The Reuse COMPANY

The Requirements Authoring Tool

Luis Alonso | Borja López | José Fuentes

www.reusecompany.com

Requirements Authoring Tool as part of the Requirements Quality Suite

The Requirements Quality Suite

- A set of integrated tools and a process aiming to:
  - Define
  - Measure
  - Manage
  - Improve
- the quality of the outputs of the Requirements' Processes

- RQS: integrated with IBM DOORS, VISURE Requirements, Reqtify and Excel
- RAT: integrated with IBM DOORS

RQS Version 4 : Features

- Definition of the Requirements Quality Process
  - PDCA cycle
- Roles Separation and tools separation
  - The Requirements Quality Suite (RQS)
- Requirements Authoring Assistance: RAT tool
- Focus on CCC metrics
  - Consistency
  - Completeness
  - Correctness

RQS Features

Metrics

- Metrics based model for measuring and improving quality
- Supports text based and NON text based measures
- Supports metrics for individual requirements and sets of requirements
- Customizable measures calculation engine
- CCC Support

Functional Operation

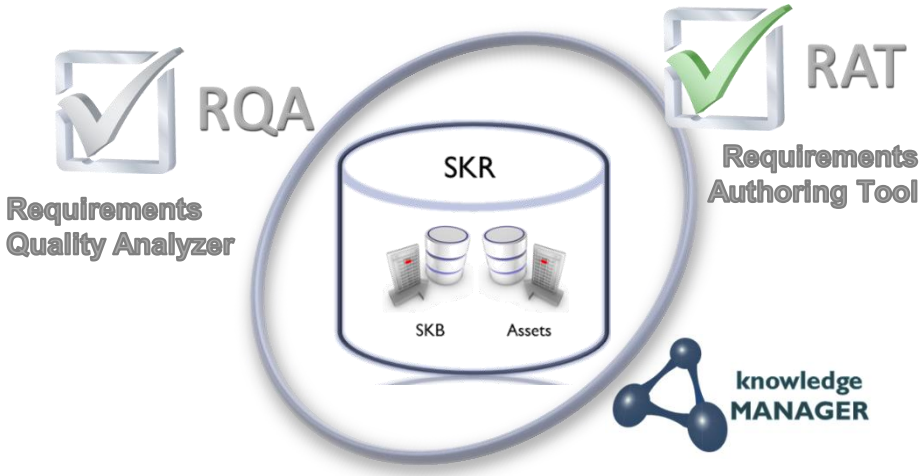
- Multi roles operability (Project Manager, QA Manager, Engineer)
- Calculations can be performed on-line (on demand) or planned
- Fully integrated with RMS

Other features

- Enhanced reporting and subscription system
- Individual requirements metrics based on Quality Functions

The Requirements Quality Suite (RQS)

- The Requirements Quality Suite (RQS) intends to tackle requirements quality management by offering a set of tools and processes.
- RQS models requirements quality 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.

knowledgeMANAGER:
to manage knowledge around a requirements specification: the ontology it is based on, the structure of the requirements to be used in the project, the communication between authors and domain architects.

The Requirements Quality Suite

- RQA – Requirements Quality Analyzer:
  - CCC: Correctness, Consistency and Completeness Analysis
  - Configure RQS with the quality policies and checklist of your organization
  - Check the correctness of your requirements specifications

- RAT – Requirements Authoring Tool:
  - Write your requirements easily by using an assistant
  - Correctness analysis on the fly
  - Consistency analysis on the fly

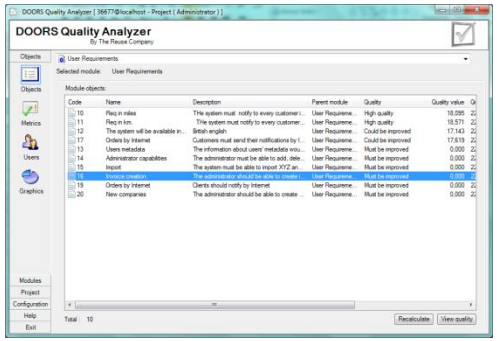
- knowledgeMANAGER:
  - Manage all the domain knowledge behind the quality analysis
  - Management of glossaries, taxonomies, thesauri and ontologies
  - Management of the boilerplates to be used by RAT

RQS V4.1

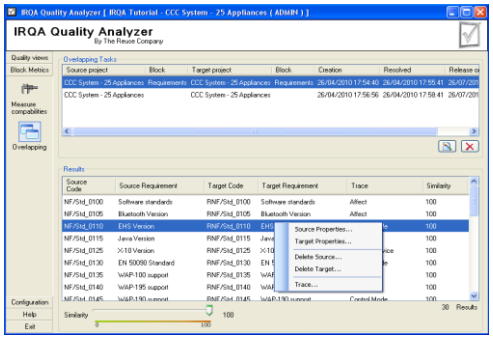
The Requirements Quality Analyzer is a software tool that aids quality assessment and improvement within requirements oriented software and systems projects.

Quality configuration, measurement, reporting… following the CCC approach.

- Measures single requirements quality (correctness)



- Measures requirements docs quality (completeness and consistency)



Requirements Authoring Tool - RAT

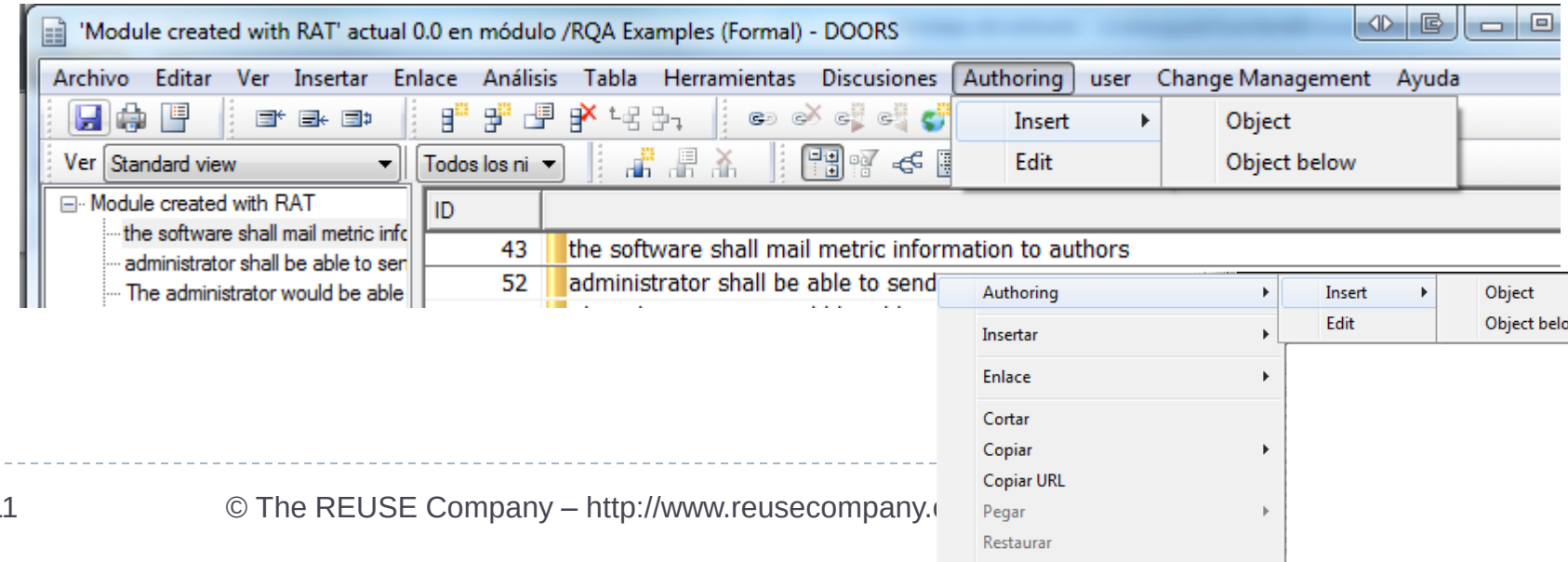
Requirements Authoring Tool – RAT V4.1

- **Main features:**
  - Assisting authors while they're writing requirements
  - Following a agreed upon set of boilerplates

- **Other (on the fly) features:**
  - Quality assessment (correctness based on individual metrics) on the fly
  - Consistency analysis on the fly
  - Missing links on the fly
  - Inconsistent units analysis on the fly

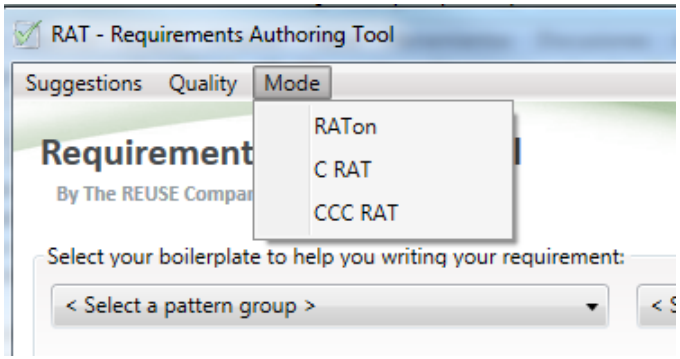
How to launch RAT

- Unlike RQA, RAT is fully integrated into your Requirements Management System (RMS)
- Open you RMS, and find the proper menu options to:
  - Create a new requirement based on RAT
  - Edit a requirement with RAT
- Both options will replace the standard way of writing/editing requirements offered by you RMS



Three different layouts/Modes

- Press <Alt> key to get the main menu, then click *Mode* to switch among modes

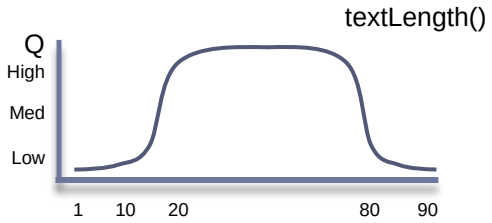


- RATon:
  - Focused just on correctness
  - Simplified layout
- C RAT: Includes RATon features plus:
  - Assistance in requirements writing
- CCC RAT: Includes C RAT features plus:
  - Full correctness information
  - Inconsistent units
  - Coupled/overlapped requirements

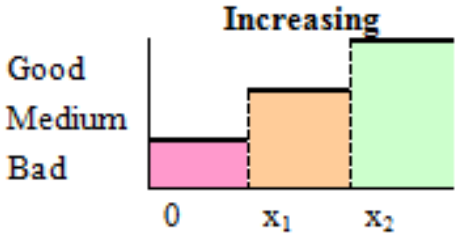
Requirements Correctness

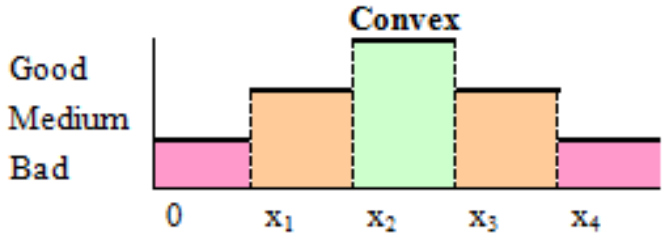
RQS's Quality Functions: Single requirement

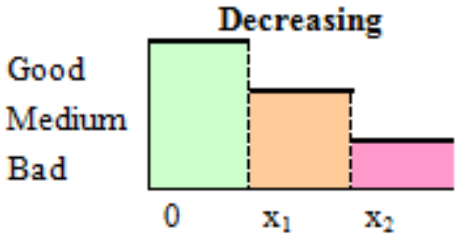
- Most of the Single Requirement Metrics are quantitative.
- The values of the metric are calculated counting items
  - Example: Metric SIZE=> Counts the number of words.
- The process is simplified by not allowing continuous quality functions

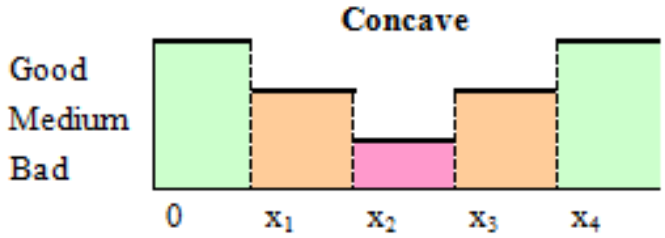


- The metrics use one of the following quality functions:









Individual requirements supported metrics : Correctness

- Size
- Readability
- Conditional vs. imperative sentences
- Active vs. passive voice
- Optional sentences
- Ambiguous sentences
- Subjective sentences
- Implicit sentences
- Abuse of connectors
- Negations
- Speculative sentences
- Use of false friends
- Use of deprecated concepts
- Design terms
- Flow terms
- Number of domain nouns and verbs
- Acronyms
- Hierarchical levels
- Volatility
- Number of dependences
- Standard Requirement

Individual requirements supported metrics 1/3

- **Size:** expressed in paragraphs, chars, nouns or verbs. Long requirements will be difficult to understand
- **Readability:** number of letters between punctuation marks and some other formulas than indicate whether the requirement will be easy to read. Ease to read requirements generates less problems all over the project
- **Conditional sentences vs. imperative sentences:** avoid *would* and use *Shall, should* and *will* in the right way
- **Active vs. passive voice:** avoid using passive voice to increase the readability of the requirement
- **Optional sentences:** maybe… Optional requirements must be stated by an attribute, never in the body of the requirement
- **Ambiguous sentences:** fast, user-friendly… What do the analyst, the coder and the customer understand by the same ambiguous sentence

Individual requirements supported metrics 2/3

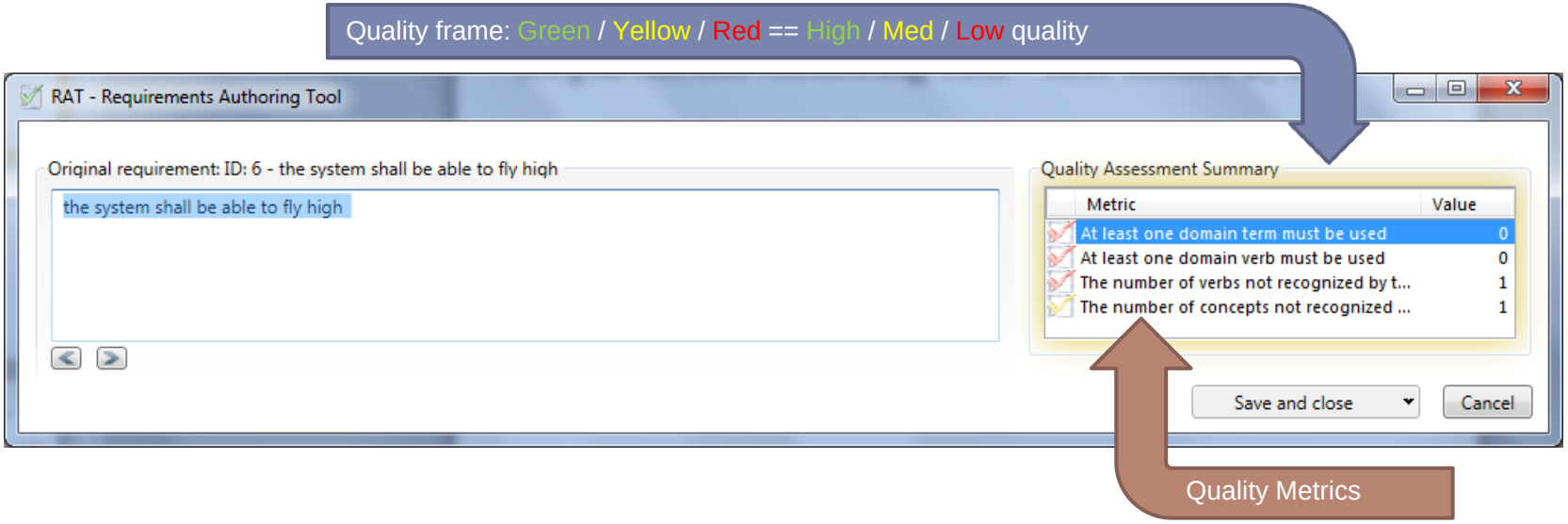
- **Subjective sentences:** in my opinion, I think that… Don't show your ideas, but what the system should do
- **Implicit sentences:** it must be provided by them… Too many pronouns make your requirements difficult to understand
- **Abuse of connectors:** and, or. Many times connectors reveal different needs enclosed within the same requirement, loosing the atomic characteristic
- **False friends:** customized according to "mother language" of your project
- **Negations:** no, never… Two or more negations in the same sentence make it difficult to understand
- **Speculative sentences:** usually, almost always… Make the requirement imprecise
- **Design terms:** loop, hash… Remember, avoid How, concentrate in What
- **Flow terms:** while, if, else… Remember avoid How, concentrate in What

Individual requirements supported metrics 3/3

- **Number of domain nouns and verbs:** domain terms and verbs should be involved into the requirement specification, nevertheless, too many different terms in the same requirement many times means multiple needs
- **Acronyms:** avoid those that don't belong to the domain representation
- **Deprecated concepts:** avoid those concepts whose use is no longer accepted
- **Hierarchical levels:** don't complicate your specification with too many indentation levels
- **Volatility:** if a requirement suffers many changes, you must be very careful with it
- **Number of dependences:** the same if your requirement is the source of too many dependences

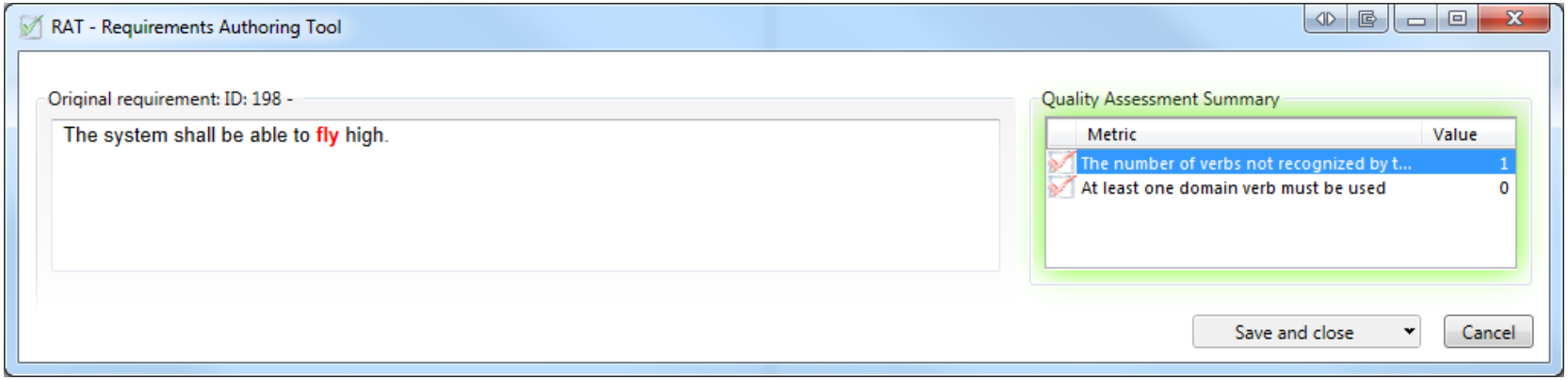
RAT – correctness

- Start your RMS
- Create/Edit a requirement with RAT
- Type/Edit the textual content of your requirement
- You'll have the quality information on the fly



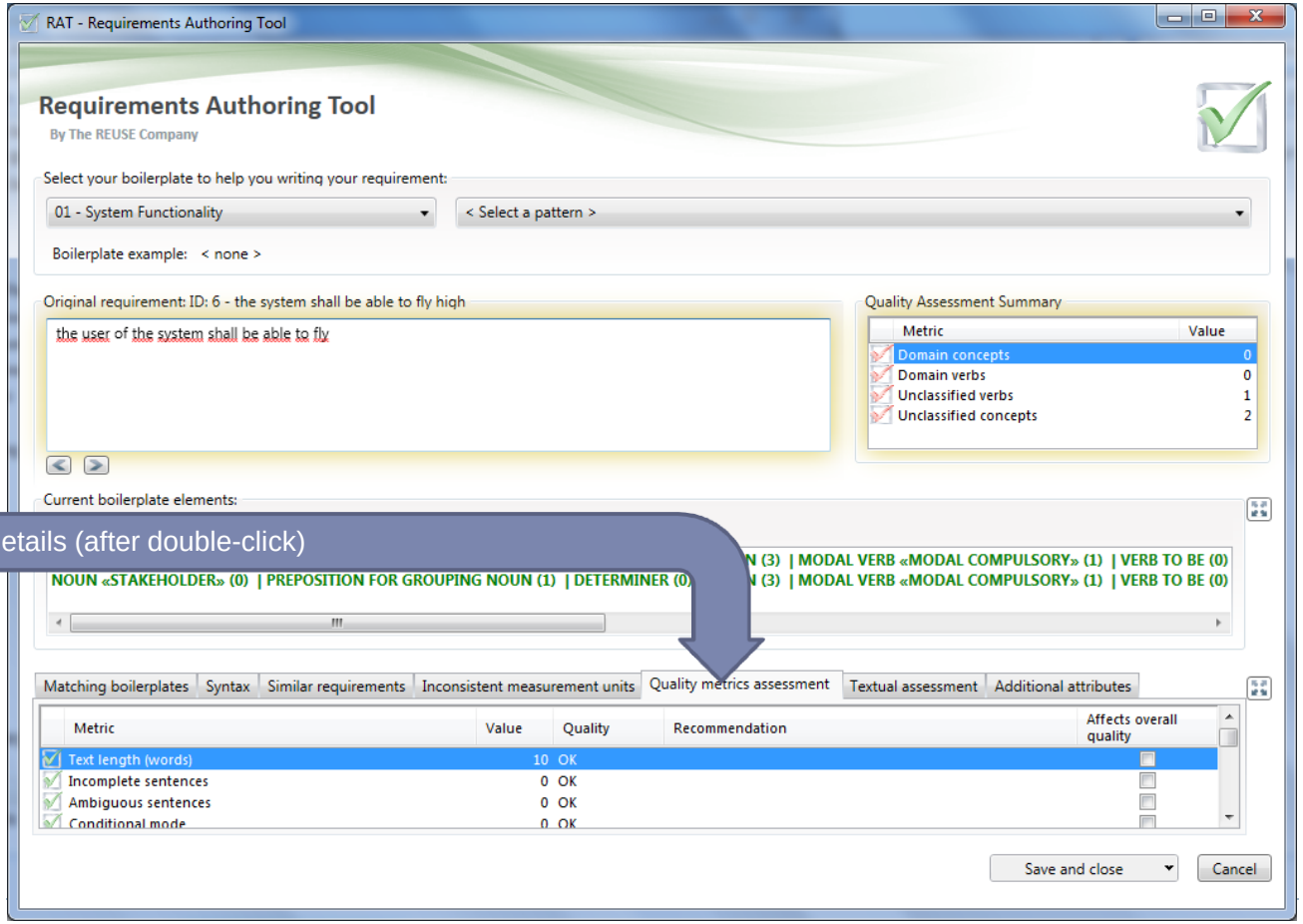
RAT – correctness

- Click on any of the metrics, in the right-hand side, to highlight the concepts in the text



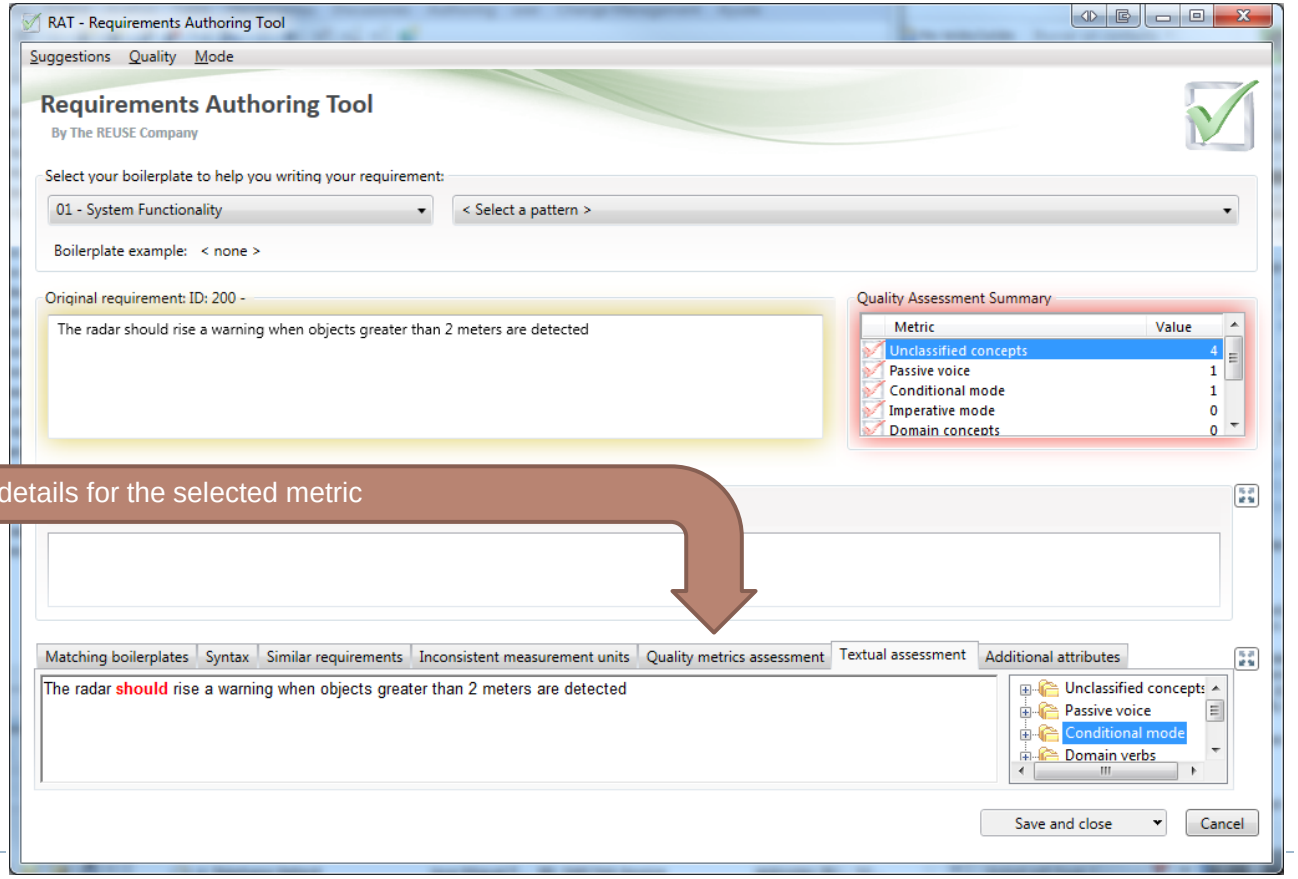
RAT – correctness

- Switch to *CCC* mode for more correctness details



RAT – correctness

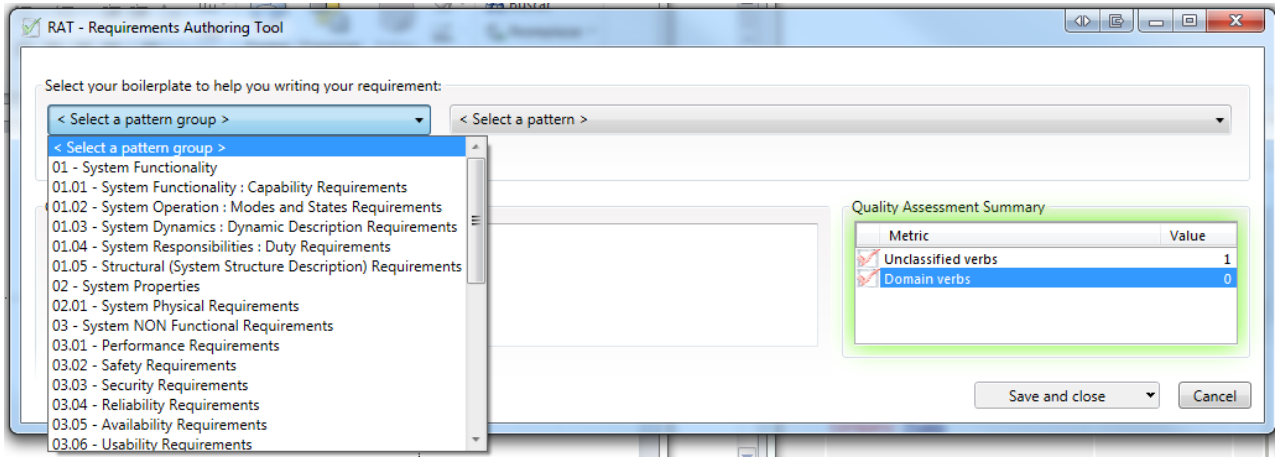
- Switch to *CCC* mode for more correctness details
- And even more textual details in the *Textual assessment* tab



Requirements writing assistance

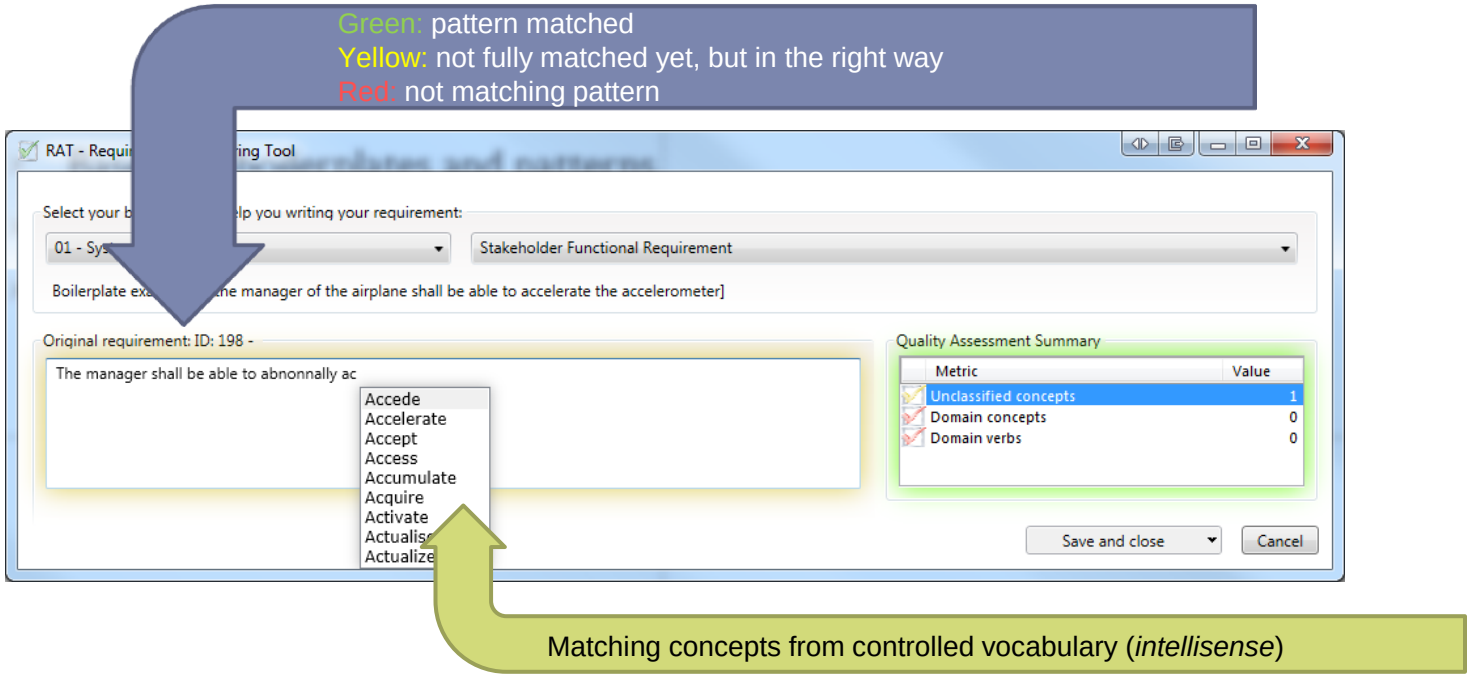
RAT – writing assistance

- Based on boilerplates and patterns
- Based on the information from a controlled vocabulary
- Switch to *C RAT* mode and select a pattern:
  - First select a group, then a specific pattern



RAT – writing assistance

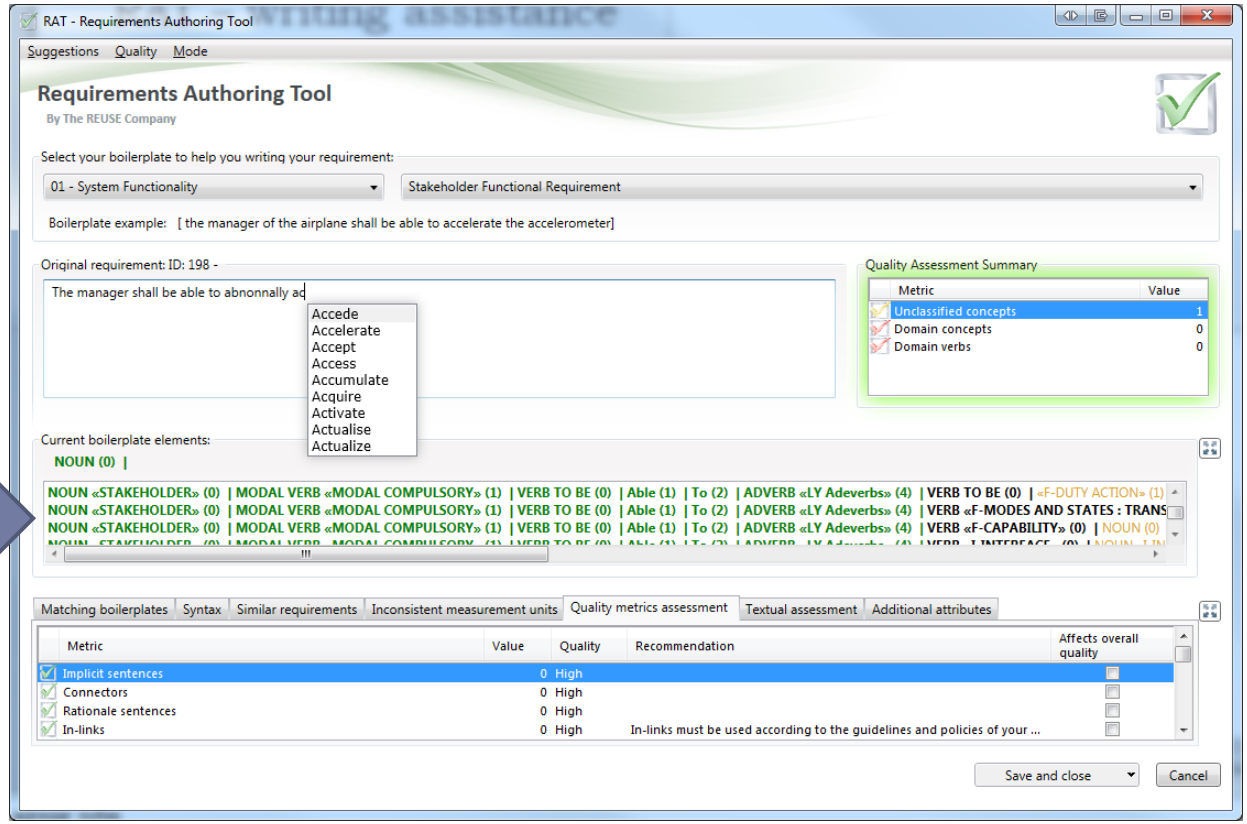
- Now RAT warns you in case you're not following the syntax of the selected pattern
- Controlled vocabulary is accessed to provide the right set of concepts in each step (*intellisense)*



RAT – writing assistance

- Switch to *CCC* mode to have even more information about the elements of the boilerplate

Matching boilerplates slots:
- Black: current token
- Green: previous tokens
- Yellow: next tokens



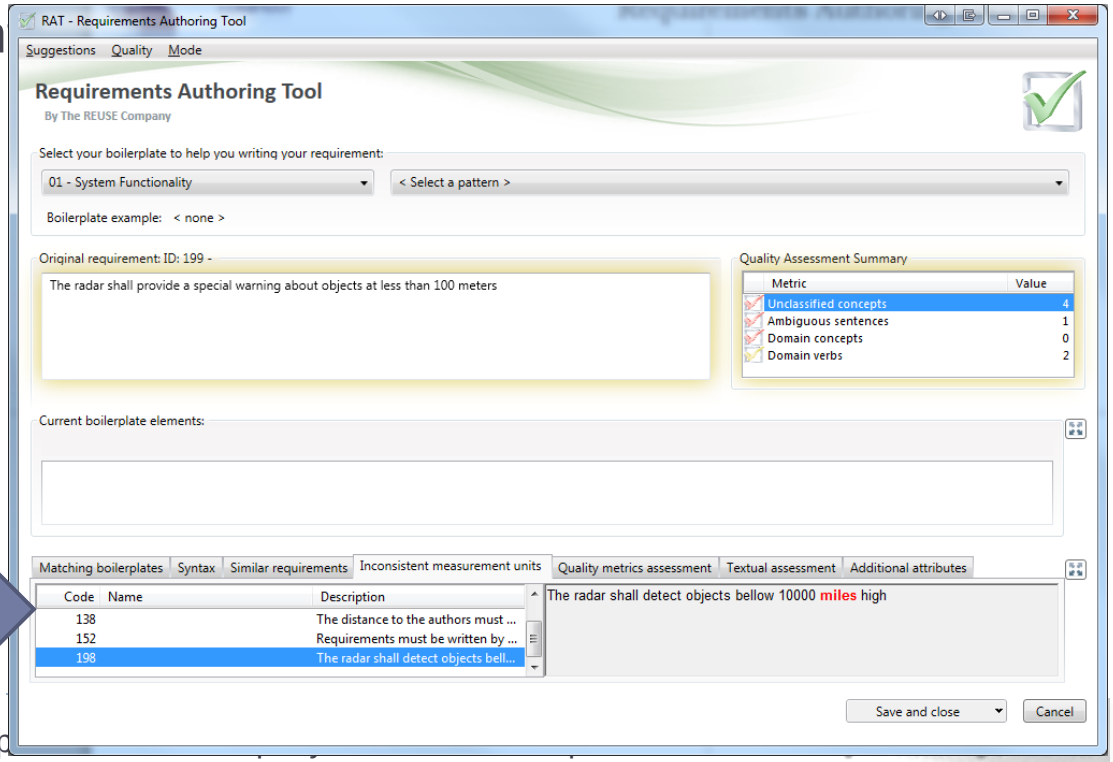
Requirements Consistency (inconsistence units)

Consistency: inconsistent units

- **Root problem:** inconsistent requirements could be difficult to find, therefore, the cost of finding them in later stages of the SDLC or even in a production environment is really high
- **Goal:** try to detect, in the same Requirements project, the use of non-consistent units (e.g. two different requirements measuring something in yards and meters)
- **Management:** RQA, out-of-the-box, already includes many of the most common measurement units. The user is able to extend this list at any moment
- **Solution:**
  - Show, on the fly, when your current requirement conflicts with other previous requirement in the same project

RAT – Inconsistent units

- Switch to *CCC* mode
- Write your requirement
- Click on *Inconsistent measurement unit* tab
- Click now on any of the inconsistent requirements to see the whole content of the requiremen



Requirements including inconsistent units

Overlapping requirements

Consistency: Overlapping Requirements

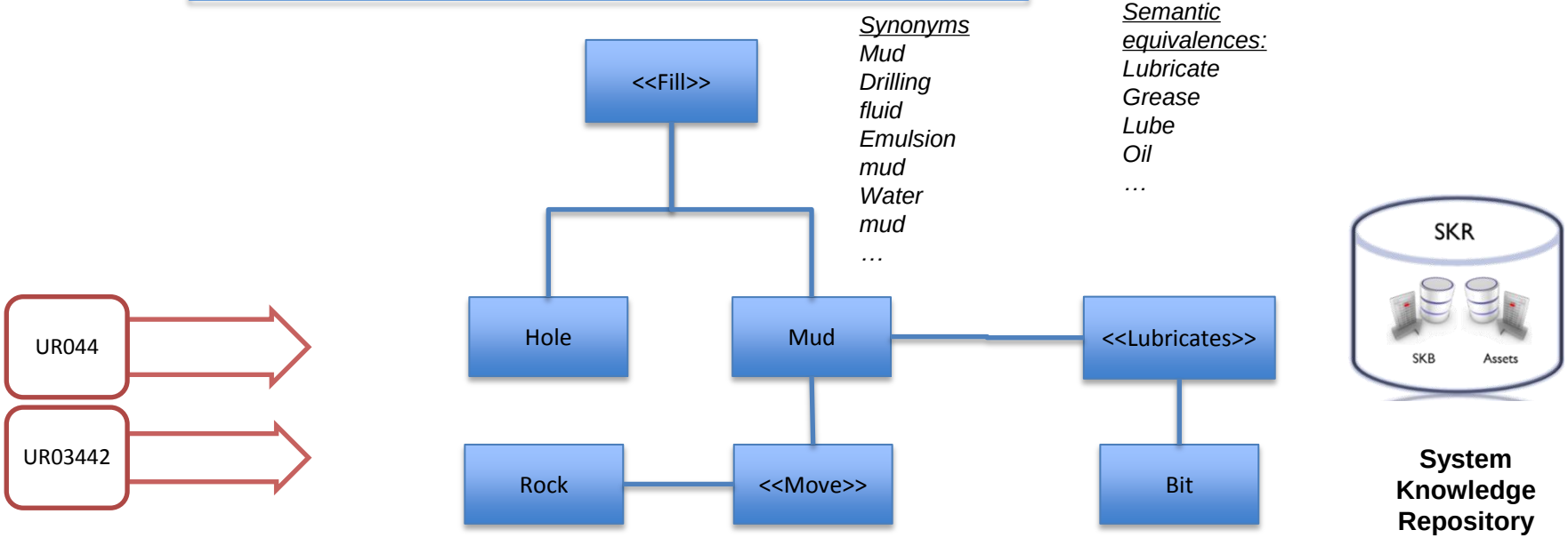
- **Root problem:** coupled specification could be the source of inconsistent specifications, therefore, the cause of many rework and poor quality projects
- **Goal:** automatically detect overlapping (inconsistent) requirements inside a single module or even among different modules or projects
- **Approach:**
  - Generate a semantic graph out of every single requirement: using linguistic techniques together with ontologies
  - This graphs don't relay on the words in the requirements, but in the real meaning (semantics) of a whole sentence
  - RQA compares those graphs to find out the semantic similarity among requirements
  - For two similar requirements, the parts of the graph are evaluated looking for inconsistencies
- **Solution:** once detected, the user can decide whether or not to go on with the current requirement, or remove a previous one

Consistency: Overlapping Requirements

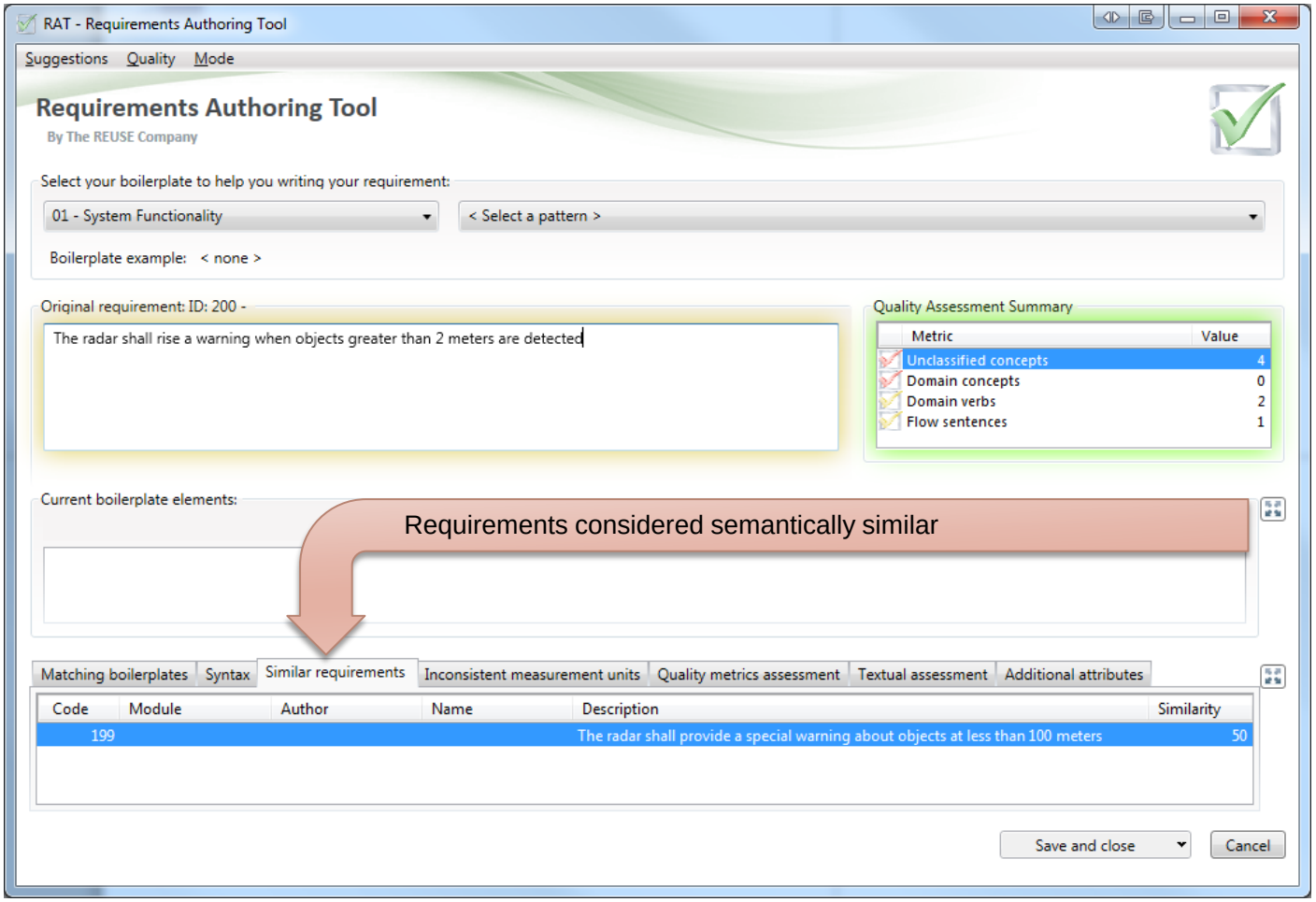
- Using requirements pattern to generate semantic graphs and find “similar” requirements

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**.

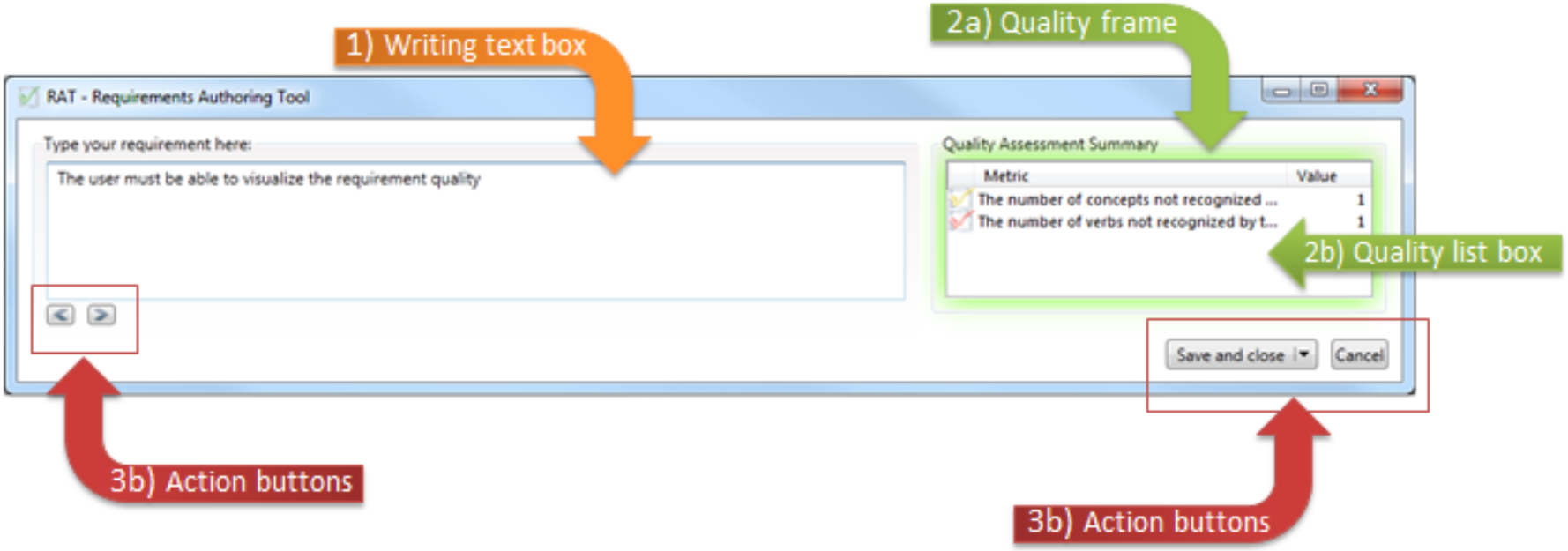


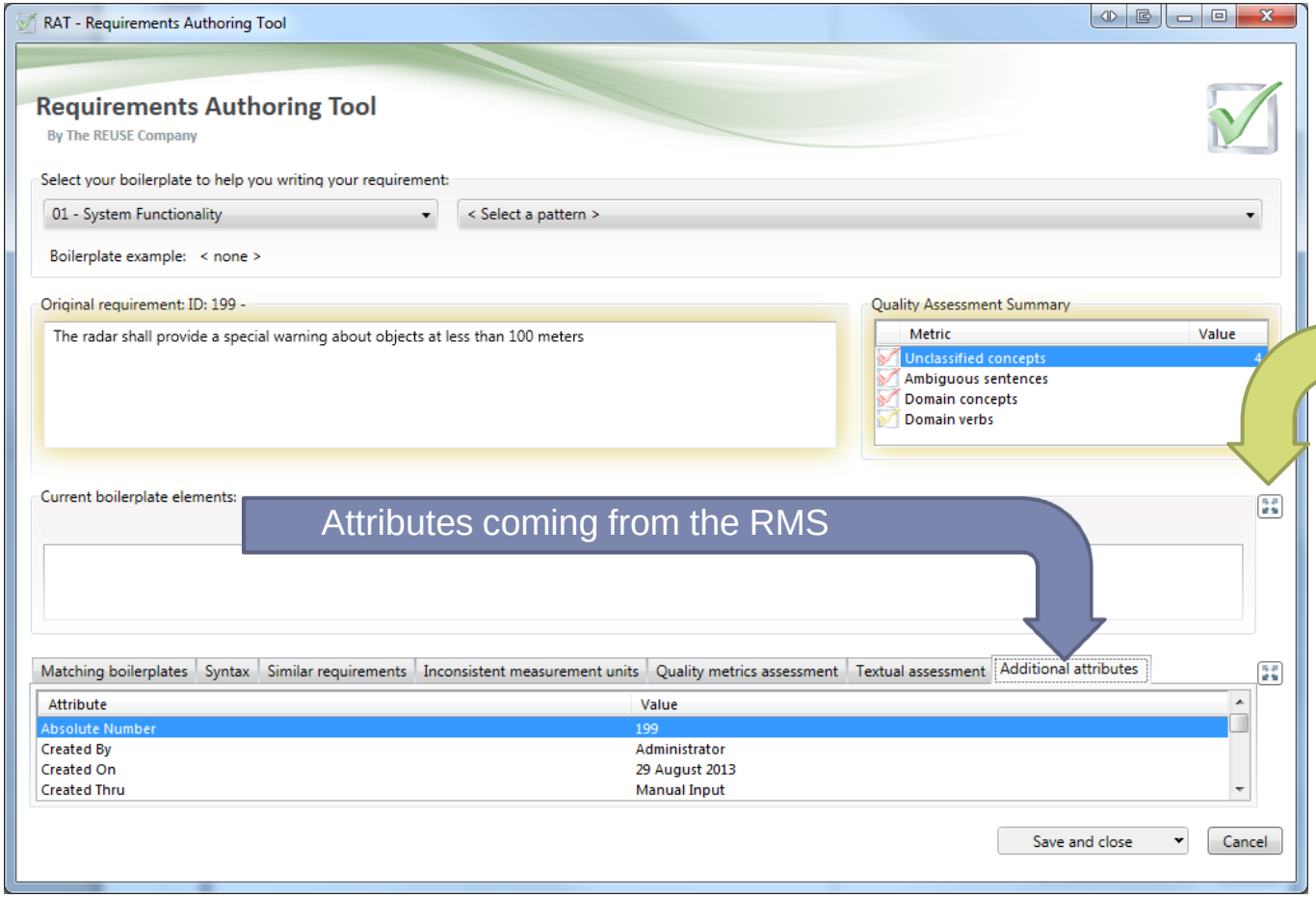
Consistency: Overlapping Requirements



Other actions and UI elements

Requirements Authoring Tool – RAT





Maximize/Minimize the section

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