



the
REUSE
company



From Requirements Quality to Requirements Authoring

By José Fuentes

Copyright © 2014 by (José Fuentes, The REUSE Company).
Published and used by The SSSE and INCOSE with permission

Table of Contents

- Introduction to The Reuse Company
- Requirements Quality
- Requirements Authoring
- RQS – Requirements Quality Suite

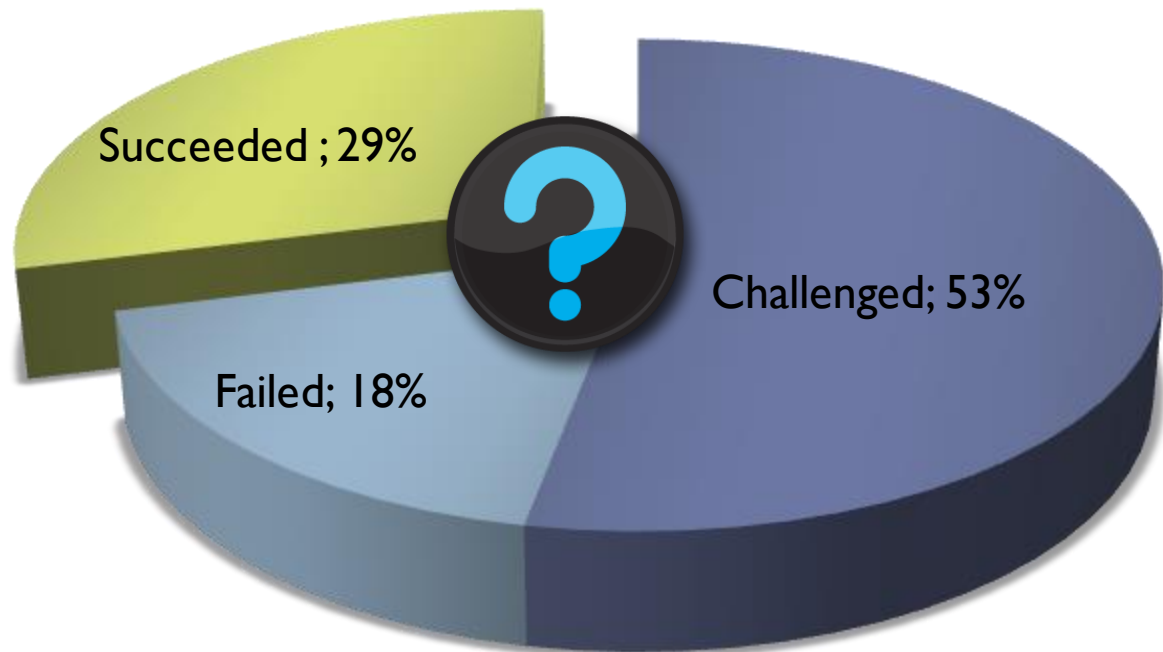
Introduction to The REUSE Company

- The REUSE Company, a *spinoff* company started in 1999
- Experts in:
 - Requirements Engineering,
 - Systems Engineering,
 - and mainly Reuse and Quality around Requirements and Systems Engineering
- Solutions and services related to these topics
- Creators of RQA and RQS

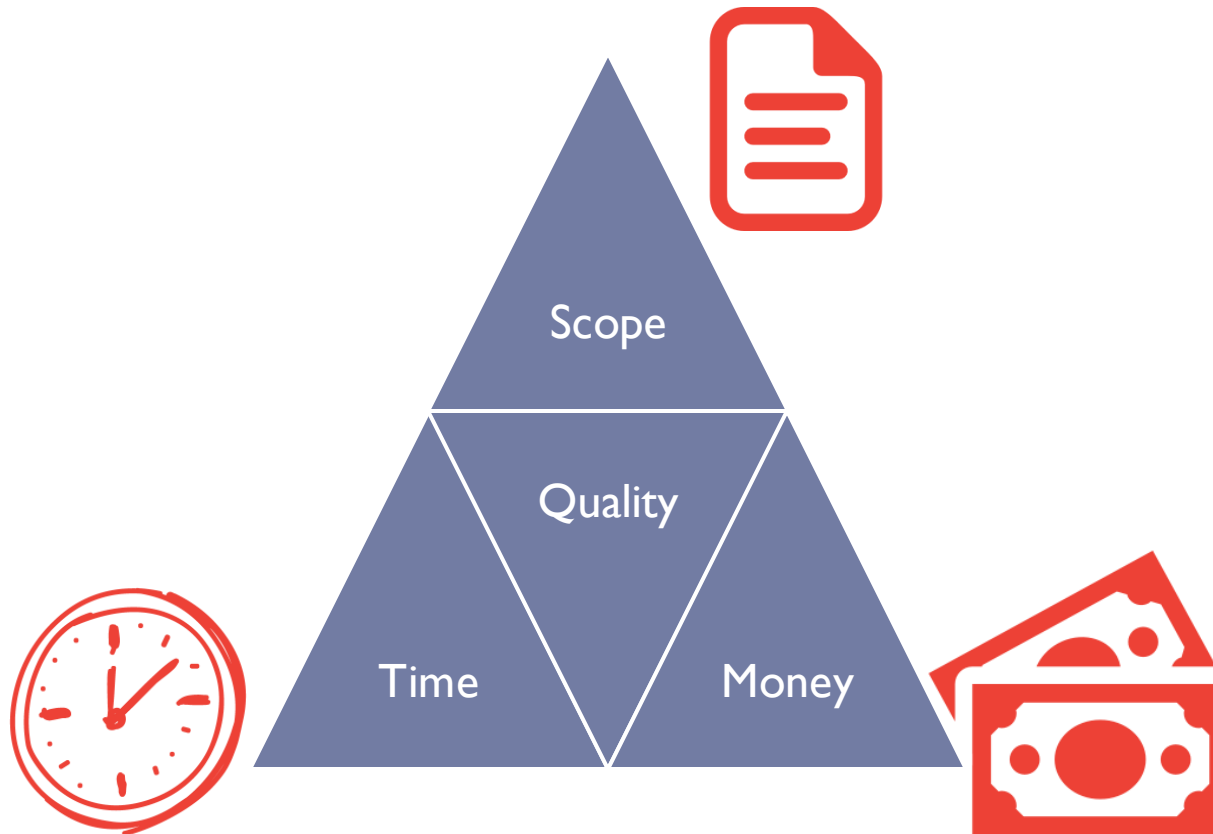
Requirements Quality

Requirements quality: Successful projects

Chaos Report, 2004



Requirements quality: Successful projects



Requirements Quality: source of defects

Project Success Factors	% of Responses	
1. User Involvement	15.9%	15.9%
2. Executive Management Support	13.9%	
3. Clear Statement of Requirements	13.0%	13.0%
4. Proper Planning	9.6%	
5. Realistic Expectations	8.2%	8.2 %
6. Smaller Project Milestones	7.7%	
7. Competent Staff	7.2%	
8. Ownership	5.3%	
9. Clear Vision & Objectives	2.9%	2.9 %
10. Hard-Working, Focused Staff	2.4%	
Other	13.9%	

(Source: CHAOS Report, 2004)

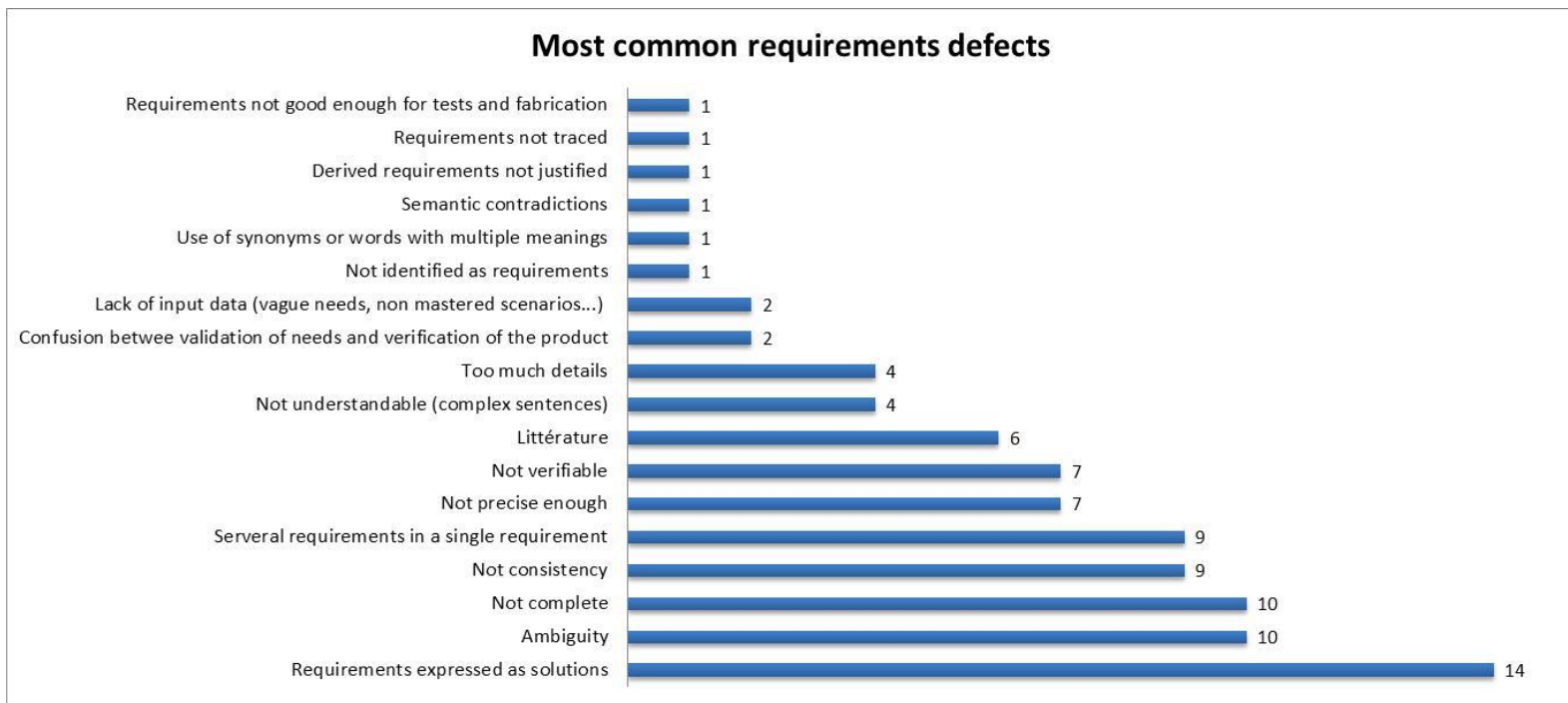
Based on requirements

+40% directly related with requirements definition and management

Requirements Quality: source of defects

Study in the scope of **RAMP project** (Requirements Analysis and Modeling Process) in partnership with Airbus Group, RENAULT, EDF, ADN, CORTIM, ENSTA, IRIT, PARIS 1 UNIVERSITY

(end 2010 over 22 industrials in several domains worldwide: interviews and questionnaires)



Requirements Quality characteristics

- ▶ IEEE Std. 830:

- ▶ Correct
- ▶ Unambiguous
- ▶ Complete
- ▶ Consistent
- ▶ Ranked
- ▶ Verifiable
- ▶ Modifiable
- ▶ Traceable

- ▶ ESA PSS-05:

- ▶ Pretty much the same characteristics

- ▶ SMART:

- ▶ Specific
- ▶ Measurable
- ▶ Aligned
- ▶ Realistic
- ▶ Time-limited



"I believe that this nation should commit itself to achieving the goal, before this decade is out, of landing a man on the Moon and returning him safely to Earth"

Requirements Quality characteristics

- ▶ Good characteristics to check but...
- ▶ Can we measure how correct, how complete, how consistent, how measurable... a specification is??
- ▶ Are those characteristics SMART?
 - ▶ Are they specific?
 - ▶ Easy to measure? From an objective point of view?
 - ▶ Is it realistic to ask for those characteristics?



Requirements Quality Metrics

- ▶ Different initiatives to use a set of easy-to-measure metrics/rules instead of the former *fuzzy* characteristics:
 - ▶ ARM (Automated Requirement Measurement) by NASA

INDICATORS OF QUALITY ATTRIBUTES											
Categories of Quality Indicators	Quality Attributes										
	1. Complete	2. Consistent	3. Correct	4. Modifiable	5. Ranked	6. Testable	7. Traceable	8. Unambiguous	9. Understandable	10. Validatable	11. Verifiable
1. Imperatives	X			X			X	X	X	X	X
2. Continuances	X			X	X	X	X	X	X	X	X
3. Directives	X		X			X		X	X	X	X
4. Options	X					X		X	X	X	
5. Weak Phrases	X		X			X		X	X	X	X
6. Size	X					X		X	X	X	X
7. Text Structure	X	X		X	X		X		X		X
8. Spec. Depth	X	X		X			X		X		X
9. Readability				X		X	X	X	X	X	X



Requirements Quality Metrics

- ▶ Different initiatives to use a set of easy-to-measure metrics/rules instead of the former characteristics:
 - ▶ Artemis EU Projects:



CESAR



CRYSTAL

- ▶ Classifying a number of different measurable rules into three main clusters:
 - ▶ Correctness: mainly for individual requirements
 - ▶ Consistency: mainly for whole specifications, but also with SysML models
 - ▶ Completeness: mainly for whole specifications, but also with SysML models

Requirements Quality Metrics

- ▶ Different initiatives to use a set of easy-to-measure metrics/rules instead of the former characteristics:
 - ▶ INCOSE Guide for Writing Requirements
 - ▶ Matching among characteristics and easy-to-measure rules








Requirements Quality Metrics

- ▶ Different initiatives to use a set of easy-to-measure metrics/rules instead of the former characteristics:
 - ▶ Génova et al.

Indicators	Function	Desirable properties										
		Atomicity	Precision	Completeness	Consistency	Understandability	Inambiguity	Traceability	Abstraction	Validity	Verifiability	Modifiability
Size	Convex	X	•	•	•	•	•	•	•	•	•	•
Readability	Incr./Decr.					X				•	•	
Punctuation	Convex					X				•	•	
Acron. & Abbrev.	Decreasing					X				•	•	
Connective terms	Decreasing	X	X	•	•	X	X	•	X	•	•	•
Imprecise terms	Decreasing		X	•	•	•	X			•	•	
Design terms	Decreasing								X		•	•
Imperative verbs	Convex	X	•	•	•	•	•	•	•	•	•	•
Conditional verbs	Decreasing		X	•	•	•	•			•	•	
Passive voice	Decreasing		X	•	•	•	•			•	•	
Domain terms	Convex	X	X	•	•	•	•	•	•	•	•	•
Versions	Decreasing									X	X	
Nesting	Convex					X				•	•	
Dependencies	Convex	X	•	•	•	X	•	X	•	•	•	•
Overlappings	Decreasing	X	•	•	•	X	X	X	•	•	•	•

Measurable indicators and related desirable properties:
 x = direct influence; • = indirect influence

Requirements Quality Metrics: a “perfect”... bad requirement

 Whatever the context, the system should be able to switch-off as quickly as possible, to restart without inducing any perturbation and at the same time to inform the operator in a friendly and understandable manner...    

Not so SMART requirement (Specific, Measurable, Aligned, Realistic, Time-limited)

Ambiguous (*What System ?*)

Several unclear requirements (*switch-off, restart, inform*)

No use of shall

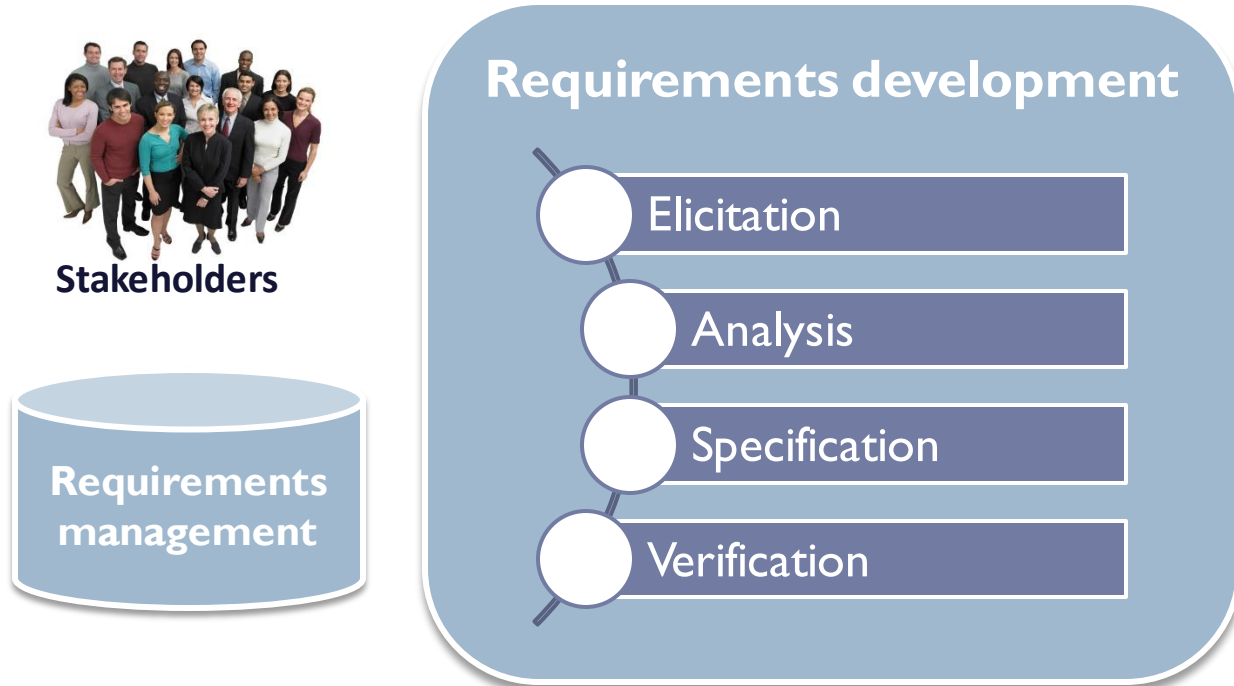
Not measurable (*quickly, same time,...*)

Not testable (*friendly, understandable,...*)

...

→ Need to perform a **Requirement Quality Analysis** against all the set of the system Requirements to **improve the requirement quality** before any delivery

Requirements Engineering Process



- It's good to automate the verification process but...
- ... it's even better to provide such a help to requirements authors

Requirements Authoring

Requirements Authoring

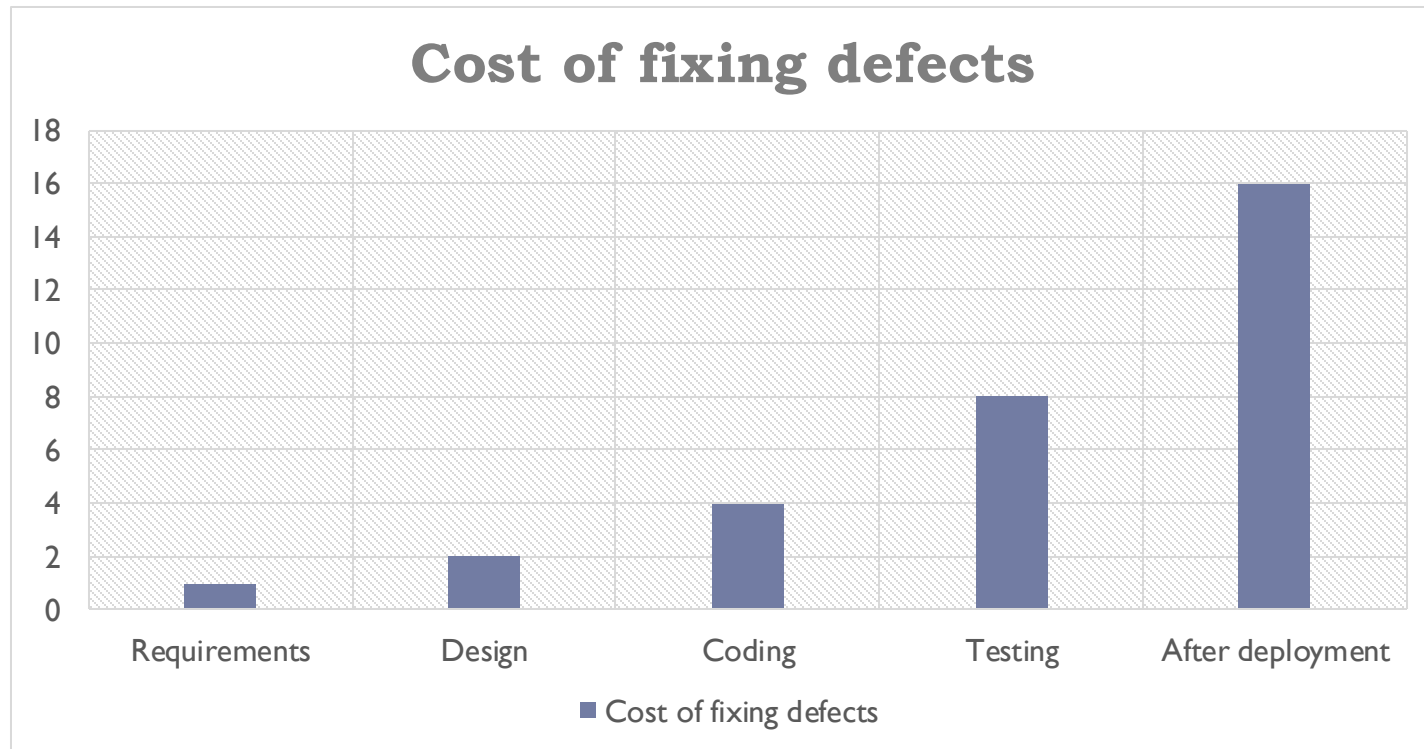
- Experiences shown that about **25% of system Requirements are critical and can grammatically be improved**
 - No Shall: 8 to 10%
 - Forbidden words: 10 to 15%
 - Subject, multiple objects, design: 15%
 - Incorrect grammar: 50%, ...

- Requirements **error costs are high**
 - Fixing requirements after delivery may cost up to **100 times more** than fixing in the requirements definition stage

- Training, best practices and **verifying requirements by reviews** can help to get SMART requirements:
 - But the process is **costly and time consuming**

- Introducing quality analysis during the **authoring** activity:
 - **Reduce the number of iterations between System Engineers and sub-contractors and improve the verification activities**

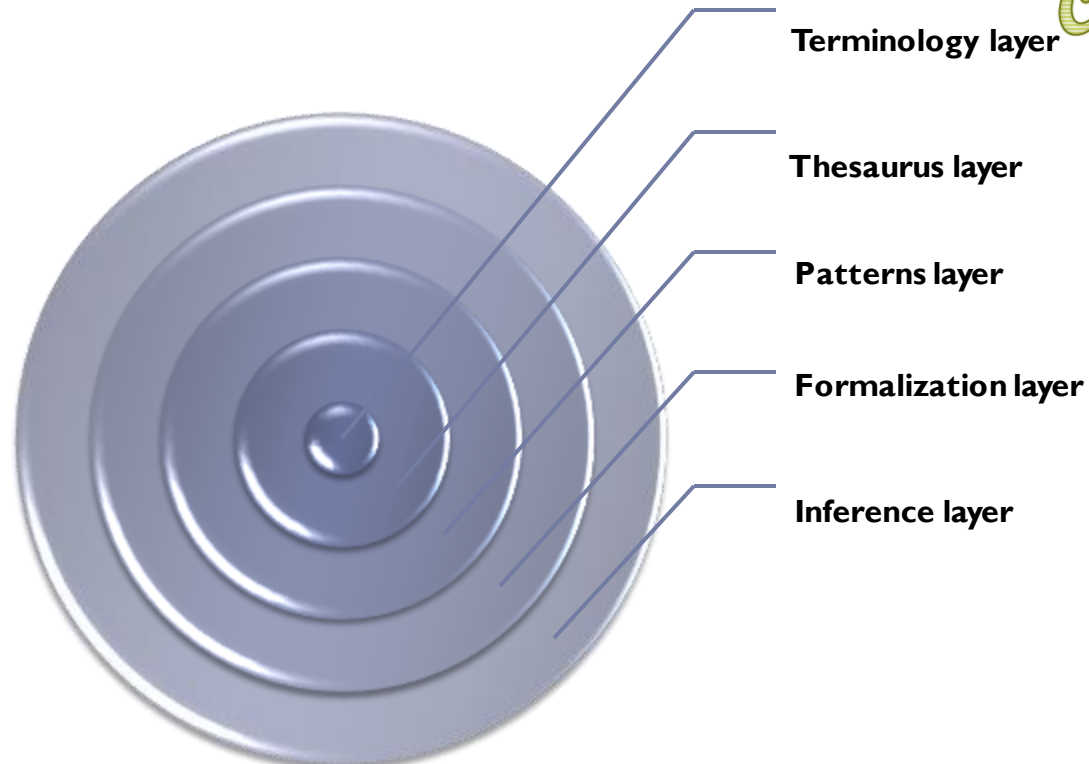
Requirements Authoring



- Classical figures for software engineering...
- ... But clearly too short in case of safety-critical systems

Requirements Authoring

- ▶ Authors of the specifications can be empowered by (1 of 4):
 - ▶ Checking a number of *correctness* issues on-the-fly *Correctness*
 - ▶ Using a consistent vocabulary through the use of a domain ontology *Consistency*

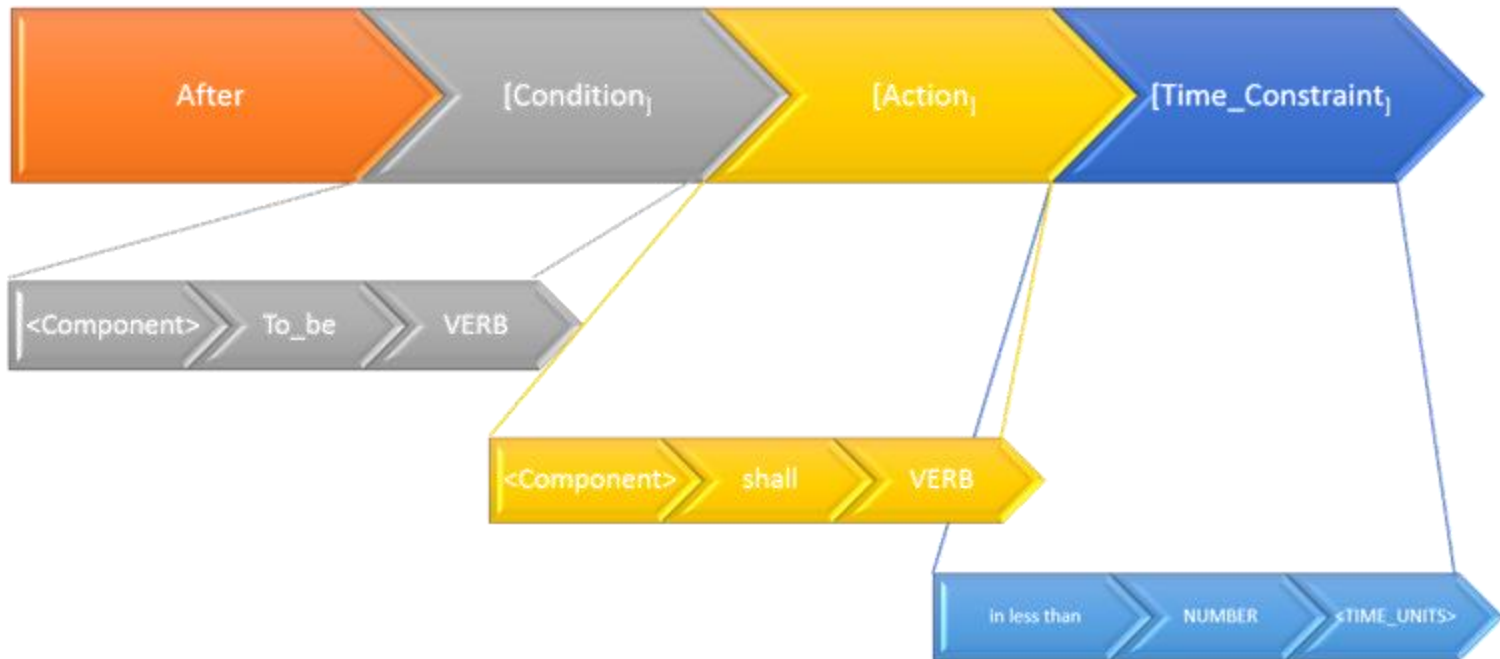


Requirements Authoring

- ▶ Authors of the specifications can be empowered by (2 of 4):
 - ▶ Using patterns to “force” a agreed way of writing
 - ▶ Providing all the expected data for the requirements, according to their types (e.g. performance information)

Consistency

Completeness



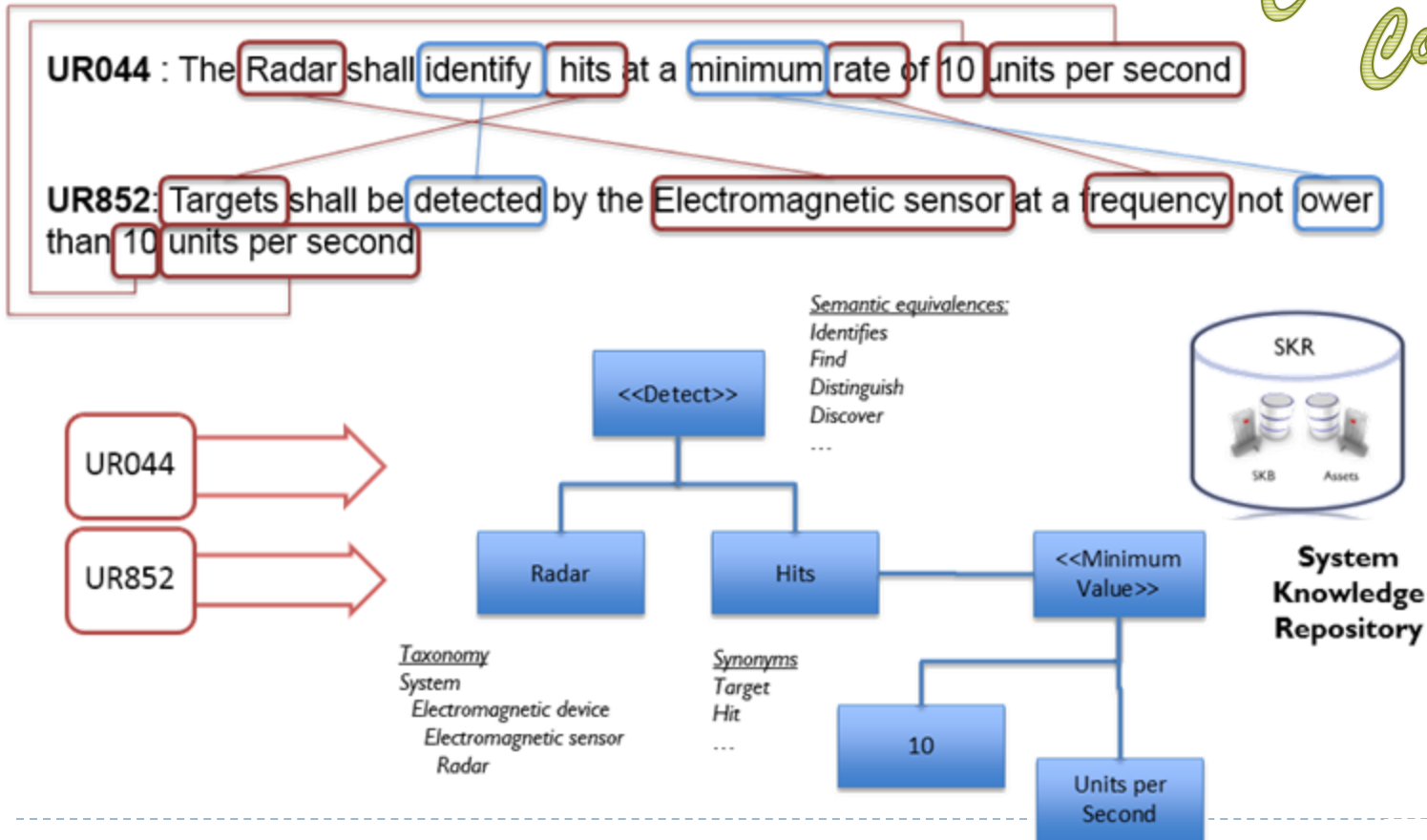
Requirements Authoring

- Authors of the specifications can be empowered by (3 of 4):
 - Identification of inconsistent information: *Consistency*
 - Duplicated requirements: by using a semantic search engine
 - Inconsistent content among requirements: e.g. inconsistent unit systems
 - Inconsistent content requirements vs. SysML: e.g. MTBF, weight of components...

Requirements Authoring

- Authors of the specifications can be empowered by (4 of 4):
 - Semantic reuse of requirements among previous projects

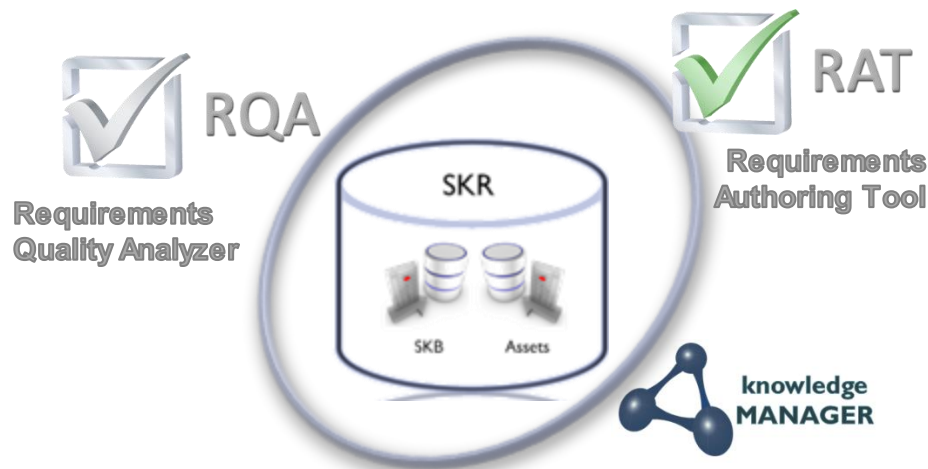
*Consistency
Completeness*



RQS – Requirements Quality Suite

Requirements Quality Suite

- The Requirements Quality Suite (RQS) intends to tackle requirements quality management by offering a set of tools and processes.
- RQS defines, measures, manages and improves requirements quality
- 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.

Requirements Authoring Tool (RAT):

to assist authors in the process of creating requirements with the right level of quality

knowledgeMANAGER (km):

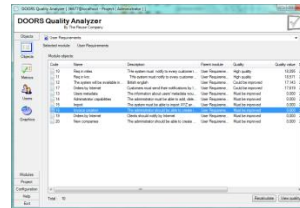
to manage knowledge around a requirements specification:

- the ontology it is based on
- the structure of the requirements to be used in the project
- the communication between authors and domain architects.

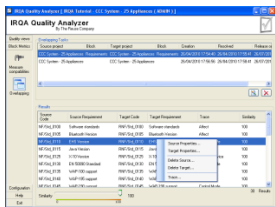
Requirements Quality Suite

➤ Metrics

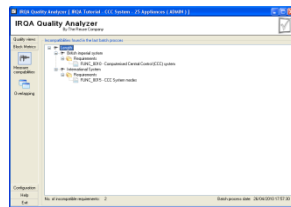
Correctness (individual metrics)



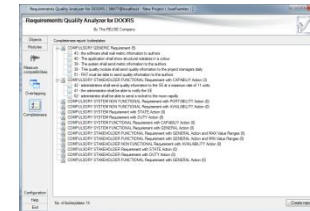
Consistency (semantic)



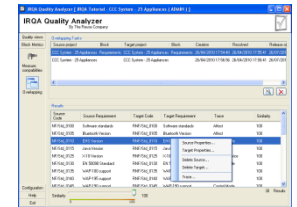
Consistency (inconsistent units)



Completeness (missing req.)

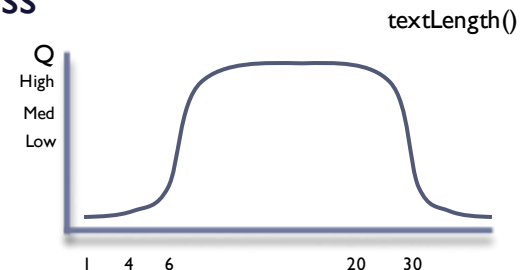


Completeness (missing links)



Requirements Quality Suite

- ▶ Types of metrics:
 - ▶ Automatic metrics
 - ▶ Parameterized metrics
 - ▶ *In-house coded* metrics
 - ▶ Manual metrics
- ▶ Metric customization:
 - ▶ Which metrics to use, metrics weight
 - ▶ Metrics limits/thresholds
 - ▶ How to parameterized some of the metrics: e.g. named links
 - ▶ Can include new *in-house coded* metrics
 - ▶ Manual metrics to support the further verification process
- ▶ Customization according to:
 - ▶ The maturity of every company or team
 - ▶ The type of requirements document: level of abstraction



Requirements Quality Suite

▶ **Example of correctness metrics:**

- ▶ Requirements size/length
- ▶ Readability
- ▶ Conditional vs. imperative sentences
- ▶ Active vs. passive voice
- ▶ Ambiguous sentences
- ▶ Optional sentences
- ▶ Subjective sentences
- ▶ Implicit sentences
- ▶ Abuse of connectors
- ▶ Negations
- ▶ Speculative sentences
- ▶ Use of false friends
- ▶ Design terms
- ▶ Flow terms
- ▶ Number of domain nouns and verbs
- ▶ Acronyms
- ▶ Hierarchical levels
- ▶ Volatility
- ▶ Number of dependences
- ▶ Forbidden Words
- ▶ Standard Requirement (match pattern)
- ▶ ...

Requirements Quality Suite

- ▶ Other features:
 - ▶ Connectors:



v/sure requirements ▶



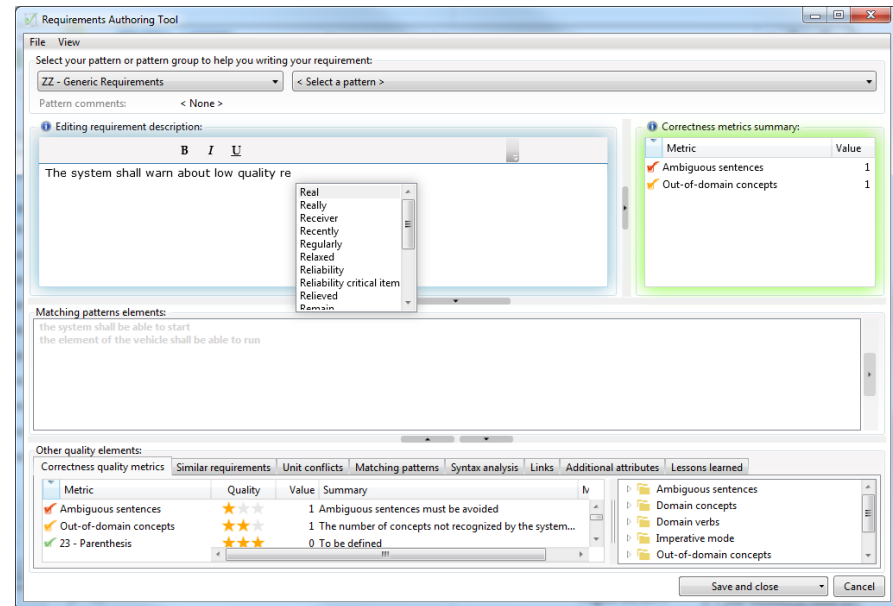
- ▶ Languages:



RAT : Requirements Authoring Tool

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

- ▶ 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





RAT : Requirements Authoring Tool

- ▶ Author assistance on the fly (typing requirement)

Pattern choice

The screenshot shows the RAT interface with the following components:

- Pattern choice:** A dropdown menu at the top left showing "ZZ - Generic Requirements" and a "Select a pattern" button.
- Editing requirement description:** A text area containing "The sys" with a dropdown menu showing "System" and "System element".
- Correctness metrics summary:** A table on the right showing metrics and their values:

Metric	Value
Domain concepts	0
Domain verbs	0
Imperative mode	0
Pattern matching	0
Text length (words)	1
- Matching patterns elements:** A list of pattern examples such as "[OPT] + NOUN | MODAL VERB «MODAL COMPULSORY» | Become | ADJECTIVE | O".
- Other quality elements:** A table at the bottom showing quality metrics:

Metric	Quality	Value	Summary
Domain concepts	★☆☆	0	At least one domain term must be used
Domain verbs	★☆☆	0	At least one domain verb must be used
Imperative mode	★☆☆	0	At least one imperative verb must be involved

Ontology terms to keep fulfilling the selected patterns

Valid paths to fulfill the selected patterns and pattern examples



RAT : Requirements Authoring Tool

► Quality assessment on the fly

The screenshot shows the 'Requirements Authoring Tool Edition Module' interface. The requirement description is: *Whatever the context, the system should be able to switch-off as quickly as possible, to restart without inducing any perturbation and at the same time to inform the operator in a friendly and understandable manner*. The correctness metrics summary table is as follows:

Name	Value
Absolute ambiguous sentences	4
Connectors	2
Absolute conditional sentences	1
Relative flow sentences by total number of terms	0
Relative speculative sentences by total number of terms	0
Relative incomplete sentences by total number of terms	0
Relative negative sentences by total number of terms	0
Relative imperative sentences by total number of verbs	0
Absolute imperative sentences	0

The 'Other quality elements' table is also visible:

Name	Value	Quality	Recommendation	Affects overall qu
Absolute design sentences	0	OK		<input type="checkbox"/>
Absolute flow sentences	0	OK		<input type="checkbox"/>
Absolute speculative sentences	0	OK		<input type="checkbox"/>
Dependencies	0	OK		<input type="checkbox"/>
Volatility (Versioning)	0	OK		<input type="checkbox"/>
Readability	16	OK		<input type="checkbox"/>
Absolute negative sentences	0	OK		<input type="checkbox"/>
Substantive sentences	0	OK		<input type="checkbox"/>

Bad quality (red)

Bad requirement



RAT : Requirements Authoring Tool

- ▶ Semantically similar requirements on the fly

Requirements Authoring Tool

Suggestions Quality View

Select your pattern or pattern group to help you writing your requirement:

All patterns < Select a pattern >

Editing requirement description:

The Requirements Authoring Tool shall be able to offer **quality assessment**

Correctness metrics summary:

Name	Value
<input checked="" type="checkbox"/> Absolute domain verbs	0
<input checked="" type="checkbox"/> Relative negative sentences by to...	0
<input checked="" type="checkbox"/> Relative speculative sentences by...	0
<input checked="" type="checkbox"/> Relative flow sentences by total n...	0
<input checked="" type="checkbox"/> Relative design sentences by tota...	0
<input checked="" type="checkbox"/> Absolute domain terms	0
<input checked="" type="checkbox"/> Relative incomplete sentences by...	0
<input checked="" type="checkbox"/> Relative domain terms by total n...	0
<input checked="" type="checkbox"/> Relative ambiguous sentences by...	20
<input checked="" type="checkbox"/> Relative imperative sentences by...	50
<input checked="" type="checkbox"/> Relative conditional sentences by...	0

Other quality elements:

Matching patterns Syntax analysis **Similar requirements** Unit conflicts Correctness quality metrics Textual assessment Additional attributes Traces

Code	Author	Description	Similarity
2096e8f2	KR\blopez	The RAT must be able to provide quality assessment	66

There's a requirement in the SKB very similar to the writing requirement

Save and close Cancel



RAT : Requirements Authoring Tool

► Inconsistent measurement units on the fly

Requirements Authoring Tool

Suggestions Quality View

Select your pattern or pattern group to help you writing your requirement:

All patterns < Select a pattern >

Editing requirement description:

The second rocket must be able to measure at least 20 inches

Correctness metrics summary:

Name	Value
Relative flow sentences by total nu...	0
Relative speculative sentences by t...	0
Relative design sentences by total...	0
Relative domain terms by total nu...	0
Relative conditional sentences by t...	0
Relative incomplete sentences by t...	0
Absolute domain terms	0
Relative negative sentences by tot...	0
Absolute ambiguous sentences	2
Text length	12
Relative ambiguous sentences by t...	25

Other quality elements:

Matching patterns Syntax analysis Similar requirements Unit conflicts Correctness quality metrics Textual assessment Additional attributes Traces

Code	Description
d734fd7c-5129-44b6	The plane must be able to fly at least 1600 kilometres without landing

The plane must be able to fly at least 1600 kilometres without landing

Conflictive issue (kilometres VS inches)

There's a requirement in the SKB conflicting with the writing requirement

Thank you for your attention!!

Questions??

