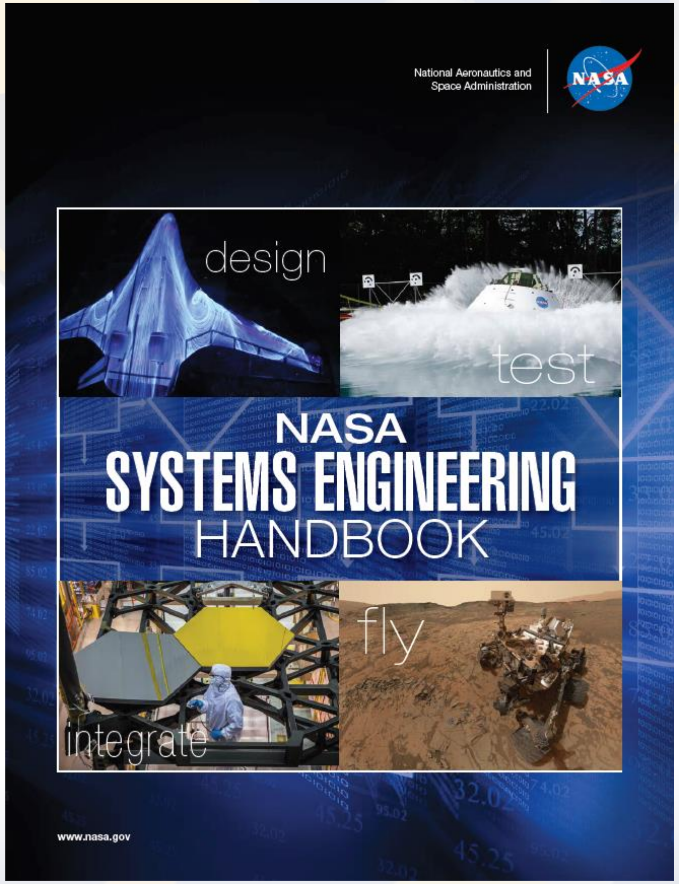
NASA Requirements Checklist library

THE REUSE COMPANY

The source : NASA Systems Engineering Handbook

- After some events and reports **NASA Office of the Chief Engineer (OCE)** took the initiative to **improve the overall Agency systems engineering** infrastructure and capability for the efficient and effective engineering of NASA systems.
- As part of this initiative the **initial writing** of NASA/SP-6105 was in **1995**.
- **Latest** version is **NASA SP-2016-6105 Rev2** (supersedes SP-2007-6105 Rev 1).
- **Objectives** highlighted in the handbook:
  - *"to provide **general guidance and information on systems engineering** that will be useful to the NASA community."*
  - *"to bring the **fundamental concepts and techniques of systems engineering** to NASA personnel in a way that recognized the nature of NASA systems and the NASA environment"*
- **NASA** defines **Systems Engineering** in the Handbook as:
  - ***"a methodical, multi-disciplinary approach for the design, realization, technical management, operations, and retirement of a system. A "system" is the combination of elements that function together to produce the capability required to meet a need."***

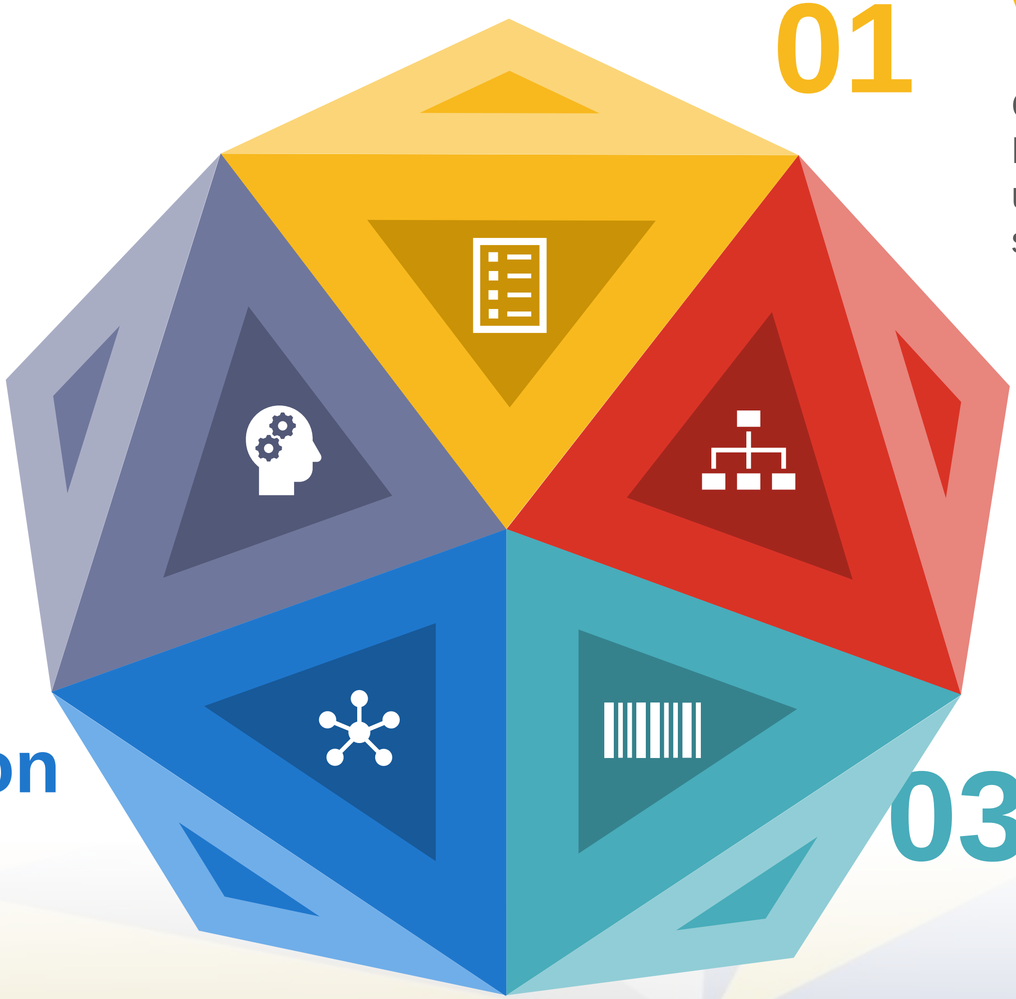


What is a Knowledge Library

- 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.

What is a Knowledge Library ?

What is a Knowledge Base



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. Stablish 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

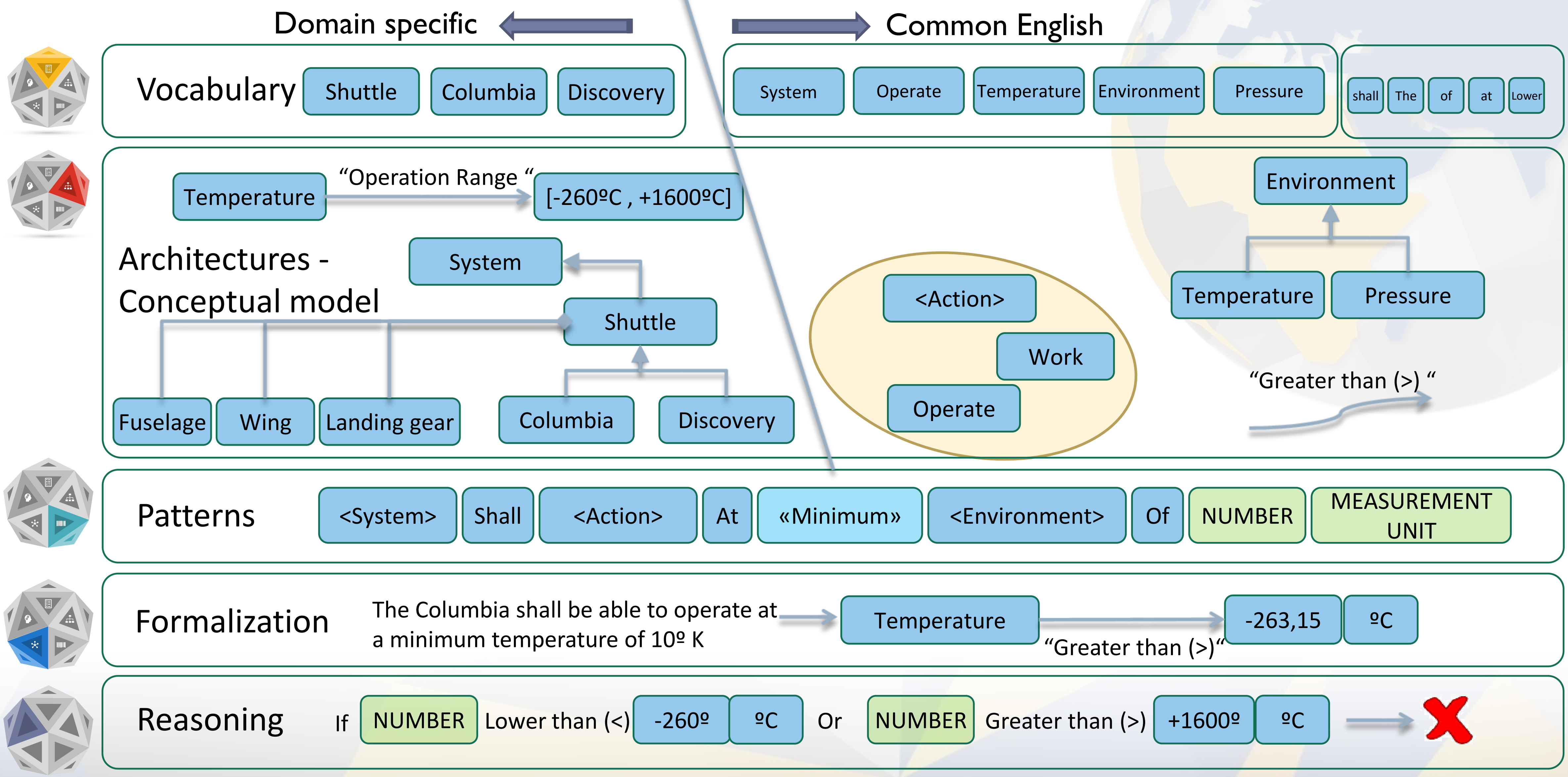
04 Formalization

Representation of assets semantic through SRL – System Representation Language

05 Reasoning

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

What is a Knowledge Library ?



Domain specific

Common English

Vocabulary

Shuttle Columbia Discovery

System Operate Temperature Environment Pressure

shall The of at Lower

Architectures - Conceptual model

Temperature “Operation Range “ [-260ºC , +1600ºC]

System

Shuttle

Fuselage Wing Landing gear Columbia Discovery

<Action>

Work

Operate

Environment

Temperature Pressure

“Greater than (>) “

Patterns

<System> Shall <Action> At «Minimum» <Environment> Of NUMBER MEASUREMENT UNIT

Formalization

The Columbia shall be able to operate at a minimum temperature of 10º K

Temperature “Greater than (>)“ -263,15 ºC

Reasoning

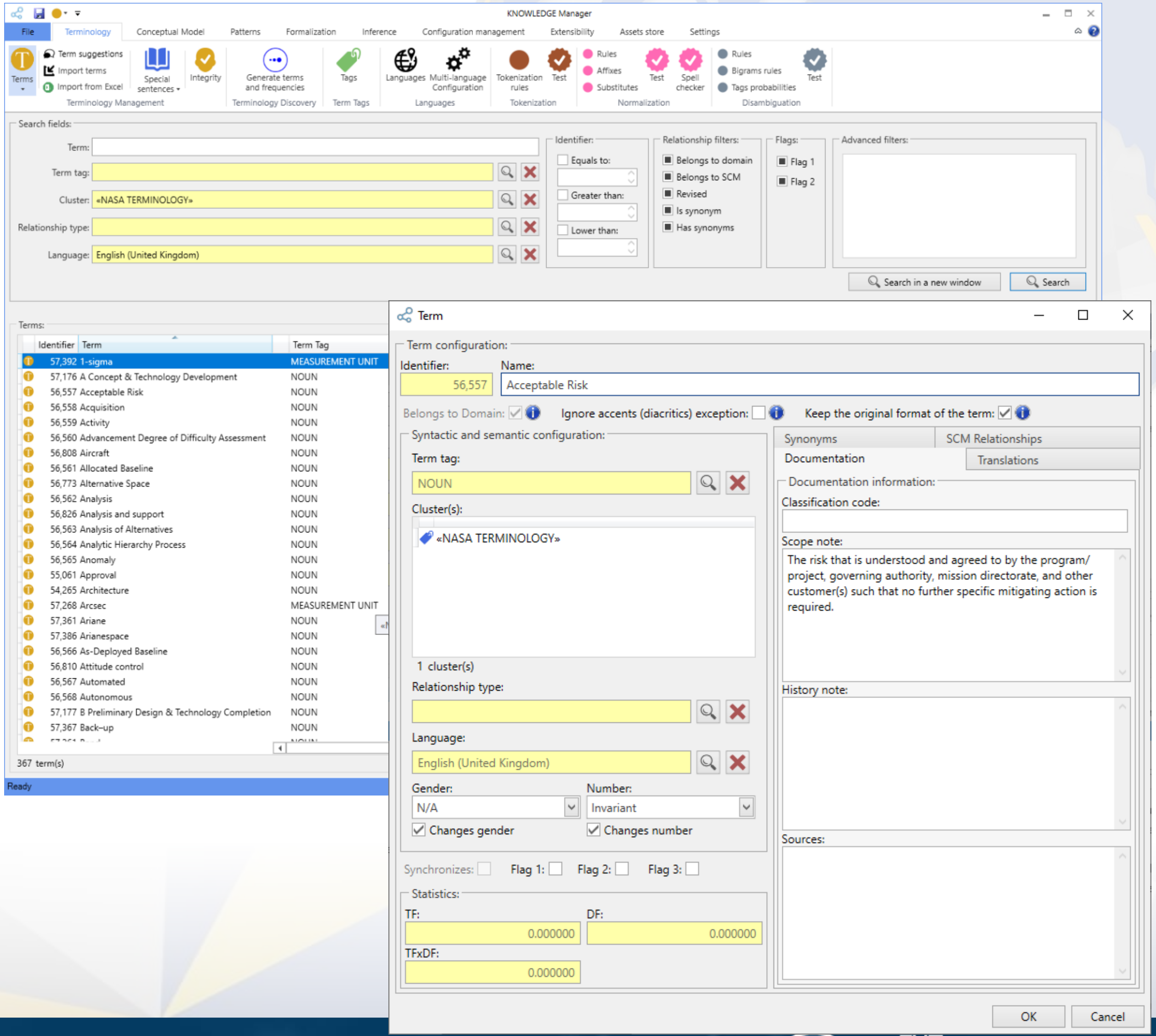
If NUMBER Lower than (<) -260º ºC Or NUMBER Greater than (>) +1600º ºC ✗

01

Vocabulary

Terms from the Handbook glossary have been included in the library

- **Provides a consistent way to name and understand all the concepts across the industry**
- **The system can highlight and link references to these entries in the body of the documents**



Search fields:

Term:

Term tag:

Cluster: «NASA TERMINOLOGY»

Relationship type:

Language: English (United Kingdom)

Identifier:
□ Equals to:
□ Greater than:
□ Lower than:

Relationship filters:
■ Belongs to domain
■ Belongs to SCM
■ Revised
■ Is synonym
■ Has synonyms

Flags:
■ Flag 1
■ Flag 2

Advanced filters:

Terms:

Identifier	Term	Term Tag
57,392	1-sigma	MEASUREMENT UNIT
57,176	A Concept & Technology Development	NOUN
56,557	Acceptable Risk	NOUN
56,558	Acquisition	NOUN
56,559	Activity	NOUN
56,560	Advancement Degree of Difficulty Assessment	NOUN
56,808	Aircraft	NOUN
56,561	Allocated Baseline	NOUN
56,773	Alternative Space	NOUN
56,562	Analysis	NOUN
56,826	Analysis and support	NOUN
56,563	Analysis of Alternatives	NOUN
56,564	Analytic Hierarchy Process	NOUN
56,565	Anomaly	NOUN
55,061	Approval	NOUN
54,265	Architecture	NOUN
57,268	Arcsec	MEASUREMENT UNIT
57,361	Ariane	NOUN
57,386	Arianespace	NOUN
56,566	As-Deployed Baseline	NOUN
56,810	Attitude control	NOUN
56,567	Automated	NOUN
56,568	Autonomous	NOUN
57,177	B Preliminary Design & Technology Completion	NOUN
57,367	Back–up	NOUN

367 term(s)

Ready

Term

Term configuration:

Identifier: 56,557

Name: Acceptable Risk

Belongs to Domain: ☑ Ignore accents (diacritics) exception: □ Keep the original format of the term: ☑

Syntactic and semantic configuration:

Term tag: NOUN

Cluster(s):
«NASA TERMINOLOGY»

1 cluster(s)

Relationship type:

Language: English (United Kingdom)

Gender: N/A

Number: Invariant

☑ Changes gender ☑ Changes number

Synchronizes: □ Flag 1: □ Flag 2: □ Flag 3: □

Statistics:

TF: 0.000000

DF: 0.000000

TFxDF: 0.000000

Synonyms | SCM Relationships | Documentation | Translations

Documentation information:

Classification code:

Scope note:
The risk that is understood and agreed to by the program/project, governing authority, mission directorate, and other customer(s) such that no further specific mitigating action is required.

History note:

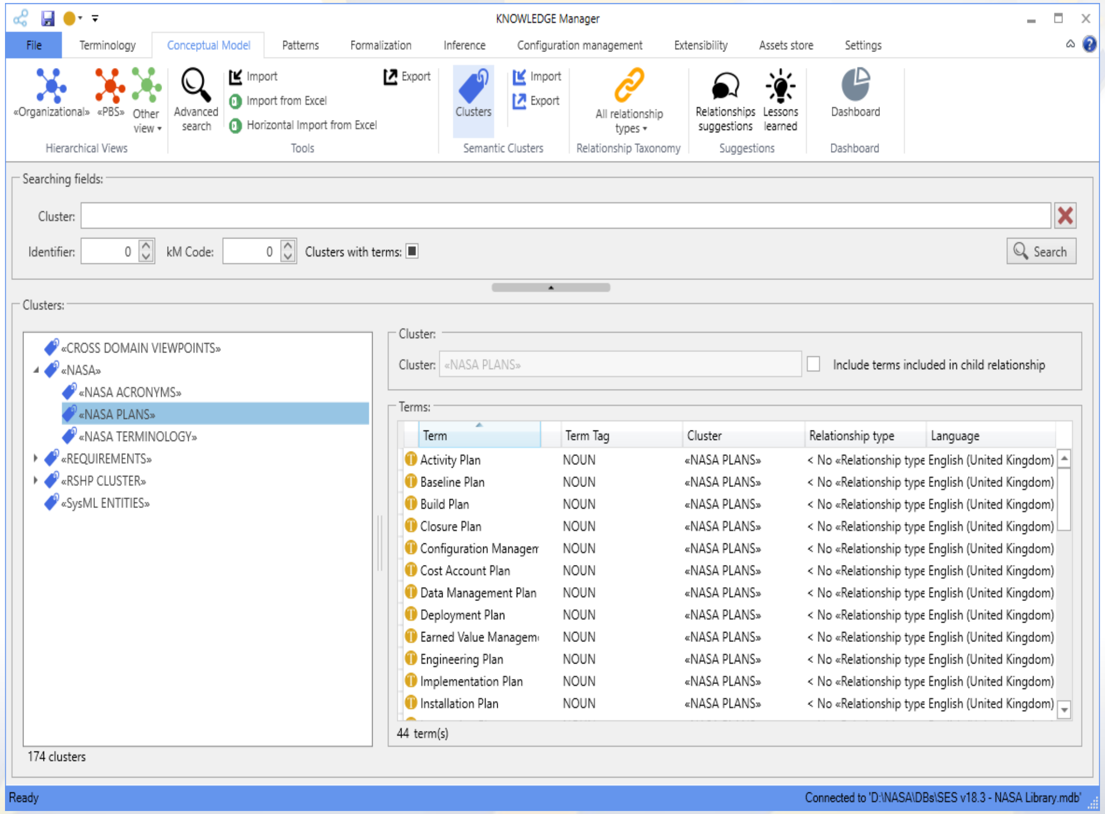
Sources:

02

Conceptual Models

- **Clustering:** according to the semantic of the terms in the library

- **Provides means to fit the textual paterns and help authors while the write requirements or other types of textual assets**



Searching fields:

Cluster:

Identifier: 0 kM Code: 0 Clusters with terms: ■

Clusters:

- «CROSS DOMAIN VIEWPOINTS»
- «NASA»
  - «NASA ACRONYMS»
  - «NASA PLANS»
  - «NASA TERMINOLOGY»
- «REQUIREMENTS»
- «RSHP CLUSTER»
- «SysML ENTITIES»

174 clusters

Cluster:

Cluster: «NASA PLANS» □ Include terms included in child relationship

Terms:

Term	Term Tag	Cluster	Relationship type	Language
Activity Plan	NOUN	«NASA PLANS»	< No «Relationship type	English (United Kingdom)
Baseline Plan	NOUN	«NASA PLANS»	< No «Relationship type	English (United Kingdom)
Build Plan	NOUN	«NASA PLANS»	< No «Relationship type	English (United Kingdom)
Closure Plan	NOUN	«NASA PLANS»	< No «Relationship type	English (United Kingdom)
Configuration Managem	NOUN	«NASA PLANS»	< No «Relationship type	English (United Kingdom)
Cost Account Plan	NOUN	«NASA PLANS»	< No «Relationship type	English (United Kingdom)
Data Management Plan	NOUN	«NASA PLANS»	< No «Relationship type	English (United Kingdom)
Deployment Plan	NOUN	«NASA PLANS»	< No «Relationship type	English (United Kingdom)
Earned Value Managem	NOUN	«NASA PLANS»	< No «Relationship type	English (United Kingdom)
Engineering Plan	NOUN	«NASA PLANS»	< No «Relationship type	English (United Kingdom)
Implementation Plan	NOUN	«NASA PLANS»	< No «Relationship type	English (United Kingdom)
Installation Plan	NOUN	«NASA PLANS»	< No «Relationship type	English (United Kingdom)

44 term(s)

Ready

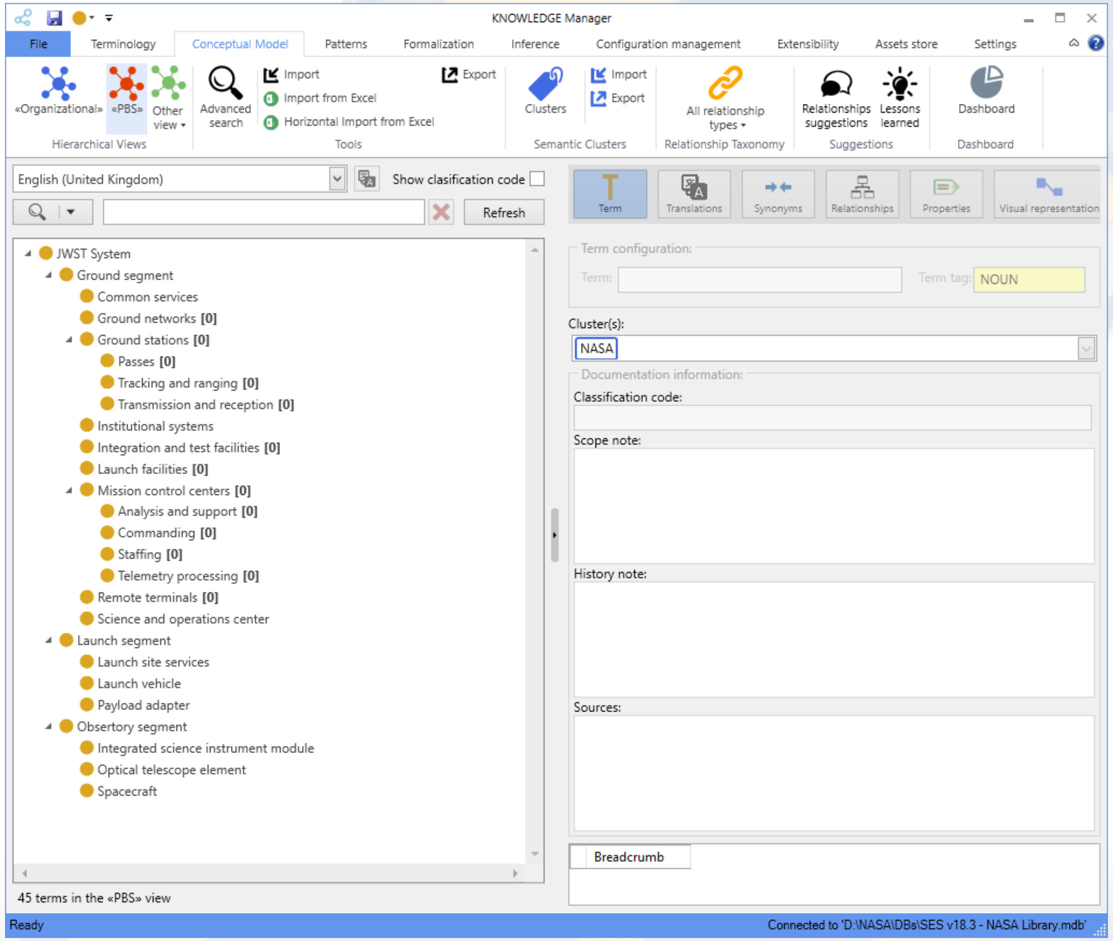
Connected to 'D:\NASA\DBs\SES v18.3 - NASA Library.mdb'

02

Conceptual Models

- Based on some relationships showed in the Handbook
- Including PBS views

- **Provides means to propagate queries in further reuse stages or just for information retrieval**



English (United Kingdom)

Show clasification code □

- JWST System
  - Ground segment
    - Common services
    - Ground networks **[0]**
    - Ground stations **[0]**
      - Passes **[0]**
      - Tracking and ranging **[0]**
      - Transmission and reception **[0]**
    - Institutional systems
    - Integration and test facilities **[0]**
    - Launch facilities **[0]**
    - Mission control centers **[0]**
      - Analysis and support **[0]**
      - Commanding **[0]**
      - Staffing **[0]**
      - Telemetry processing **[0]**
    - Remote terminals **[0]**
    - Science and operations center
  - Launch segment
    - Launch site services
    - Launch vehicle
    - Payload adapter
  - Obsertory segment
    - Integrated science instrument module
    - Optical telescope element
    - Spacecraft

45 terms in the «PBS» view

Term configuration:

Term:

Term tag: NOUN

Cluster(s): NASA

Documentation information:

Classification code:

Scope note:

History note:

Sources:

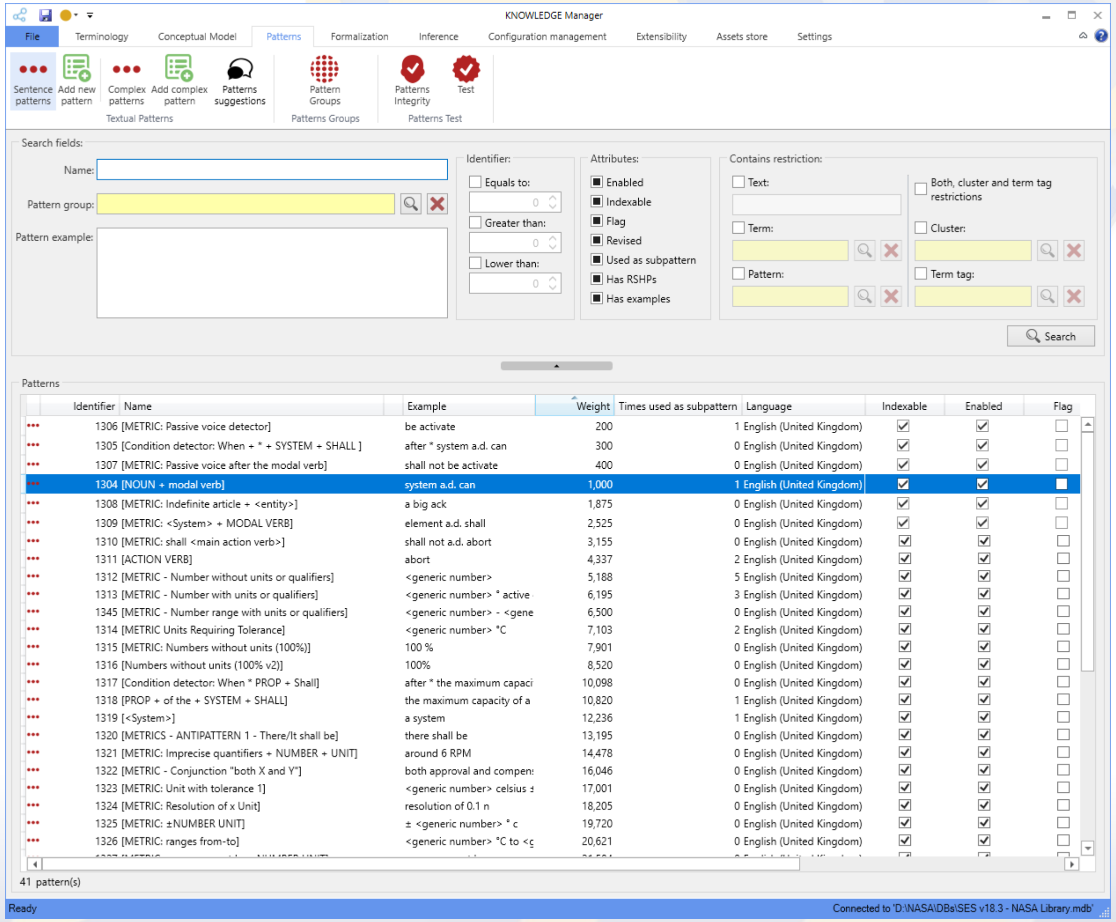
Ready

Connected to 'D:\NASA\DBs\SES v18.3 - NASA Library.mdb'

03 Patterns

- Patters for enabling the Handbook mapping metrics have been included in the library

- **Represent requirements similarities and enable formal representation, automatic recognition and aid authors**



Search fields:

Name:

Pattern group:

Pattern example:

Identifier:
☐ Equals to: 0
☐ Greater than: 0
☐ Lower than: 0

Attributes:
■ Enabled
■ Indexable
■ Flag
■ Revised
■ Used as subpattern
■ Has RSHPs
■ Has examples

Contains restriction:
☐ Text:
☐ Term:
☐ Pattern:
☐ Both, cluster and term tag restrictions
☐ Cluster:
☐ Term tag:

Patterns

Identifier	Name	Example	Weight	Times used as subpattern	Language	Indexable	Enabled	Flag
1306	[METRIC: Passive voice detector]	be activate	200	1	English (United Kingdom)	☑	☑	☐
1305	[Condition detector: When + * + SYSTEM + SHALL]	after * system a.d. can	300	0	English (United Kingdom)	☑	☑	☐
1307	[METRIC: Passive voice after the modal verb]	shall not be activate	400	0	English (United Kingdom)	☑	☑	☐
1304	[NOUN + modal verb]	system a.d. can	1,000	1	English (United Kingdom)	☑	☑	☐
1308	[METRIC: Indefinite article + <entity>]	a big ack	1,875	0	English (United Kingdom)	☑	☑	☐
1309	[METRIC: <System> + MODAL VERB]	element a.d. shall	2,525	0	English (United Kingdom)	☑	☑	☐
1310	[METRIC: shall <main action verb>]	shall not a.d. abort	3,155	0	English (United Kingdom)	☑	☑	☐
1311	[ACTION VERB]	abort	4,337	2	English (United Kingdom)	☑	☑	☐
1312	[METRIC - Number without units or qualifiers]	<generic number>	5,188	5	English (United Kingdom)	☑	☑	☐
1313	[METRIC - Number with units or qualifiers]	<generic number> ° active	6,195	3	English (United Kingdom)	☑	☑	☐
1345	[METRIC - Number range with units or qualifiers]	<generic number> - <gene	6,500	0	English (United Kingdom)	☑	☑	☐
1314	[METRIC Units Requiring Tolerance]	<generic number> °C	7,103	2	English (United Kingdom)	☑	☑	☐
1315	[METRIC: Numbers without units (100%)]	100 %	7,901	0	English (United Kingdom)	☑	☑	☐
1316	[Numbers without units (100% v2)]	100%	8,520	0	English (United Kingdom)	☑	☑	☐
1317	[Condition detector: When * PROP + Shall]	after * the maximum capaci	10,098	0	English (United Kingdom)	☑	☑	☐
1318	[PROP + of the + SYSTEM + SHALL]	the maximum capacity of a	10,820	1	English (United Kingdom)	☑	☑	☐
1319	[<System>]	a system	12,236	1	English (United Kingdom)	☑	☑	☐
1320	[METRICS - ANTIPATTERN 1 - There/It shall be]	there shall be	13,195	0	English (United Kingdom)	☑	☑	☐
1321	[METRIC: Imprecise quantifiers + NUMBER + UNIT]	around 6 RPM	14,478	0	English (United Kingdom)	☑	☑	☐
1322	[METRIC - Conjunction "both X and Y"]	both approval and compen:	16,046	0	English (United Kingdom)	☑	☑	☐
1323	[METRIC: Unit with tolerance 1]	<generic number> celsius ±	17,001	0	English (United Kingdom)	☑	☑	☐
1324	[METRIC: Resolution of x Unit]	resolution of 0.1 n	18,205	0	English (United Kingdom)	☑	☑	☐
1325	[METRIC: ±NUMBER UNIT]	± <generic number> ° c	19,720	0	English (United Kingdom)	☑	☑	☐
1326	[METRIC: ranges from-to]	<generic number> °C to <g	20,621	0	English (United Kingdom)	☑	☑	☐

41 pattern(s)

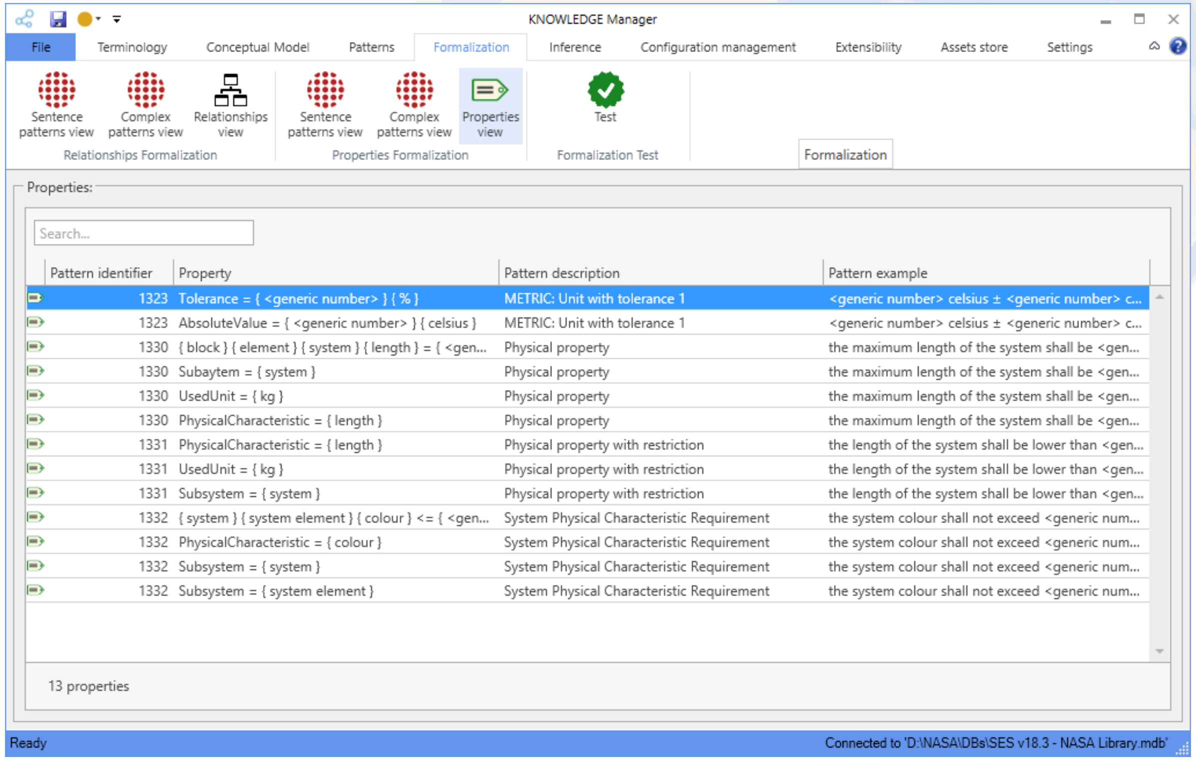
Ready

Connected to 'D:\NASA\DBs\SES v18.3 - NASA Library.mdb'

04 Formalization

- Formalization extracts relations and properties for enabling the metrics that have been included in the library

- **Representation of assets semantic through SRL – System Representation Language**



Pattern identifier	Property	Pattern description	Pattern example
1323	Tolerance = { <generic number> } { % }	METRIC: Unit with tolerance 1	<generic number> celsius ± <generic number> c...
1323	AbsoluteValue = { <generic number> } { celsius }	METRIC: Unit with tolerance 1	<generic number> celsius ± <generic number> c...
1330	{ block } { element } { system } { length } = { <gen...	Physical property	the maximum length of the system shall be <gen...
1330	Subaytem = { system }	Physical property	the maximum length of the system shall be <gen...
1330	UsedUnit = { kg }	Physical property	the maximum length of the system shall be <gen...
1330	PhysicalCharacteristic = { length }	Physical property	the maximum length of the system shall be <gen...
1331	PhysicalCharacteristic = { length }	Physical property with restriction	the length of the system shall be lower than <gen...
1331	UsedUnit = { kg }	Physical property with restriction	the length of the system shall be lower than <gen...
1331	Subsystem = { system }	Physical property with restriction	the length of the system shall be lower than <gen...
1332	{ system } { system element } { colour } <= { <gen...	System Physical Characteristic Requirement	the system colour shall not exceed <generic num...
1332	PhysicalCharacteristic = { colour }	System Physical Characteristic Requirement	the system colour shall not exceed <generic num...
1332	Subsystem = { system }	System Physical Characteristic Requirement	the system colour shall not exceed <generic num...
1332	Subsystem = { system element }	System Physical Characteristic Requirement	the system colour shall not exceed <generic num...

13 properties

Ready

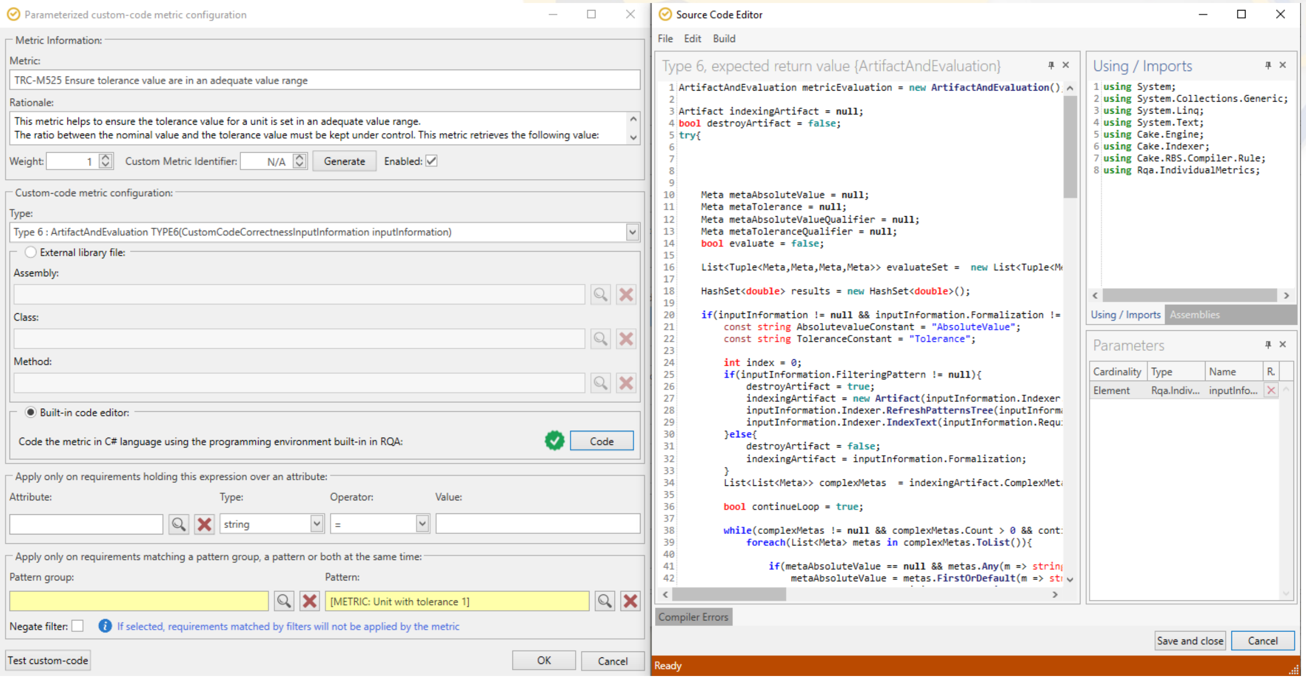
Connected to 'D:\NASA\DBs\SES v18.3 - NASA Library.mdb'

05

Reasoning

- Reasoning & advanced semantic feature the Handbook mapping metrics included in the library

- **A combination of rules, tasks and groups to infer information from valuable assets**



Parameterized custom-code metric configuration

Metric Information:

Metric: TRC-M525 Ensure tolerance value are in an adequate value range

Rationale: This metric helps to ensure the tolerance value for a unit is set in an adequate value range.
The ratio between the nominal value and the tolerance value must be kept under control. This metric retrieves the following value:

Weight: 1 Custom Metric Identifier: N/A Generate Enabled: ☑

Custom-code metric configuration:

Type: Type 6 : ArtifactAndEvaluation TYPE6(CustomCodeCorrectnessInputInformation inputInformation)

○ External library file:

Assembly:

Class:

Method:

● Built-in code editor:

Code the metric in C# language using the programming environment built-in in RQA: ✓ Code

Apply only on requirements holding this expression over an attribute:

Attribute: Type: string Operator: = Value:

Apply only on requirements matching a pattern group, a pattern or both at the same time:

Pattern group: Pattern: [METRIC: Unit with tolerance 1]

Negate filter: □ If selected, requirements matched by filters will not be applied by the metric

Test custom-code OK Cancel

Source Code Editor

File Edit Build

Type 6, expected return value {ArtifactAndEvaluation}

```
ArtifactAndEvaluation metricEvaluation = new ArtifactAndEvaluation()

Artifact indexingArtifact = null;
bool destroyArtifact = false;
try{




    Meta metaAbsoluteValue = null;
    Meta metaTolerance = null;
    Meta metaAbsoluteValueQualifier = null;
    Meta metaToleranceQualifier = null;
    bool evaluate = false;

    List<Tuple<Meta,Meta,Meta,Meta>> evaluateSet =  new List<Tuple<Me

    HashSet<double> results = new HashSet<double>();

    if(inputInformation != null && inputInformation.Formalization !=
        const string AbsolutevalueConstant = "AbsoluteValue";
        const string ToleranceConstant = "Tolerance";

        int index = 0;
        if(inputInformation.FilteringPattern != null){
            destroyArtifact = true;
            indexingArtifact = new Artifact(inputInformation.Indexer
            inputInformation.Indexer.RefreshPatternsTree(inputInforma
            inputInformation.Indexer.IndexText(inputInformation.Requ
        }else{
            destroyArtifact = false;
            indexingArtifact = inputInformation.Formalization;
        }
        List<List<Meta>> complexMetas  = indexingArtifact.ComplexMeta

        bool continueLoop = true;

        while(complexMetas != null && complexMetas.Count > 0 && cont
            foreach(List<Meta> metas in complexMetas.ToList()){

                if(metaAbsoluteValue == null && metas.Any(m => string
                    metaAbsoluteValue = metas.FirstOrDefault(m => st
```

Compiler Errors

Ready

Using / Imports

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using Cake.Engine;
using Cake.Indexer;
using Cake.RBS.Compiler.Rule;
using Rqa.IndividualMetrics;
```

Using / Imports Assemblies

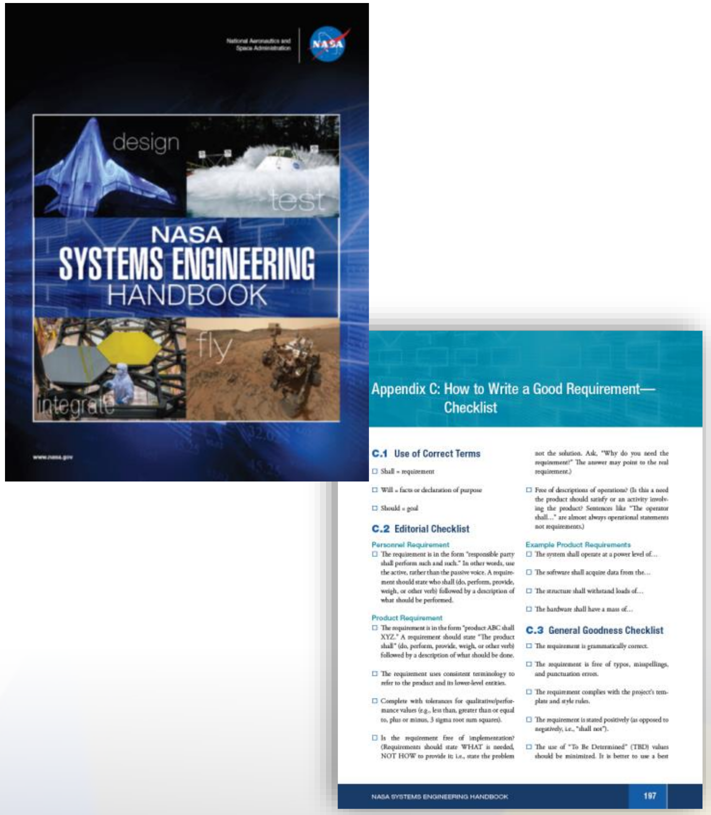
Parameters

Cardinality	Type	Name	R.
Element	Rqa.Indiv...	inputInfo...	✕

Save and close Cancel

The rules to automatically run the checklist

- Implementation into the CCC model of the TRC Systems Engineering Suite RQA (quality metrics)
- 42 metrics mapping the NASA SE Handbook requirements checklist (Appendix C)



Appendix C: How to Write a Good Requirement—Checklist

C.1 Use of Correct Terms

□ Shall = requirement

□ Will = facts or declaration of purpose

□ Should = goal

C.2 Editorial Checklist

Personnel Requirement

□ The requirement is in the form "responsible party shall perform such and such." In other words, use the active, rather than the passive voice. A requirement should state who shall (do, perform, provide, weigh, or other verb) followed by a description of what should be performed.

Product Requirement

□ The requirement is in the form "product ABC shall XYZ." A requirement should state "The product shall" (do, perform, provide, weigh, or other verb) followed by a description of what should be done.

□ The requirement uses consistent terminology to refer to the product and its lower-level entities.

□ Complete with tolerances for qualitative/performance values (e.g., less than, greater than or equal to, plus or minus, 3 sigma root sum squares).

□ Is the requirement free of implementation? (Requirements should state WHAT is needed, NOT HOW to provide it; i.e., state the problem not the solution. Ask, "Why do you need the requirement?" The answer may point to the real requirement.)

□ Free of descriptions of operations? (Is this a need the product should satisfy or an activity involving the product? Sentences like "The operator shall..." are almost always operational statements not requirements.)

Example Product Requirements

□ The system shall operate at a power level of...

□ The software shall acquire data from the...

□ The structure shall withstand loads of...

□ The hardware shall have a mass of...

C.3 General Goodness Checklist

□ The requirement is grammatically correct.

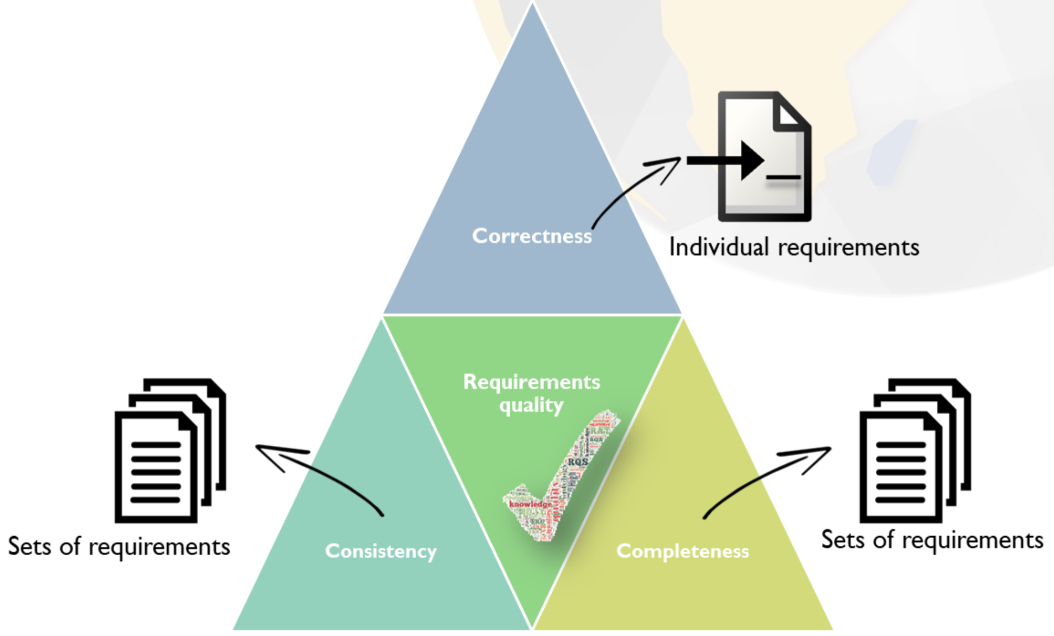
□ The requirement is free of typos, misspellings, and punctuation errors.

□ The requirement complies with the project's template and style rules.

□ The requirement is stated positively (as opposed to negatively, i.e., "shall not").

□ The use of "To Be Determined" (TBD) values should be minimized. It is better to use a best

NASA SYSTEMS ENGINEERING HANDBOOK 197

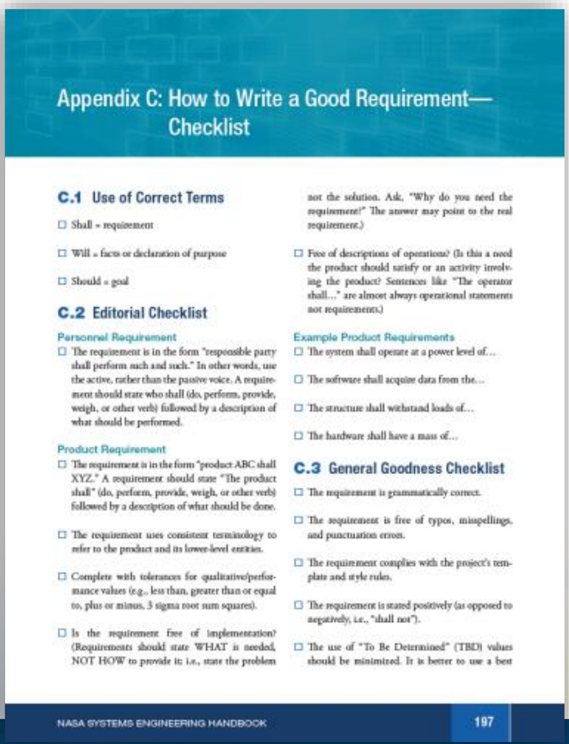


The implementation into the TRC tools consists in converting each characteristic of the checklist into **quantifiable sets of metrics** in order to create a quality assessment baseline.

NASA Requirements Checklist Library

The checklist : 4 main characteristics

- **Handbook section 4.2.1.2.3 Define Requirements in Acceptable Statements**
- *“the requirements should be defined in acceptable “shall” statements, which are complete sentences with a single “shall” per statement. Rationale for the requirement should also be captured to ensure the reason and context of the requirement is understood.”*
- **Appendix C: How to Write a Good Requirement— Checklist**



Appendix C: How to Write a Good Requirement— Checklist

C.1 Use of Correct Terms

□ Shall = requirement

□ Will = facts or declaration of purpose

□ Should = goal

C.2 Editorial Checklist

Personnel Requirement

□ The requirement is in the form “responsible party shall perform such and such.” In other words, use the active, rather than the passive voice. A requirement should state who shall (do, perform, provide, weigh, or other verb) followed by a description of what should be performed.

Product Requirement

□ The requirement is in the form “product ABC shall XYZ.” A requirement should state “The product shall” (do, perform, provide, weigh, or other verb) followed by a description of what should be done.

□ The requirement uses consistent terminology to refer to the product and its lower-level entities.

□ Complete with tolerances for qualitative/performance values (e.g., less than, greater than or equal to, plus or minus, 3 sigma root sum squares).

□ Is the requirement free of implementation? (Requirements should state WHAT is needed, NOT HOW to provide it; i.e., state the problem not the solution. Ask, “Why do you need the requirement?” The answer may point to the real requirement.)

□ Free of descriptions of operations? (Is this a need the product should satisfy or an activity involving the product? Sentences like “The operator shall...” are almost always operational statements not requirements.)

Example Product Requirements

□ The system shall operate at a power level of...

□ The software shall acquire data from the...

□ The structure shall withstand loads of...

□ The hardware shall have a mass of...

C.3 General Goodness Checklist

□ The requirement is grammatically correct.

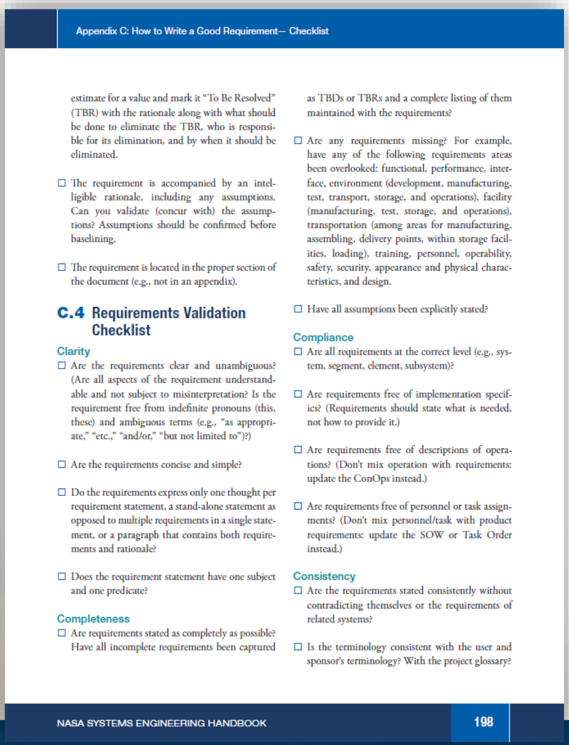
□ The requirement is free of typos, misspellings, and punctuation errors.

□ The requirement complies with the project’s template and style rules.

□ The requirement is stated positively (as opposed to negatively, i.e., “shall not”).

□ The use of “To Be Determined” (TBD) values should be minimized. It is better to use a best

NASA SYSTEMS ENGINEERING HANDBOOK 197



Appendix C: How to Write a Good Requirement— Checklist

estimate for a value and mark it “To Be Resolved” (TBR) with the rationale along with what should be done to eliminate the TBR, who is responsible for its elimination, and by when it should be eliminated.

□ The requirement is accompanied by an intelligible rationale, including any assumptions. Can you validate (concur with) the assumptions? Assumptions should be confirmed before baselining.

□ The requirement is located in the proper section of the document (e.g., not in an appendix).

C.4 Requirements Validation Checklist

Clarity

□ Are the requirements clear and unambiguous? (Are all aspects of the requirement understandable and not subject to misinterpretation? Is the requirement free from indefinite pronouns (this, these) and ambiguous terms (e.g., “as appropriate,” “etc.,” “and/or,” “but not limited to”)?)

□ Are the requirements concise and simple?

□ Do the requirements express only one thought per requirement statement, a stand-alone statement as opposed to multiple requirements in a single statement, or a paragraph that contains both requirements and rationale?

□ Does the requirement statement have one subject and one predicate?

Completeness

□ Are requirements stated as completely as possible? Have all incomplete requirements been captured as TBDs or TBRs and a complete listing of them maintained with the requirements?

□ Are any requirements missing? For example, have any of the following requirements areas been overlooked: functional, performance, interface, environment (development, manufacturing, test, transport, storage, and operations), facility (manufacturing, test, storage, and operations), transportation (among areas for manufacturing, assembling, delivery points, within storage facilities, loading), training, personnel, operability, safety, security, appearance and physical characteristics, and design.

□ Have all assumptions been explicitly stated?

Compliance

□ Are all requirements at the correct level (e.g., system, segment, element, subsystem)?

□ Are requirements free of implementation specifics? (Requirements should state what is needed, not how to provide it.)

□ Are requirements free of descriptions of operations? (Don’t mix operation with requirements: update the ConOps instead.)

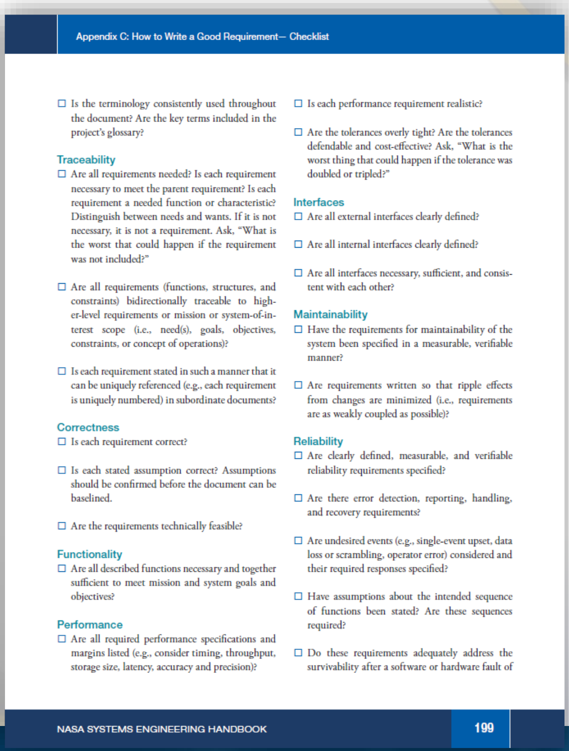
□ Are requirements free of personnel or task assignments? (Don’t mix personnel/task with product requirements: update the SOW or Task Order instead.)

Consistency

□ Are the requirements stated consistently without contradicting themselves or the requirements of related systems?

□ Is the terminology consistent with the user and sponsor’s terminology? With the project glossary?

NASA SYSTEMS ENGINEERING HANDBOOK 198



Appendix C: How to Write a Good Requirement— Checklist

□ Is the terminology consistently used throughout the document? Are the key terms included in the project’s glossary?

Traceability

□ Are all requirements needed? Is each requirement necessary to meet the parent requirement? Is each requirement a needed function or characteristic? Distinguish between needs and wants. If it is not necessary, it is not a requirement. Ask, “What is the worst that could happen if the requirement was not included?”

□ Are all requirements (functions, structures, and constraints) bidirectionally traceable to higher-level requirements or mission or system-of-interest scope (i.e., need(s), goals, objectives, constraints, or concept of operations)?

□ Is each requirement stated in such a manner that it can be uniquely referenced (e.g., each requirement is uniquely numbered) in subordinate documents?

Correctness

□ Is each requirement correct?

□ Is each stated assumption correct? Assumptions should be confirmed before the document can be baselined.

□ Are the requirements technically feasible?

Functionality

□ Are all described functions necessary and together sufficient to meet mission and system goals and objectives?

Performance

□ Are all required performance specifications and margins listed (e.g., consider timing, throughput, storage size, latency, accuracy and precision)?

□ Is each performance requirement realistic?

□ Are the tolerances overly tight? Are the tolerances defendable and cost-effective? Ask, “What is the worst thing that could happen if the tolerance was doubled or tripled?”

Interfaces

□ Are all external interfaces clearly defined?

□ Are all internal interfaces clearly defined?

□ Are all interfaces necessary, sufficient, and consistent with each other?

Maintainability

□ Have the requirements for maintainability of the system been specified in a measurable, verifiable manner?

□ Are requirements written so that ripple effects from changes are minimized (i.e., requirements are as weakly coupled as possible)?

Reliability

□ Are clearly defined, measurable, and verifiable reliability requirements specified?

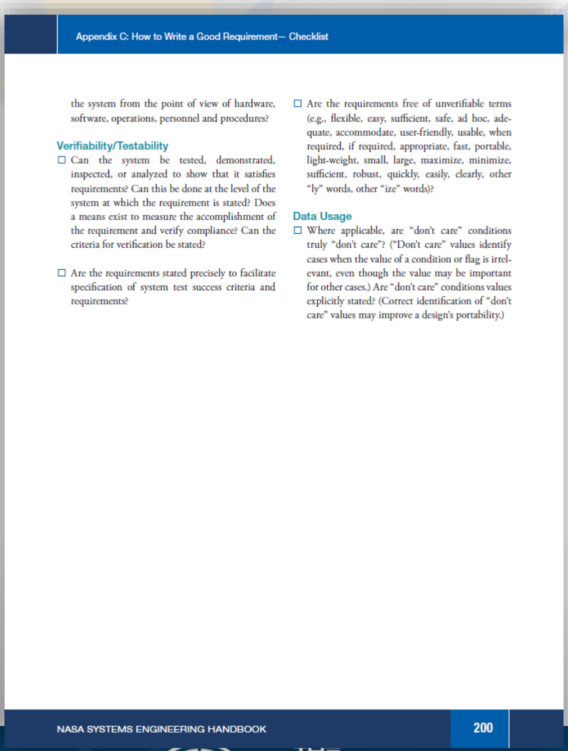
□ Are there error detection, reporting, handling, and recovery requirements?

□ Are undesired events (e.g., single-event upset, data loss or scrambling, operator error) considered and their required responses specified?

□ Have assumptions about the intended sequence of functions been stated? Are these sequences required?

□ Do these requirements adequately address the survivability after a software or hardware fault of

NASA SYSTEMS ENGINEERING HANDBOOK 199



Appendix C: How to Write a Good Requirement— Checklist

the system from the point of view of hardware, software, operations, personnel and procedures?

Verifiability/Testability

□ Can the system be tested, demonstrated, inspected, or analyzed to show that it satisfies requirements? Can this be done at the level of the system at which the requirement is stated? Does a means exist to measure the accomplishment of the requirement and verify compliance? Can the criteria for verification be stated?

□ Are the requirements stated precisely to facilitate specification of system test success criteria and requirements?

□ Are the requirements free of unverifiable terms (e.g., flexible, easy, sufficient, safe, ad hoc, adequate, accommodate, user-friendly, usable, when required, if required, appropriate, fast, portable, light-weight, small, large, maximize, minimize, sufficient, robust, quickly, easily, clearly, other “ly” words, other “ize” words)?

Data Usage

□ Where applicable, are “don’t care” conditions truly “don’t care”? (“Don’t care” values identify cases when the value of a condition or flag is irrelevant, even though the value may be important for other cases.) Are “don’t care” conditions values explicitly stated? (Correct identification of “don’t care” values may improve a design’s portability.)

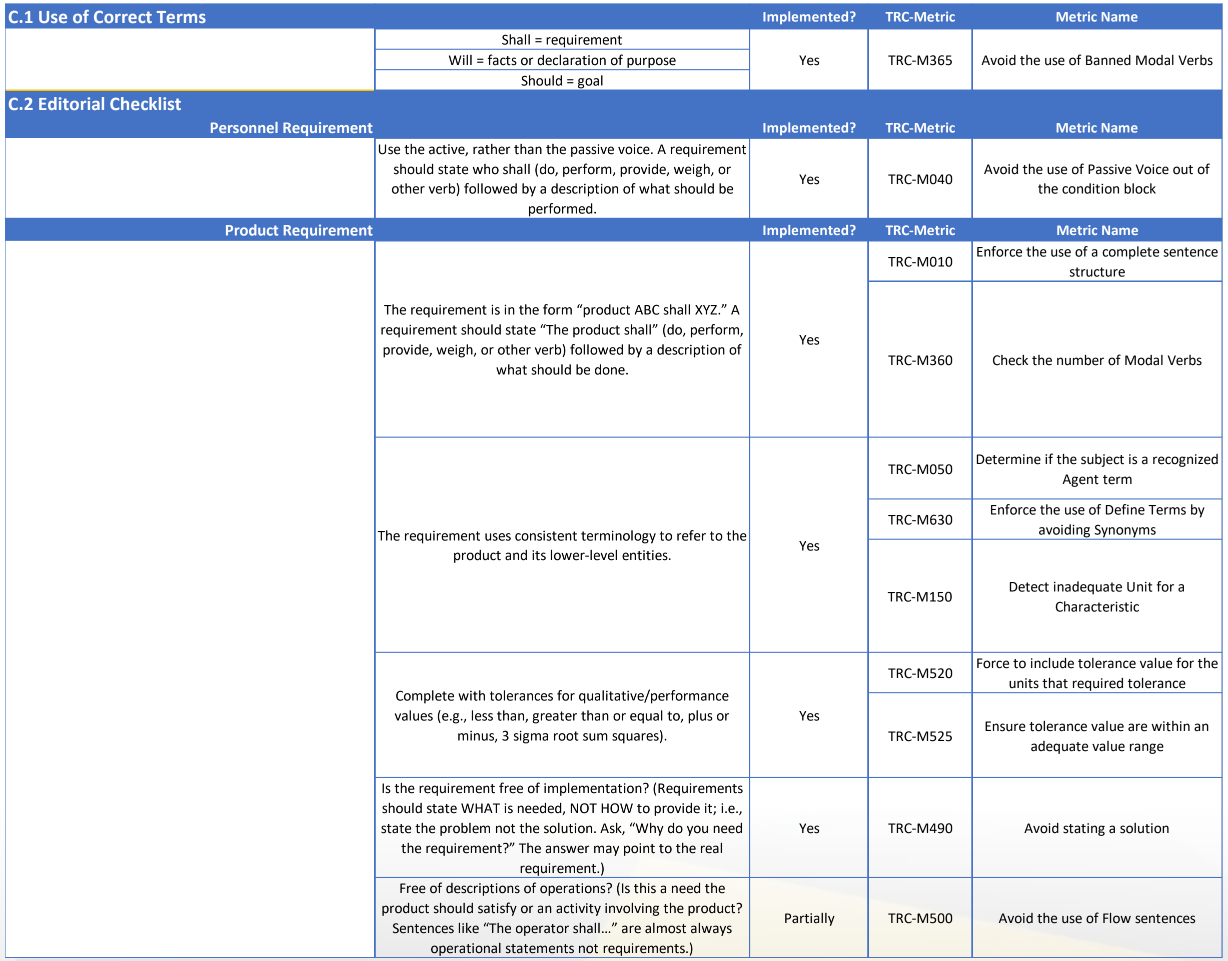
NASA SYSTEMS ENGINEERING HANDBOOK 200

The checklist : 4 main characteristics

C.1 Use of Correct Terms	
C.2 Editorial Checklist	
	Personnel Requirement
	Product Requirement
C.3 General Goodness Checklist	
C.4 Requirements Validation Checklist	
	Clarity
	Completeness
	Compliance
	Consistency
	Traceability
	Correctness
	Functionality
	Performance
	Interfaces
	Maintainability
	Reliability
	Verifiability/Testability
	Data Usage

NASA Requirements Checklist Library

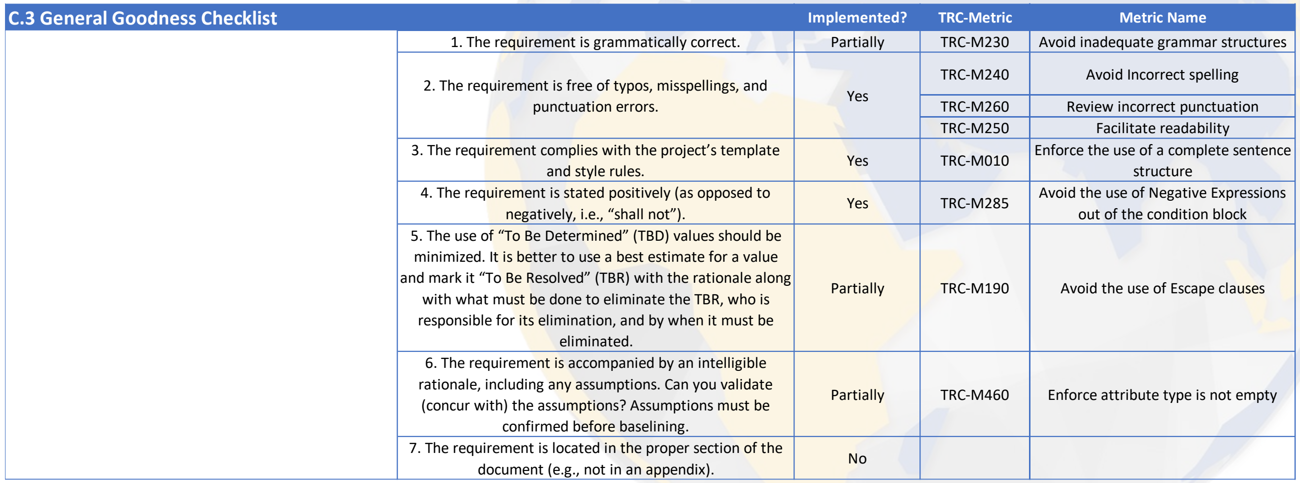
Mapping TRC metrics vs. Requirements Checklist



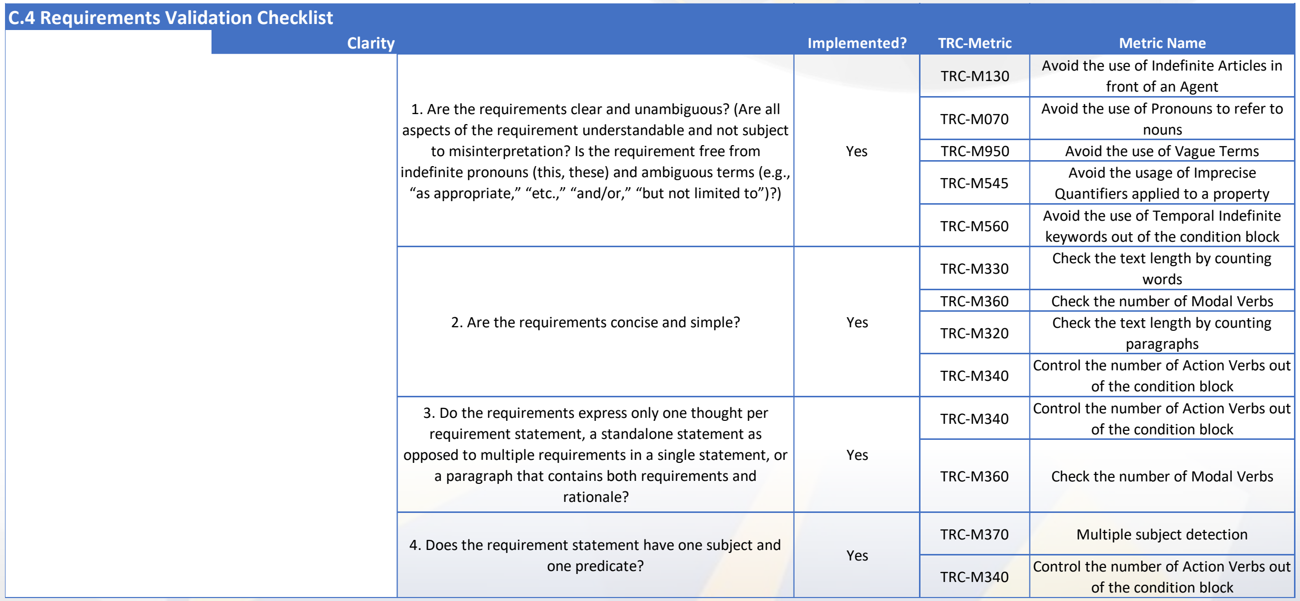
C.1 Use of Correct Terms		Implemented?	TRC-Metric	Metric Name

	Shall = requirement
	Will = facts or declaration of purpose
	Should = goal
		Yes	TRC-M365	Avoid the use of Banned Modal Verbs

C.2 Editorial Checklist				
Personnel Requirement		Implemented?	TRC-Metric	Metric Name
	Use the active, rather than the passive voice. A requirement should state who shall (do, perform, provide, weigh, or other verb) followed by a description of what should be performed.	Yes	TRC-M040	Avoid the use of Passive Voice out of the condition block
Product Requirement		Implemented?	TRC-Metric	Metric Name
	The requirement is in the form "product ABC shall XYZ." A requirement should state "The product shall" (do, perform, provide, weigh, or other verb) followed by a description of what should be done.	Yes	TRC-M010	Enforce the use of a complete sentence structure
			TRC-M360	Check the number of Modal Verbs
	The requirement uses consistent terminology to refer to the product and its lower-level entities.	Yes	TRC-M050	Determine if the subject is a recognized Agent term
			TRC-M630	Enforce the use of Define Terms by avoiding Synonyms
			TRC-M150	Detect inadequate Unit for a Characteristic
	Complete with tolerances for qualitative/performance values (e.g., less than, greater than or equal to, plus or minus, 3 sigma root sum squares).	Yes	TRC-M520	Force to include tolerance value for the units that required tolerance
			TRC-M525	Ensure tolerance value are within an adequate value range
	Is the requirement free of implementation? (Requirements should state WHAT is needed, NOT HOW to provide it; i.e., state the problem not the solution. Ask, "Why do you need the requirement?" The answer may point to the real requirement.)	Yes	TRC-M490	Avoid stating a solution
	Free of descriptions of operations? (Is this a need the product should satisfy or an activity involving the product? Sentences like "The operator shall…" are almost always operational statements not requirements.)	Partially	TRC-M500	Avoid the use of Flow sentences

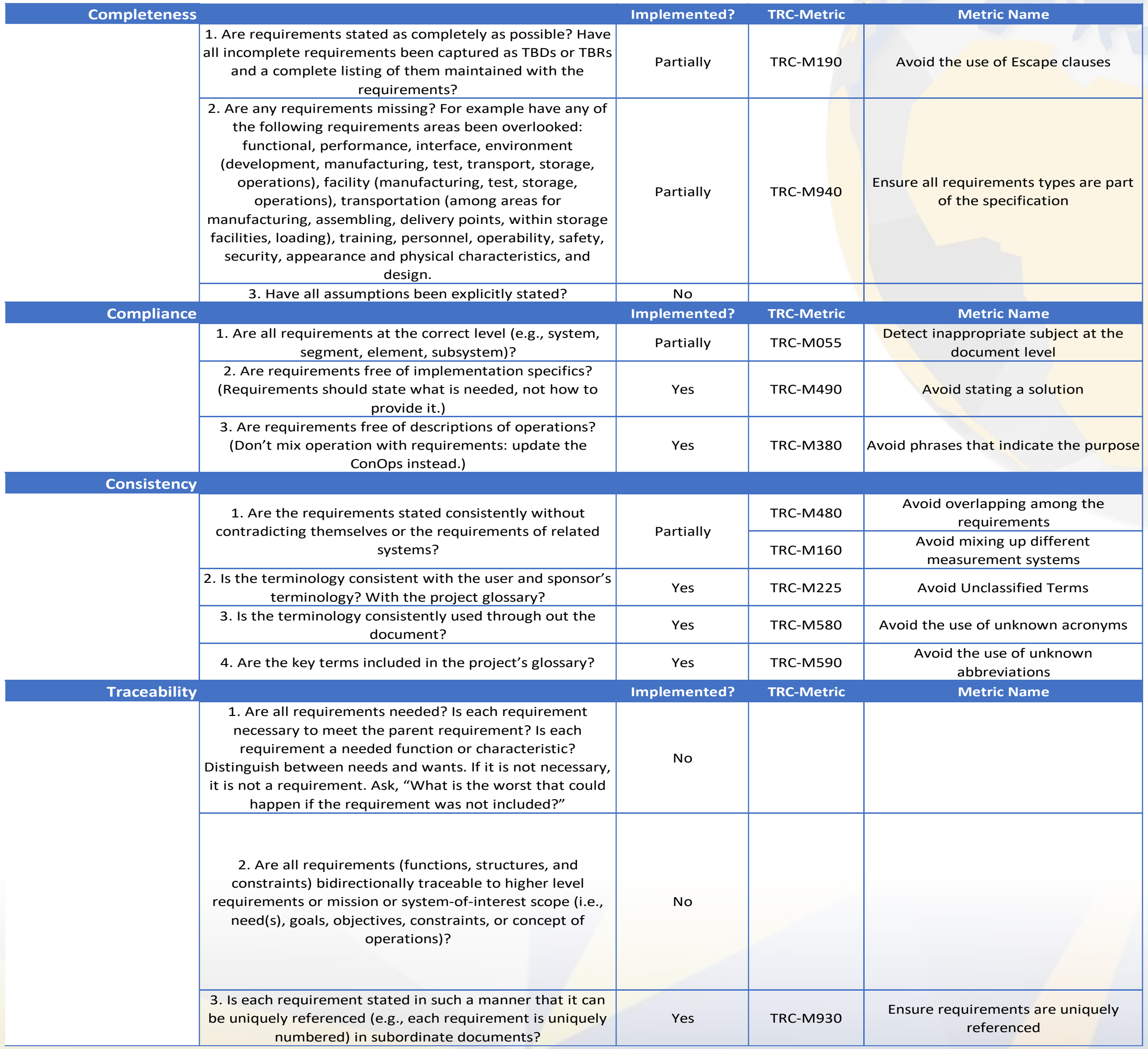


C.3 General Goodness Checklist		Implemented?	TRC-Metric	Metric Name
	1. The requirement is grammatically correct.	Partially	TRC-M230	Avoid inadequate grammar structures
	2. The requirement is free of typos, misspellings, and punctuation errors.	Yes	TRC-M240	Avoid Incorrect spelling
			TRC-M260	Review incorrect punctuation
			TRC-M250	Facilitate readability
	3. The requirement complies with the project's template and style rules.	Yes	TRC-M010	Enforce the use of a complete sentence structure
	4. The requirement is stated positively (as opposed to negatively, i.e., "shall not").	Yes	TRC-M285	Avoid the use of Negative Expressions out of the condition block
	5. The use of "To Be Determined" (TBD) values should be minimized. It is better to use a best estimate for a value and mark it "To Be Resolved" (TBR) with the rationale along with what must be done to eliminate the TBR, who is responsible for its elimination, and by when it must be eliminated.	Partially	TRC-M190	Avoid the use of Escape clauses
	6. The requirement is accompanied by an intelligible rationale, including any assumptions. Can you validate (concur with) the assumptions? Assumptions must be confirmed before baselining.	Partially	TRC-M460	Enforce attribute type is not empty
	7. The requirement is located in the proper section of the document (e.g., not in an appendix).	No		



C.4 Requirements Validation Checklist				
Clarity		Implemented?	TRC-Metric	Metric Name
	1. Are the requirements clear and unambiguous? (Are all aspects of the requirement understandable and not subject to misinterpretation? Is the requirement free from indefinite pronouns (this, these) and ambiguous terms (e.g., "as appropriate," "etc.," "and/or," "but not limited to")?)	Yes	TRC-M130	Avoid the use of Indefinite Articles in front of an Agent
			TRC-M070	Avoid the use of Pronouns to refer to nouns
			TRC-M950	Avoid the use of Vague Terms
			TRC-M545	Avoid the usage of Imprecise Quantifiers applied to a property
			TRC-M560	Avoid the use of Temporal Indefinite keywords out of the condition block
	2. Are the requirements concise and simple?	Yes	TRC-M330	Check the text length by counting words
			TRC-M360	Check the number of Modal Verbs
			TRC-M320	Check the text length by counting paragraphs
			TRC-M340	Control the number of Action Verbs out of the condition block
	3. Do the requirements express only one thought per requirement statement, a standalone statement as opposed to multiple requirements in a single statement, or a paragraph that contains both requirements and rationale?	Yes	TRC-M340	Control the number of Action Verbs out of the condition block
			TRC-M360	Check the number of Modal Verbs
	4. Does the requirement statement have one subject and one predicate?	Yes	TRC-M370	Multiple subject detection
			TRC-M340	Control the number of Action Verbs out of the condition block

Mapping TRC metrics vs. Requirements Checklist



Completeness		Implemented?	TRC-Metric	Metric Name
	1. Are requirements stated as completely as possible? Have all incomplete requirements been captured as TBDs or TBRs and a complete listing of them maintained with the requirements?	Partially	TRC-M190	Avoid the use of Escape clauses
	2. Are any requirements missing? For example have any of the following requirements areas been overlooked: functional, performance, interface, environment (development, manufacturing, test, transport, storage, operations), facility (manufacturing, test, storage, operations), transportation (among areas for manufacturing, assembling, delivery points, within storage facilities, loading), training, personnel, operability, safety, security, appearance and physical characteristics, and design.	Partially	TRC-M940	Ensure all requirements types are part of the specification
	3. Have all assumptions been explicitly stated?	No		
Compliance		**Implemented?**	**TRC-Metric**	**Metric Name**
	1. Are all requirements at the correct level (e.g., system, segment, element, subsystem)?	Partially	TRC-M055	Detect inappropriate subject at the document level
	2. Are requirements free of implementation specifics? (Requirements should state what is needed, not how to provide it.)	Yes	TRC-M490	Avoid stating a solution
	3. Are requirements free of descriptions of operations? (Don’t mix operation with requirements: update the ConOps instead.)	Yes	TRC-M380	Avoid phrases that indicate the purpose
Consistency				
	1. Are the requirements stated consistently without contradicting themselves or the requirements of related systems?	Partially	TRC-M480	Avoid overlapping among the requirements
			TRC-M160	Avoid mixing up different measurement systems
	2. Is the terminology consistent with the user and sponsor’s terminology? With the project glossary?	Yes	TRC-M225	Avoid Unclassified Terms
	3. Is the terminology consistently used through out the document?	Yes	TRC-M580	Avoid the use of unknown acronyms
	4. Are the key terms included in the project’s glossary?	Yes	TRC-M590	Avoid the use of unknown abbreviations
Traceability		**Implemented?**	**TRC-Metric**	**Metric Name**
	1. Are all requirements needed? Is each requirement necessary to meet the parent requirement? Is each requirement a needed function or characteristic? Distinguish between needs and wants. If it is not necessary, it is not a requirement. Ask, “What is the worst that could happen if the requirement was not included?”	No		
	2. Are all requirements (functions, structures, and constraints) bidirectionally traceable to higher level requirements or mission or system-of-interest scope (i.e., need(s), goals, objectives, constraints, or concept of operations)?	No		
	3. Is each requirement stated in such a manner that it can be uniquely referenced (e.g., each requirement is uniquely numbered) in subordinate documents?	Yes	TRC-M930	Ensure requirements are uniquely referenced

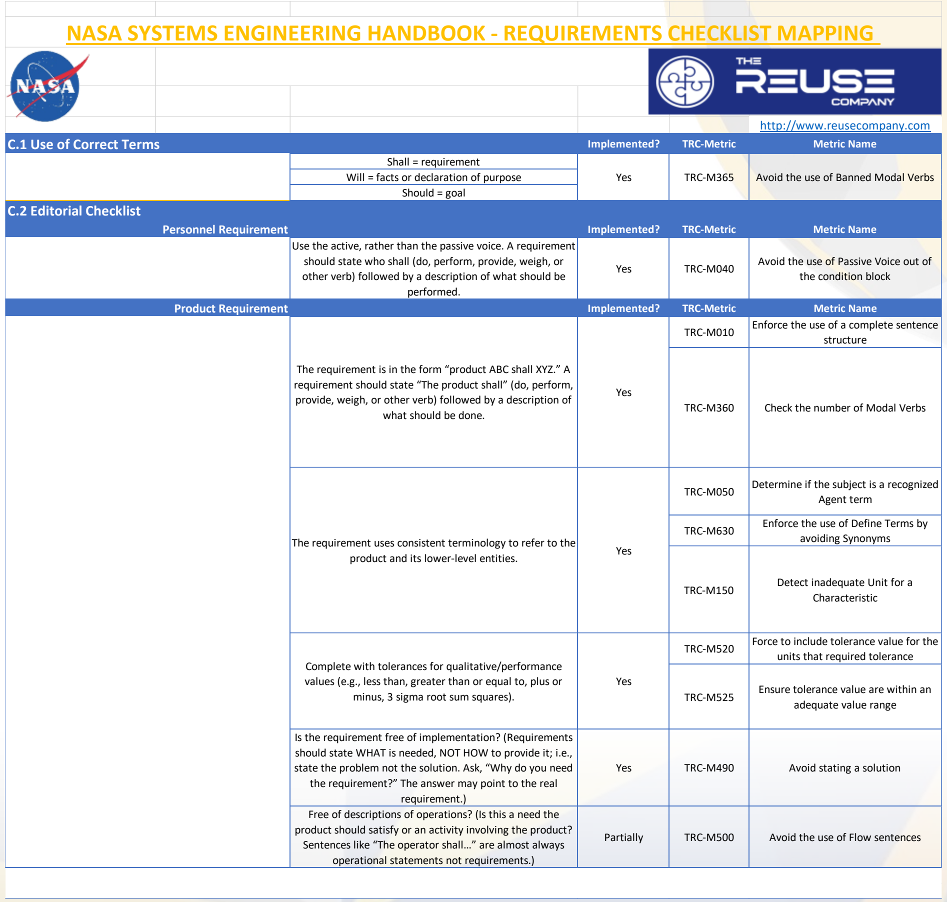
Mapping TRC metrics vs. Requirements Checklist

Correctness		Implemented?	TRC-Metric	Metric Name
	1. Is each requirement correct? 2. Is each stated assumption correct? Assumptions must be confirmed before the document can be baselined. 3. Are the requirements technically feasible?	Yes	All TRC correctness metrics*	*The whole set of correctness metrics help in the correctness quality check
Functionality		**Implemented?**	**TRC-Metric**	**Metric Name**
	1. Are all described functions necessary and together sufficient to meet mission and system goals and objectives?	No		
Performance		**Implemented?**	**TRC-Metric**	**Metric Name**
	1. Are all required performance specifications and margins listed (e.g., consider timing, throughput, storage size, latency, accuracy and precision)?	Partially	TRC-M545	Avoid the usage of Imprecise Quantifiers applied to a property
			TRC-M530	Confirms the value for a property is within a controlled range
	2. Is each performance requirement realistic?	Yes	TRC-M430	Avoid unachievable Absolutes expressions impossible to verify
			TRC-M140	Ensure Numbers are followed by Units or noun qualifications
	3. Are the tolerances overly tight? Are the tolerances defendable and cost-effective? Ask, “What is the worst thing that could happen if the tolerance was doubled or tripled?”	Yes	TRC-M525	Ensure tolerance value are within an adequate value range
Interfaces		**Implemented?**	**TRC-Metric**	**Metric Name**
	1. Are all external interfaces clearly defined?	Partially	TRC-M945	SCM PBS completeness
	2. Are all internal interfaces clearly defined?		TRC-M945	SCM PBS completeness
	3. Are all interfaces necessary, sufficient, and consistent with each other?		TRC-M945	SCM PBS completeness
Maintainability		**Implemented?**	**TRC-Metric**	**Metric Name**
	1. Have the requirements for system maintainability been specified in a measurable, verifiable manner?	Partially	TRC-M140	Ensure Numbers are followed by Units or noun qualifications
			TRC-M540	Avoid the usage of Imprecise Quantifiers
			TRC-M430	Avoid unachievable Absolutes expressions impossible to verify
			TRC-M940	SCM organization completeness
	2. Are requirements written so that ripple effects from changes are minimized (i.e., requirements are as weakly coupled as possible)?	Partially	TRC-M200	Avoid the use of Open-Ended clauses

Reliability		Implemented?	TRC-Metric	Metric Name
	1. Are clearly defined, measurable, and verifiable reliability requirements specified?	Partially	TRC-M540	Avoid the usage of Imprecise Quantifiers
			TRC-M430	Avoid unachievable Absolutes expressions impossible to verify
	2. Are there error detection, reporting, handling, and recovery requirements?	Yes	TRC-M940	SCM organization completeness
	3. Are undesired events (e.g., single event upset, data loss or scrambling, operator error) considered and their required responses specified?	No		
	4. Have assumptions about the intended sequence of functions been stated? Are these sequences required?	No		
	5. Do these requirements adequately address the survivability after a software or hardware fault of the system from the point of view of hardware, software, operations, personnel and procedures?	No		
Verifiability/Testability		**Implemented?**	**TRC-Metric**	**Metric Name**
	1. Can the system be tested, demonstrated, inspected, or analyzed to show that it satisfies requirements? Can this be done at the level of the system at which the requirement is stated? Does a means exist to measure the accomplishment	Partially	TRC-M540	Avoid the usage of Imprecise Quantifiers
			TRC-M430	Avoid unachievable Absolutes expressions impossible to verify
	2. Are the requirements stated precisely to facilitate specification of system test success criteria and requirements?	Yes	TRC-M940	SCM organization completeness
	3. Are the requirements free of unverifiable terms (e.g., flexible, easy, sufficient, safe, ad hoc, adequate, accommodate, user-friendly, usable, when required, if required, appropriate, fast, portable, light-weight, small	Yes	TRC-M430	Avoid unachievable Absolutes expressions impossible to verify
			TRC-M950	Avoid the use of Vague Terms
Data Usage		**Implemented?**	**TRC-Metric**	**Metric Name**
	1. Where applicable, are “don’t care” conditions truly “don’t care”? (“Don’t care” values identify cases when the value of a condition or flag is irrelevant, even though the value may be important for other cases.) Are “don’t care” conditions values explicitly stated? (Correct identification of “don’t care” values may improve a design’s portability.)	No		

Mapping TRC metrics vs. Requirements Checklist

- PDF version : available on The REUSE Company's website !



NASA SYSTEMS ENGINEERING HANDBOOK - REQUIREMENTS CHECKLIST MAPPING

http://www.reusecompany.com

C.1 Use of Correct Terms		Implemented?	TRC-Metric	Metric Name

	Shall = requirement
	Will = facts or declaration of purpose
	Should = goal
		Yes	TRC-M365	Avoid the use of Banned Modal Verbs

C.2 Editorial Checklist				
Personnel Requirement		**Implemented?**	**TRC-Metric**	**Metric Name**
	Use the active, rather than the passive voice. A requirement should state who shall (do, perform, provide, weigh, or other verb) followed by a description of what should be performed.	Yes	TRC-M040	Avoid the use of Passive Voice out of the condition block
Product Requirement		**Implemented?**	**TRC-Metric**	**Metric Name**
	The requirement is in the form "product ABC shall XYZ." A requirement should state "The product shall" (do, perform, provide, weigh, or other verb) followed by a description of what should be done.	Yes	TRC-M010	Enforce the use of a complete sentence structure
			TRC-M360	Check the number of Modal Verbs
	The requirement uses consistent terminology to refer to the product and its lower-level entities.	Yes	TRC-M050	Determine if the subject is a recognized Agent term
			TRC-M630	Enforce the use of Define Terms by avoiding Synonyms
			TRC-M150	Detect inadequate Unit for a Characteristic
	Complete with tolerances for qualitative/performance values (e.g., less than, greater than or equal to, plus or minus, 3 sigma root sum squares).	Yes	TRC-M520	Force to include tolerance value for the units that required tolerance
			TRC-M525	Ensure tolerance value are within an adequate value range
	Is the requirement free of implementation? (Requirements should state WHAT is needed, NOT HOW to provide it; i.e., state the problem not the solution. Ask, "Why do you need the requirement?" The answer may point to the real requirement.)	Yes	TRC-M490	Avoid stating a solution
	Free of descriptions of operations? (Is this a need the product should satisfy or an activity involving the product? Sentences like "The operator shall…" are almost always operational statements not requirements.)	Partially	TRC-M500	Avoid the use of Flow sentences

THE REUSE COMPANY