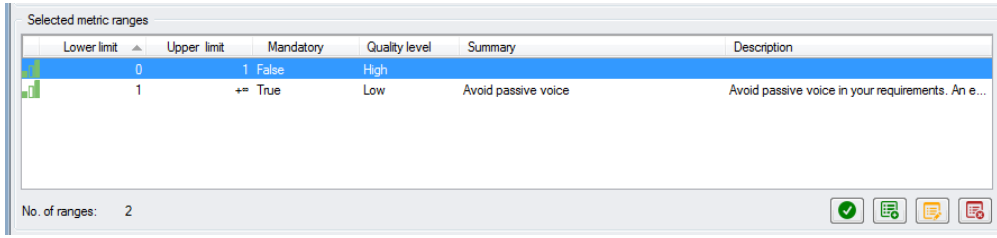
Parameterized pattern matching metric

- Receives a specific pattern as the parameter (watch this Webinar for more info about patterns) and counts how many times (if any) the pattern appears in the requirement

  N.B.: Only the selected pattern is checked for a possible match, regardless the weight of the rest of the patterns

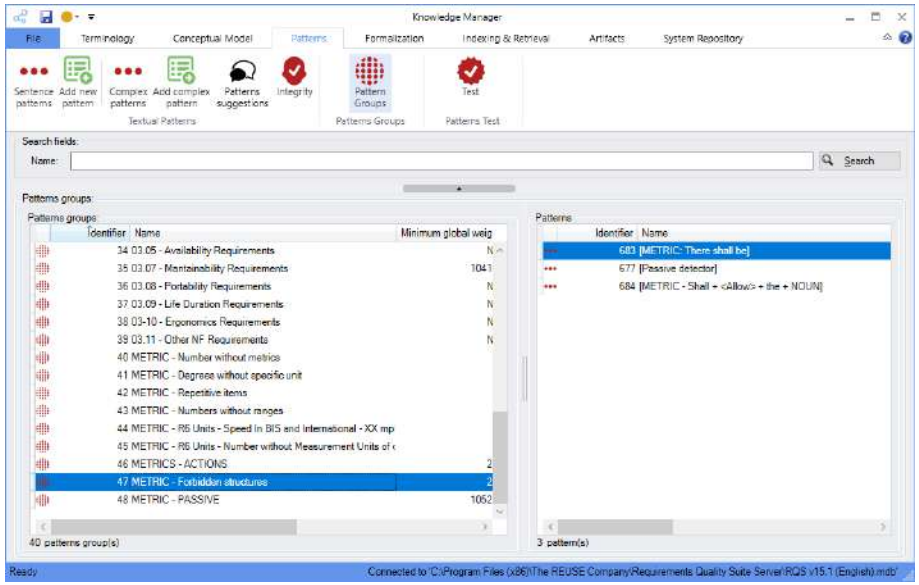
- Example: let's look for passive voice using patterns in order to avoid passive voice in a requirement





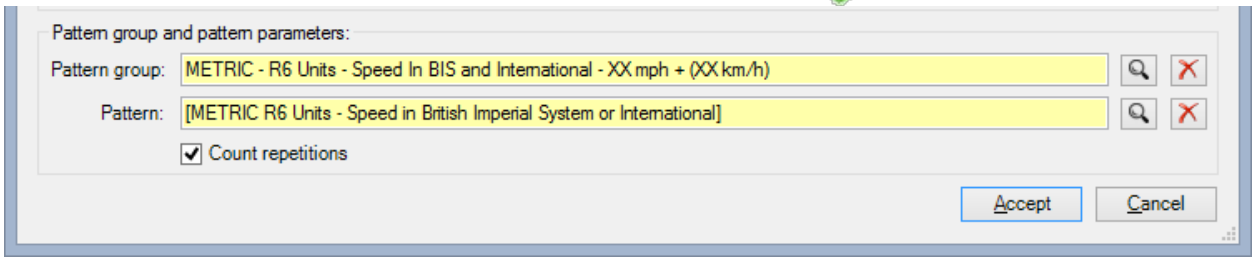
Parameterized pattern group matching metric

- Computes the number of times any of the patterns belonging to the selected pattern group (the parameter) is matched in a requirement
  - The weight of other patterns not belonging to the group are not interfering in this metric
- Example: let's try to identify a set of wrongly written requirements so that RQA/RAT can easily identify them. In this example, the following types of requirements will be identified:
  - "There shall be ….": every requirement must clearly state the subject of the action
  - "… shall allow + NOUN" for example, "The system shall allow the activation…". The action must be a verb instead of a noun
  - Passive structures
  - …



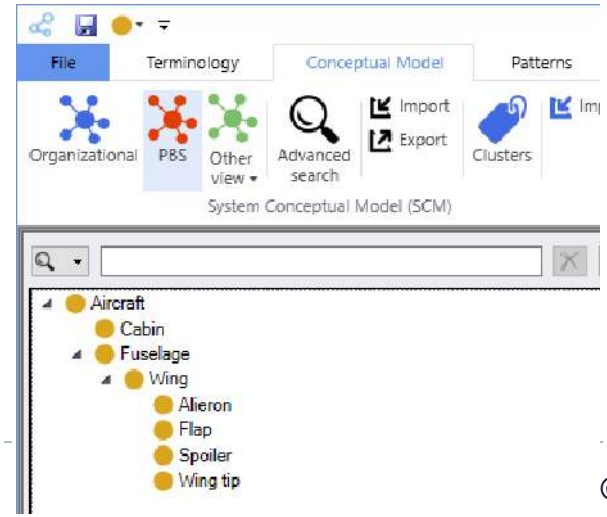
Parameterized pattern group and pattern matching metric

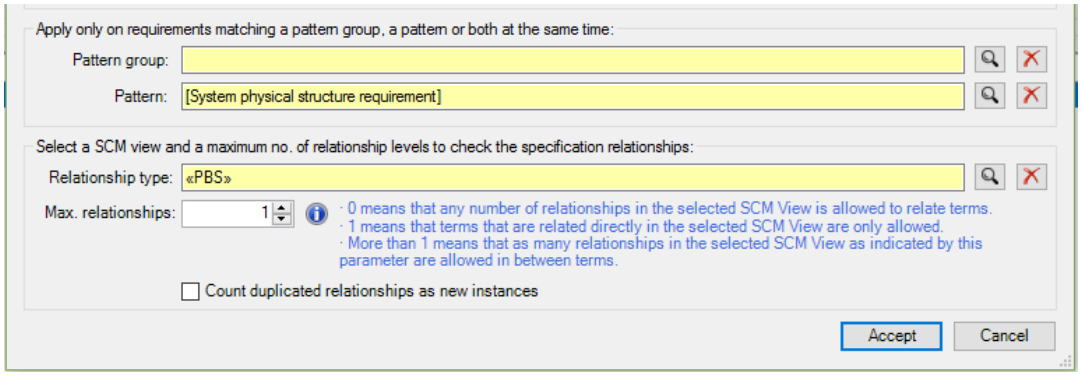
- This metrics takes a pattern group as a parameter (par#1) and a specific pattern into this group as the second parameter (par#2)
  - The weight of other patterns not belonging to this group are **not** interfering
  - But all the patterns that do belong to the group will "compete" with the rest by using their weights
- Example:
  - Let`s detect if every time a speed is stated in *mph*, the corresponding speed in *km/h* is also stated
  - E.g. 1: "When the speed of the car is below 5 mph, the driver shall…"
  - E.g. 2: "When the speed of the car is below 5 mph (8 km/h), the driver shall…"



Parameterized RSHP not in SCM

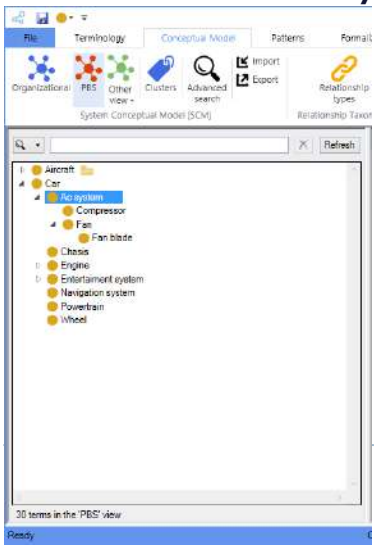
- Takes a specific type of relationship (a view in the SCM) as parameter
  - In the requirement, it takes all the relationships, with the selected type, that are generated (formalized based on requirements patterns)
  - For each couple of related terms in the requirement, it checks whether those related terms are **not** related in the SCM
- Example:
  - Let's check the PBS to avoid *bad* requirements such as: "The cabin shall have wings", or even "The Aircraft shall have flaps",

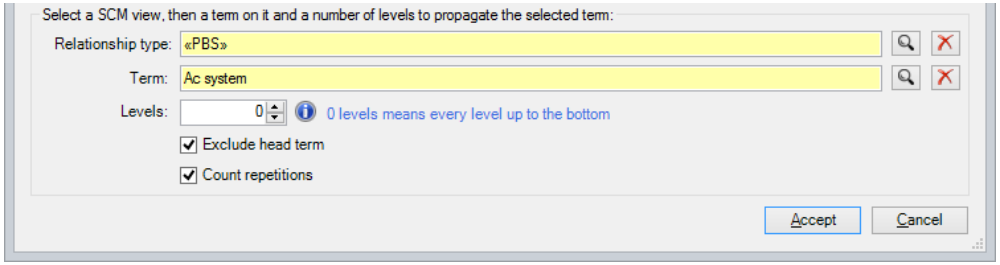




Parameterized Super/Sub-terms in SCM

- Given a specific view of the SCM (e.g. a PBS structure)
  - Counts the number of terms under a specific point in the SCM
  - Note that the opposite (super-terms) can also be checked
- Example:
  - Let's assume a requirements specification about a specific item in the PBS
  - The metric detects (and forbids) any more detailed sub-element in the PBS structure to avoid early architectural details





Custom code Metrics

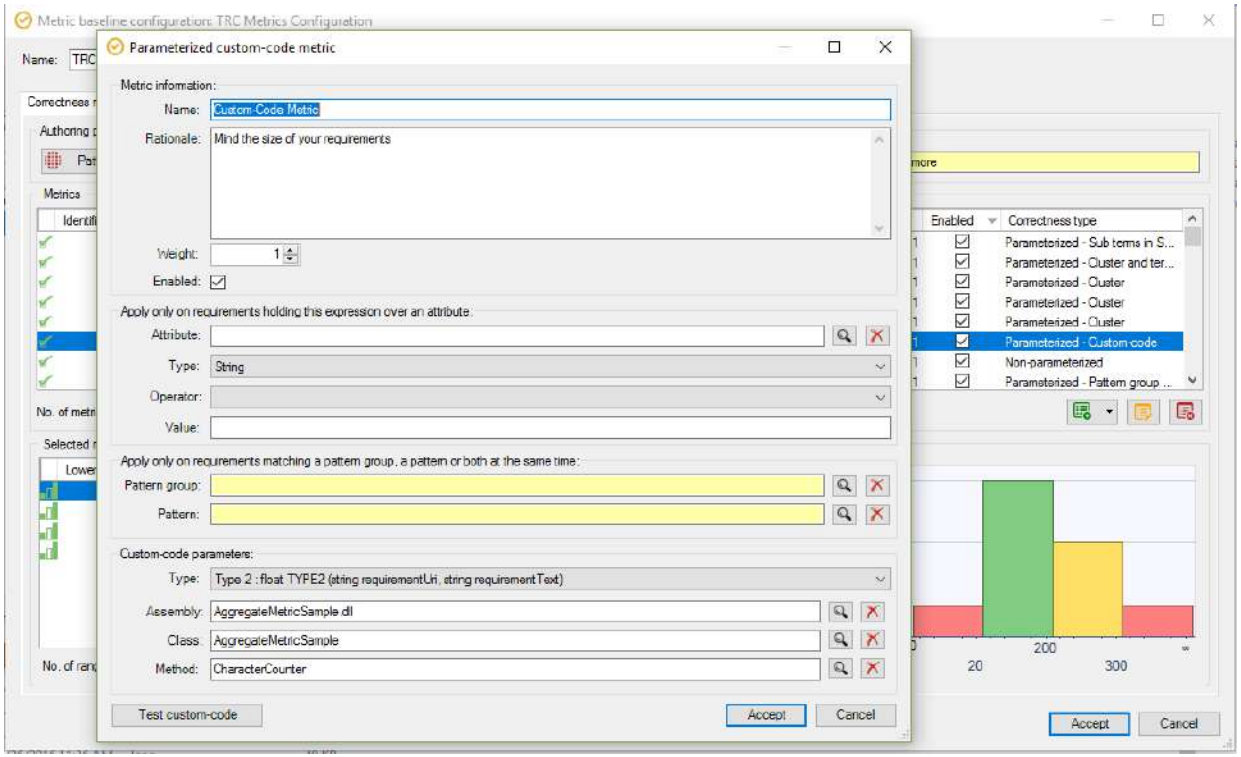
Parameterized Custom – Code metrics

- This extensibility mechanism allows you to code your own metrics
- Use .NET Framework 4.0 to create your library
- 5 different interfaces. The most common:
  - TYPE 1. Custom function only taking into account the requirement URI
    - `float custom_Function_Type1 (string requirementUri)`
  - TYPE 2. Custom function taking into account also the requirement text
    - `float custom_Function_Type2 ( string requirementUri, string requirementText)`
  - TYPE 3. Custom function taking into account the requirement text and the out-of-the box metric evaluations
    - `float custom_Function_Type3(string requirementUri, string requirementText, RequirementEvaluation[] metricIdsAndEvaluations)`

Parameterized Custom – Code metrics

- Example of type 2:

```
0 references
public static double CharacterCounter(string requirementUri, string requirementText) {
    int counter;
    counter = requirementText.Length;
    return counter;
} // CharacterCounter
```



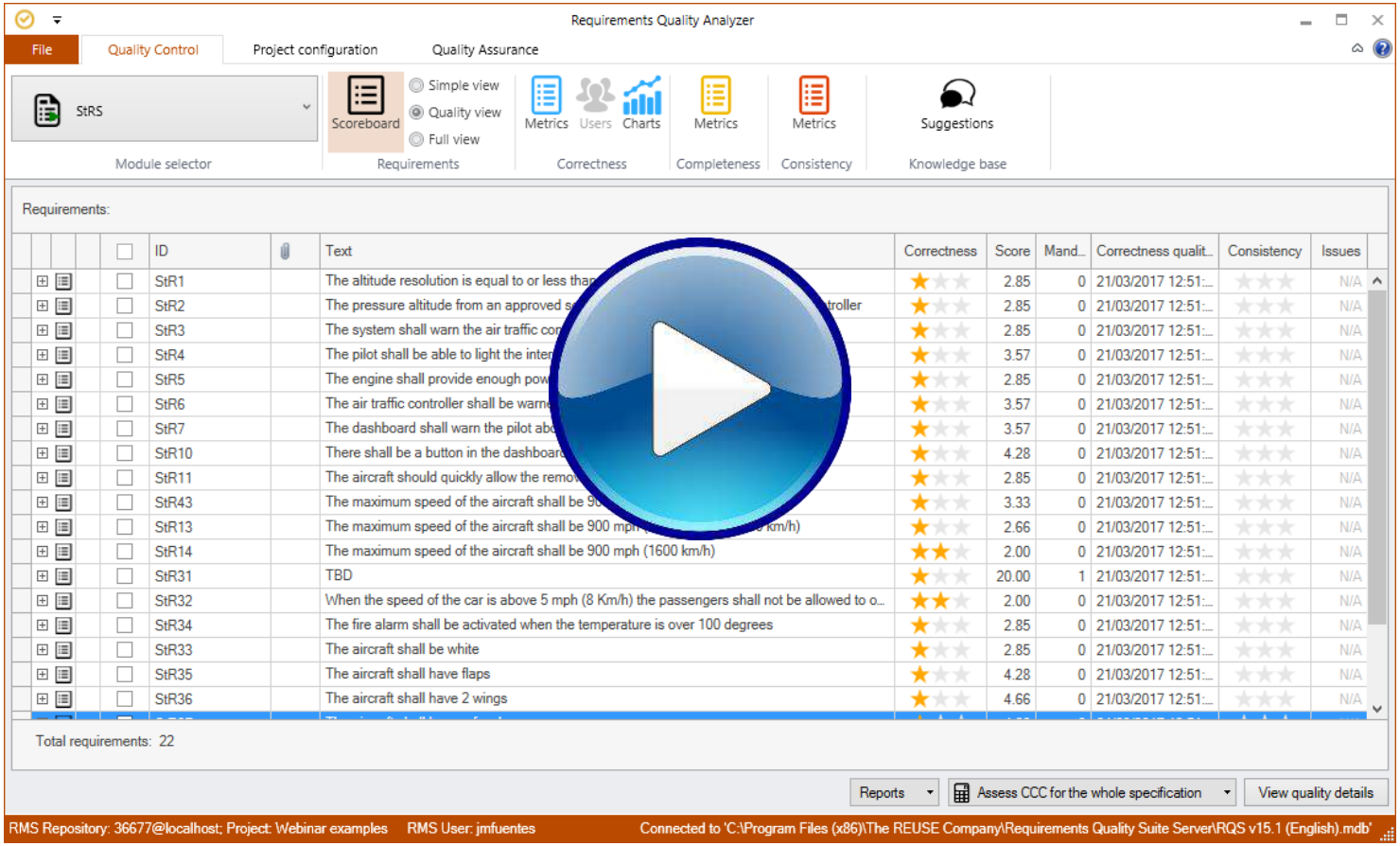
Parameterized Custom – Code metrics

- Example of type 3:

```
0 references
public static double TraceabilityChecker(string requirementUri, string requirementText, IRequirementMetric[] requirementEvaluations) {
    double result;
    result = 0f;
    if (requirementEvaluations != null) {
        IRequirementMetric OutlinksDerives, InlinksVerifies;
        OutlinksDerives = (from IRequirementMetric requirementEvaluation in requirementEvaluations
                           where requirementEvaluation.MetricId == 44 && requirementEvaluation.MetricPerTypeId == 2906
                           select requirementEvaluation).FirstOrDefault();
        InlinksVerifies = (from IRequirementMetric requirementEvaluation in requirementEvaluations
                               where requirementEvaluation.MetricId == 44 && requirementEvaluation.MetricPerTypeId == 2876
                               select requirementEvaluation).FirstOrDefault();

        if (InlinksVerifies.AbsoluteScoring == 1 && OutlinksDerives.AbsoluteScoring > 0) {
            result = 1; // High quality
        } else {
            result = 0; // Low quality
        }
    }
    return result;
} // TraceabilityChecker
```

Live demo



Questions & Answers

the REUSE company

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

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

http://www.reusecompany.com

@ReuseCompany

contact@reusecompany.com