PROJECT PERFORMANCE INTERNATIONAL

helping projects succeed…

www.ppi-int.com

DATA ITEM DESCRIPTION

1. TITLE	2. IDENTIFICATION NUMBER
REQUIREMENTS TRACEABILITY REPORT IN SYSTEM DESIGN (RTR-SD)	PPI-005696-6 14 October 2020

3. DESCRIPTION

The Requirements Traceability Report in System Design (RTR-SD) provides information on requirements flow-down from one physical level of problem definition to requirements on elements of solution, one physical level below the problem level.

For example, the RTR-SD may depict traceability from a set of capability system requirements to sets of requirements on end-use technology items, support elements and other solution elements implementing that capability.

In product development, the RTR-SD may depict traceability from a set of requirements defined in a product development specification to sets of requirements on subsystems comprising solution elements for the product. Solution elements may be hardware or software or even materials, such as lubricants.

In subsystem development, the RTR-SD may depict traceability from a set of requirements on the subsystem to sets of requirements on lower-level subsystems comprising solution elements for the parent subsystem.

Traceability is bi-directional between requirements at adjacent hierarchical physical levels having a problem-solution relationship, each linkage being between a pair of requirements each at its physical level.

Although requirements traceability in design linkages are always between requirements at adjacent hierarchical physical levels, the RTR-SD may be extended to incorporate requirements traceability from Physical Level 1 to Physical Level 2 to Physical Level 3, etc.

4. PURPOSE

The RTR-SD provides a means of confirming that all system requirements have been actioned, to see where they have been actioned, or if they haven't been actioned, to see why not. In the reverse direction, the RTR-SD helps ensure that are no orphan requirements (requirements without parents) – that all child requirements can be traced back to satisfaction of one or more parent system requirements.

5. PREPARATION GUIDELINES

5.1 General Instructions

a. **Automated techniques.** Use of automated techniques is encouraged. The term "document" in this DID means a collection of data regardless of its medium.

continued next page

6. SOURCE

5. PREPARATION GUIDELINES *continued*

5.2 Content Requirements

Content requirements and guidance begin on the page 3. The paragraph numbers shown are for convenience of presentation to the reader, and may bear no relationship to the organization of information in the RTR-SD. Each such number is understood here to have the prefix "5.2" within this DID. For example, the paragraph numbered 1.1 is understood to be paragraph 5.2.1.1 within this DID.

5.3 Foreword

Requirements traceability in design is of considerable importance in ensuring that the solution fully addresses the specified requirements and goals, no more and no less. Ultimately, this form of requirements traceability helps ensure that all significant development activity is responsive to requirements (which if valid, will reflect the needs of the stakeholders).

TABLE OF CONTENTS

1. INTRODUCTION AND SCOPE

1.1 Document Overview and Use

This DID provides requirements together with guidance for the content of a requirements database implementing requirements traceability in the system design and development application, including capability development. This DID also provides requirements and guidance for the content of a Requirements Traceability Report in System Design (RTR-SD). This DID contrasts with DID PPI-005695: “Requirements Traceability Report in Requirements Analysis (RTR-RA)”, which deals with the requirements analysis application of requirements traceability.

The format of the RTR-SD report is not specified, since the origin of such a report may be from a requirements management software tool that provides tool-specific report formats only.

This DID also lists other potential content of requirements traceability (in design) reporting.

2. APPLICABLE AND OTHER REFERENCED DOCUMENTS

2.1 Applicable Documents

There are no applicable documents.

2.2 Other Referenced Documents

PPI-005695: “*Requirements Traceability Report in Requirements Analysis (RTR-RA)*”.

3. DEFINITIONS, ACRONYMS AND ABBREVIATIONS

3.1 Definitions

The following definitions shall apply in the interpretation of requirements in this document:

Capability System means, in the context of this document, the system comprising the interacting set of relevant technology, people, process and other elements that satisfies, or is intended to satisfy, on a whole-of-life basis, the needs of one or more humans or organizations of humans by means of satisfaction of requirements specified in a Capability System Requirements Specification (CapSyRS).

May expresses permissive guidance.

Non-Requirement means an entity designated by the originator as a requirement (originating requirement) but classified by somebody with the authority to do so as an entity not recognized, nor to be acted upon, as requirement.

Possessed Characteristic means a characteristic possessed by a non-developmental item that is used as a solution element, where that characteristic is not used in the solution. The possessed characteristic does not, therefore, have a parent requirement. Possessed characteristics are sometimes tracked because of the potential for future use, amongst other reasons.

Requirement, in the context of this document, means a characteristic that the item that is the subject of the requirement is required to possess.

Shall expresses a requirement.

Should expresses a target or goal to be pursued, but not necessarily achieved.

Verification Requirement, in the context of this document, means a requirement specifying the quality or strength of evidence that a verification activity is to provide, the evidence being as to whether or not the system requirement to which the verification requirement corresponds has been satisfied.

Will expresses a declaration of intent by a party, usually the sponsoring or acquiring organization. "Will" does not express a requirement. "Will" may also be used in cases where the simple future tense is needed, for example, "The operating system will be supplied by the client".

3.2 Acronyms

This section should list alphabetically each acronym used in the document, together with the acronym's expanded meaning.

The acronyms used in this document shall be interpreted as follows:

CC	Creative Commons
CIV	Compromise Impact Value
DID	Data Item Description
RA	Requirements Analysis
RMT	Requirements Management Tool
RTR	Requirements Traceability Report
RTR-RA	Requirements Traceability Report in Requirements Analysis
RTR-SD	Requirements Traceability Report in System Design
SD	System Design

3.3 Abbreviations

This section should list alphabetically each abbreviation used in the document, together with the abbreviation's expanded meaning, except that abbreviations within the International System of Units (SI) should not be listed.

The abbreviations used in this document shall be interpreted as follows:

CapSyRS	Capability System Requirements Specification
I/S	Infrastructure
Info	Information
REQID	Requirement Identifier
SI	International System of Units
SyRS	System Requirements Specification
VREQID	Verification Requirement Identifier

4. BACKGROUND TO REQUIREMENTS TRACEABILITY

4.1 Description

The RTR-SD provides information on requirements flow-down from one physical level of problem definition to requirements on elements of solution at the next physical level down from the problem level, for example, a set of capability system requirements to sets of requirements on technology items, support elements and other elements implementing that capability.

In product development, the RTR-SD may depict traceability from a set of requirements defined in a product development specification (that can be regarded as the top physical level, Physical Level 1) to sets of requirements on subsystems comprising solution elements for the product (that can be regarded as Physical Level 2). Solution elements may be hardware, software or materials.

In subsystem development, the RTR-SD may depict traceability from a set of requirements on the subsystem to sets of requirements on lower-level subsystems comprising solution elements for the parent subsystem.

Traceability is bi-directional between requirements at adjacent hierarchical physical levels having a problem-solution relationship. Traceability can also be implemented goal (parent) to goal (child) and goal (parent) to requirement (child). A requirement (parent) to goal (child) traceability relationship is also possible, although less likely. It is an engineering management decision as to whether to extend traceability to goals.

The RTR-SD may be extended to incorporate requirements traceability from Level 1 to Level 2 to Level 3, etc., without limit, that is, through multiple physical levels, but never skipping a level.

These concepts and their relationships to requirements traceability in Requirements Analysis are illustrated in Figure 4.1-1, using an enterprise system as an example Level 1 system.

Solution elements may be tangible elements such as physical subsystems, intangible elements such as software or databases, or virtual elements such as interfaces between system elements, for example: hardware/hardware, hardware/software and software/software interfaces.

The RTR-SD may be used in a Product Line environment, in which requirements driving design are partitioned into a common subset for all members of the product line, together with a unique subset of requirements for each member of the product line.

The RTR-SD may also be used where requirements driving design are invoked by reference, for example, by invoking standards.

Although the parent-to-child/child-to-parent design relationship is the main focus of the RTR-SD, other relationships, including peer relationships within a set of requirements on an item, may be recorded and reported. Examples of peer relationships are "cross-references/is cross-referenced by", and "supersedes/is superseded by". A requirement may also trace to:

a. a Note to the requirement;
b. a global definition (applies throughout the requirements set) of a term used in the requirement;
c. a local definition of a term (applies only to the use of the term in the requirement).

4.2 Purpose of Requirements Traceability Report in SD

The RTR-SD provides primarily a means to confirm that all system requirements have been actioned in design, or if they haven't been, why not. In the reverse direction, the report helps ensure that there are no orphan requirements - that all child requirements can be traced back to satisfaction of one or more parent requirements, and ultimately, to the needs of the enterprise.

The RTR-SD has many other purposes, including:

a. facilitation of review of design of proposed solution at a detailed level (detailed design review), to identify errors and omissions;
b. efficient conduct of analysis of the potential span of impact of a proposed system requirement change;
c. reduction in the amount of work needed to respond to changing system requirements;
d. where a technology item forms a part of more than one system, reduction in the likelihood of change initiated for one use inadvertently rendering the technology item unable to satisfy another use;
e. reduction in the amount of work needed to correct latent design defects in system solutions;
f. reduction in the amount of work needed to deal with technology item obsolescence;
g. increased potential for economies through the creation of visible, reusable requirements sets;
h. monitoring and communication of verification status; and
i. linkage to verification traceability, through verification requirements, verification procedures, verification events carried out on verification articles, verification results, and verification records.

These are substantial benefits!

4.3 Applicability

The RTR-SD applies to requirements (normally "shall" statements) and goals (normally "should" statements), as well as to non-requirements. The RTR-SD does not apply to permissive guidance – "may" statements, and declarations of intent or futurity – "will" statements. Nor does the RTR-SD apply to headings for one or a group of requirements.

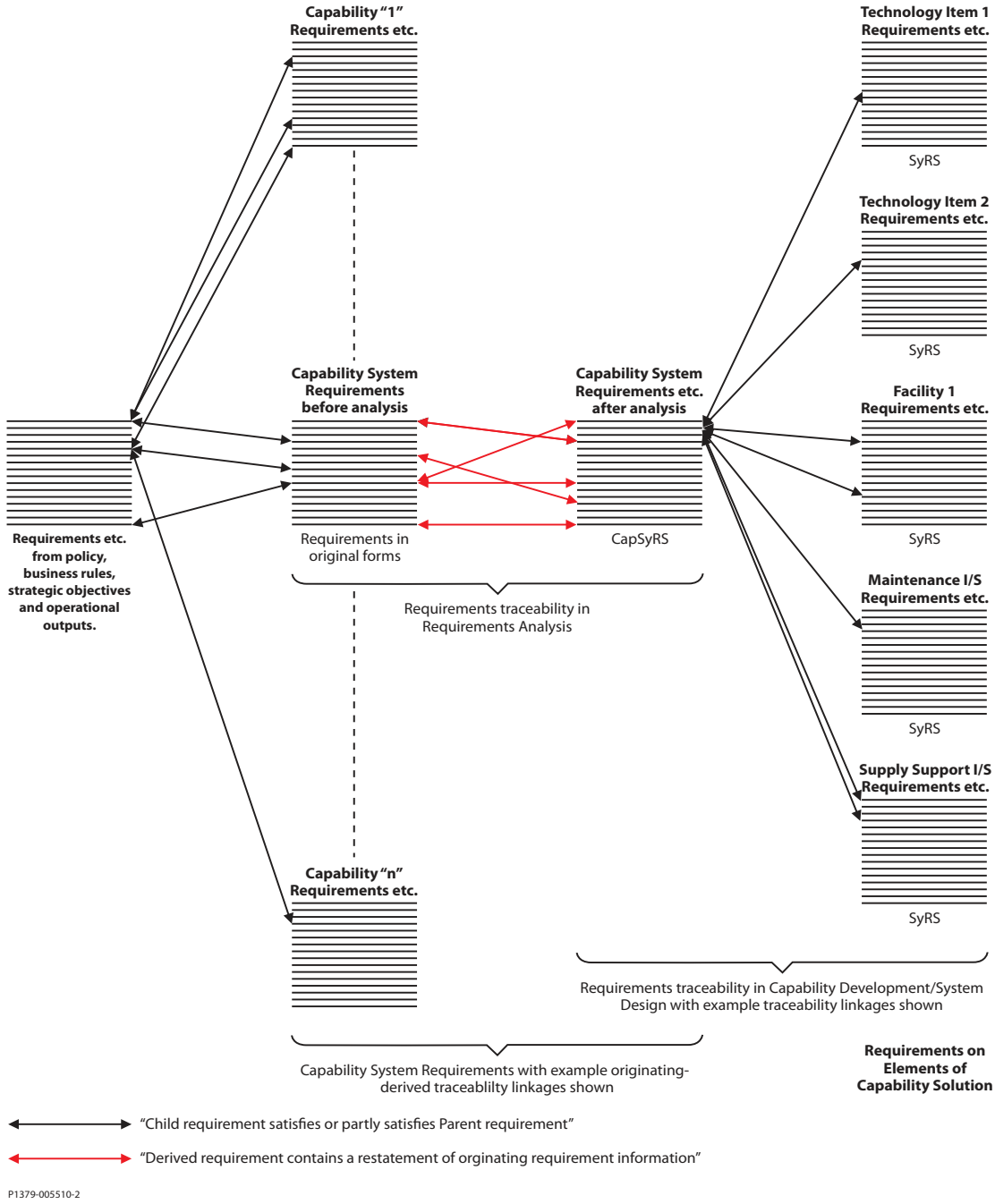

Figure 4.1-1 The Concepts of Requirements Traceability in Requirements Analysis and in Capability Development/System Design – Example

4.4 System (Capability, Software, Technology Item) Requirements Attributes in a Database

4.4.1 Minimum Requirements and Goals Attributes for SD Requirements Traceability

In order for a requirement to be uniquely identifiable and therefore traceable, the requirement is to have the following minimum attributes in a requirements database:

a. REQID - The requirement identifier (REQID) **uniquely** identifies the specific requirement within a set of requirements. This number could be project-specific or program-specific, and may have been generated automatically by a requirements management software tool in use. The REQID may incorporate identification of the item that is the subject of the requirement.

b. Subject - This is the item that is the subject of the requirement (the Actor). Every requirement must have an Actor, unless it is in a library of reusable requirements.

c. Requirement Statement - This is the expression of the requirement in some suitable language and may include additional information by reference.
d. Ownership - This is the identification of the organization and/or person who, with appropriate authority, created or has a right to change the requirement. A requirement must have at least one owner, and may have more than one owner.
e. Rationale - This attribute field must be present, however, the field should be populated selectively. The rationale for the requirement in system solution decisions can be one of the most important attributes. For child requirements, rationale provides the solution thinking that created the requirement. It may be the only source of corporate memory when requirements changes take place at a later stage in a project. The rationale record ensures that the original insight and thought process are understood before changes to the requirement are made.

Where "requirement" is referred to in this paragraph without any corresponding reference to goal, the reference applies to both requirements and goals.

4.4.2 Optional Requirements Attributes for Requirements Traceability in SD

Each requirement may have assigned in a requirements database a selection of the following additional attributes. The importance of each of these candidate attributes varies greatly, both in general and in specific implementations of requirements management (the purpose of an attribute may be better achieved in some other way). The candidate additional attributes are:

a. Importance - One of Critical, Very Important, Important, Modest Importance, Low Importance, or an alternative scale of importance. Importance may be expressed in terms of Compromise Impact Value (CIV), with a number range 10 (most important) to 1 (least important), the CIV value relating to the magnitude of damage done or loss incurred if the requirement were not met.

b. Status:
   i. For a requirement: "Incomplete Requirement" (work-in-progress), "Ready for Review", "In Review", "Baselined Requirement", "Superseded Requirement", "Possessed Characteristic" or "Non-Requirement". "Possessed Characteristic" should be used alongside child requirements. "Non-Requirement" should be used only for parent requirements.
   ii. For a goal: "Incomplete Goal" (work-in-progress), "Baselined Goal", "Superseded Goal", or "Non-Goal". "Non-Goal" should be used only for parent goals.

c. Included Note - The text (if any) of any note that is to accompany the requirement in its database form and its form in a requirements specification document. Notes may alternatively be treated as unique objects and linked to requirements. This latter approach allows for reuse of notes.

d. Comments - This attribute is used for ad-hoc comments and notes to the requirement (if any) not intended to be reproduced with the requirement in a requirements specification.

e. Corresponding Verification Requirement (if any) - For a requirement, this attribute identifies and links to the corresponding verification requirement relating to system verification (not design verification). For a goal, this attribute identifies and links to the corresponding verification requirement relating to determination as to whether, or to what degree, the goal has been satisfied. Verification requirements may alternatively be treated as unique objects and linked to corresponding system requirements. This latter approach allows for reuse of verification requirements

f. Verification Status - One of Not Used/Not Yet Verified/Passed/Failed/Passed After Re-verification/Not Applicable.

   "Not Used" means that the requirements database is not being used to record verification status. It does not necessarily mean that the satisfaction of the requirement will not be verified. "Not Yet Verified" means that verification is planned but has not yet been conducted or completed. "Not Applicable" means that there is no intent to conduct verification of satisfaction of the requirement.

g. Date on which the Requirement was first entered.

h. Approval Date: The date on which the requirement was approved in its current version by the requirement owner(s).

i. If the RMT is configured to version requirements under the same REQID, date of the last change to the requirement.

j. Version - For larger developments for more critical systems, each requirement identified by its REQID may itself be versioned, allowing the evolution of a requirement through system development to be traced. In this case, earlier versions will have the status “Superseded Requirement”. Alternatively, a requirement that is the subject of a change may be superseded and replaced by a new requirement with a new REQID, with linkage to its predecessor.

k. Name - The name of the requirement is a short unique descriptive name that can be used to refer to the requirement in addition to its REQID. Uniqueness of naming may or may not be enforced.

l. Source Reference - A reference or link into system development (solution decision-making) records.

m. Type - The primary type of requirement can be *one* of the following:
   i. State/Mode - states the required states and/or modes of the item, or the required transition between one state and another state, one mode and another mode, mode in one state to mode in another state. May state the response required as a direct consequence of a transition having occurred. A “state” is a condition of something. A “mode” is functionality related to a significant aspect of use, usually a group of functionality.
   ii. Functional - states what the item is to *do*. Functional alone is, at least technically, incomplete.
   iii. Performance - for a given function, states *how well* that function is to be performed by the item, that is, performance is an attribute of function. Performance alone is incomplete.
   iv. Functional and Performance – as defined for Functional and Performance Individually.
   v. External Interface - states the required characteristics at a localized point, or region, of connection of the item to the outside world (e.g., location of interface, geometry of connection, inputs and outputs by name and specification, allocation of signals to pins, etc.).
   vi. Environmental - limits the effect that the external enveloping environment (natural or induced) is to have on the item, and the effect that the item is to have on the external enveloping environment.
   vii. Resource - limits the usage or consumption by the item of an *externally* provided resource or requires the use of an *externally* provided resource.
   viii. Physical - states a required *physical characteristic* (property of matter) of the item as a whole (e.g., mass, dimension, volume).
   ix. Other Quality - states any other required quality of the item that is not one of the above defined types, nor is it a design requirement.
   x. Design - directs the design (internals of the item), by inclusion (build it internally this way), or exclusion (don’t build it internally this way). Any requirement on a subsystem, component, part, material, etc. in a requirement set for a parent system is, by definition, a Design Requirement.
   xi. Compound - a requirement statement that expresses more than one requirement in the one expression, e.g. sentence.

   Note: The same types apply also to goals (also termed design goals, targets, objectives).

n. Name of Requirement Enterer – the person making the entry or import of the requirement into the RMT. This person may or may not be the requirement owner.

o. Stakeholders, Other – Those stakeholders who are not the owner(s) or enterer of the requirement, but who should be informed of any proposed or actual change to the requirement or non-compliance.

p. Stability, Expected – some measure of the likelihood or otherwise that the requirement will change.

q. Responsibility for Implementation.

r. Status of Implementation in Design.

s. Priority – The preferred sequence of implementation.

t. Quality Metric Value – The quality value of a requirement on a suitable scale, reflecting absence of defects such as ambiguity, factual incorrectness, lack of verifiability, etc.

u. Risk Index - Implementation – Risk broadly is the product of the Importance of a Requirement multiplied by the probability of the requirement not being satisfied, expressed in some suitable measure or risk.

v. Risk Index - Requirement Defect – Risk is broadly the product of the Importance of the requirement multiplied by the probability of loss due to defects in the requirement, expressed in some suitable measure of risk.

w. Architectural Design Driver – This is a requirement that is expected will significantly influence the concept of the design of the system/product. Usually a YES/NO value.

x. Issues - This attribute field can be used to record relevant information not addressed by other attributes.

y. Operational Requirement – This is a requirement that serves an end-use purpose. Usually a YES/NO value.

z. Regulatory – This is a requirement that has its origin in Regulation. Usually has a YES/NO value.

aa. Legal – This is a requirement that has its origins in statute law. Usually a YES/NO value.

bb. Build Allocation – The build, release or increment to which the requirement is allocated for implementation.

cc. Product Line – Identification of the Product Line to which the product that is the subject of requirements belongs.

Where "requirement" is referred to in this paragraph without any corresponding reference to "goal", the reference applies to both requirements and goals.

4.5 Verification Requirements Attributes for Requirements Traceability in SD

4.5.1 Minimum Verification Requirements Attributes for Requirements Traceability in SD

Where verification requirements are included in the requirements database, the following attributes are to be recorded for each verification requirement:

a. VREQID - The verification requirements identifier (VREQID) **uniquely** identifies the specific verification requirement within a set of verification requirements. This identifier could be project-specific or program-specific, and may have been generated automatically by a requirements management software tool.
b. Verification Requirement Statement - This is the expression of the verification requirement, which is a statement on the qualities or strength of the evidence required that a system requirement has been satisfied: the characteristics required of any verification solution. A Verification Requirement Statement may comprise or include a statement defining a directed verification method (one or more of Analogy, Analysis, Certification, Demonstration, Examination/Inspection, Simulation, Test), but verification requirements are not (or should not be) just a list of verification methods.

In addition, the system (software, etc) requirement(s) to which each verification corresponds is/are to be identifiable, unless the requirement is a member of a pool of reusable verification requirements.

4.5.2 Optional Verification Requirements Attributes for Requirements Traceability in SD

Where verification requirements are included in the requirements database, the following attributes may be recorded for each verification requirement:

a. Name - The name of the verification requirement is a short descriptive name that can be used in addition to the VREQID to refer to the verification requirement. Such a name will often reflect the subject matter of the system or software (etc.) requirement to which the verification requirement relates. An alternative is to use the Name attribute (if used) of the corresponding system or software (etc.) requirement.
b. Ownership - This is the identification of the organization and/or person who, with appropriate authority, created or has a right to change the verification requirement. A verification requirement must have at least one owner and may have more than one owner.
c. Date the verification requirement was first entered.
d. Status - "Incomplete Verification Requirement" (work-in-progress), "Ready for Review", "In Review", "Baselined Verification Requirement", "Superseded Verification Requirement"
e. Approval Date: The date on which the verification requirement was approved in its current version by or on behalf of the verification requirement owner(s).
f. If the RMT is configured to version verification requirements under the same VREQID, date of the last change to the requirement.

4.6 Requirements Traceability Factors to Consider for Requirements Traceability in SD

Requirements traceability in SD has to provide the evidence of parent-child associations between a set of requirements on a system and a set of solution-specific requirements on solution elements, created by design of the system and subsequently communicated in a set of requirements specifications or requirements databases. Due to the number and complexity of these relationships, requirements traceability for SD is typically implemented with computer-based requirements management software tools.

There is also a further temporal aspect to requirements traceability, viz. that the requirements themselves and the traceability between requirements change over time, necessitating a formal change control process with respect to baselined requirements and associated requirements traceability information.

5. REQUIREMENTS FOR A REQUIREMENTS TRACEABILITY REPORT IN SYSTEM DESIGN

The RTR-SD should contain all, or a user-selected subset, of the following minimum information, selectable in all combinations by the creator of the RTR-SD:

a. The system (software, etc.) of which the parent requirements are requirements
b. For each requirement, the status of the requirement
c. For each parent requirement, the requirement REQID
d. For each parent requirement, Class “Parent”
e. For each child requirement, the requirement REQID
f. For each child requirement, Class “Child”
g. For each child requirement, the system solution element (technology item, support item, etc.) of which the requirement is a requirement
h. For each parent requirement, the requirement text, or a link to the requirement text
i. For each child requirement, the requirement text, or a link to the requirement text
j. For each parent requirement, the one or more child requirements derived by design
k. For each child requirement, the one or more parent requirements
l. For each requirement, the baselining status of the requirement (i.e. Baselined or Un-baselined)
m. For each requirement, its Verification Status - one of Not Yet Verified/Passed/Failed/Passed After Re-verification/Not Applicable.

For each parent Non-Requirement:

a. the “requirement” REQID
b. the “requirement” text, or a link to the “requirement” text
c. the date and time of creation of, or of any previous change to the “requirement”, whichever is the later
d. the reason for classification as a Non-Requirement
e. the identity of the person or persons authorizing the classification “Non-Requirement”.

Note that, by definition, parent non-requirements cannot have children.

An RTR-SD may contain the following additional information with respect to requirements of status: “Baselined Requirement” and “Un-baselined Requirement”, depending on need:

a. for each child requirement, the one or more verification requirements that state(s) the quality or strength of evidence that is required to adequately verify its satisfaction by the system element.
b. for each requirement that has been baselined, the date and time of creation of, or of any previous change to the requirement, whichever is the later.

6. OTHER POTENTIAL REPORTS

Other reports may be generated flexibly or be included in the RTR-SD, especially:

a. list of child requirements each with corresponding verification status.
b. list of child requirements each with identity and text of corresponding verification requirements.
c. list of child requirements, each with corresponding text and rationale.
d. Up-Down trace, showing for a given child requirement on a given solution element, the parent(s), and for each parent, the corresponding children, and for each of these children, the system solution element of which the child requirement is a requirement, possibly incorporating traceability relationships through greater than two hierarchical physical levels.
e. a variety of verification-related reports.

Although requirements traceability in design is always between requirements at adjacent hierarchical physical levels, the RTR-SD may be extended to incorporate requirements traceability from Physical Level 1 to Physical Level 2 to Physical Level 3, etc. Traceability must never jump physical levels.