| **Name** | **Documentation** | **Owner** |
| --- | --- | --- |
| Case | b (1) : a situation requiring investigation or action (as by the police) (2) : the object of investigation or consideration  [Merriam Webster on-line dictionary]  Note - Can also be an analysis case, etc. (refer to Hans Peter Property Model) | .Requirement |
| Component Specification | A specification is defined as: a : a detailed precise presentation of something or of a plan or proposal for something —usually used in plural [11, Merriam-Webster on-line] A Component Specification provides the physical, behavioral, and interface specification for a component to be designed and built. [1, created for SECM] | .Requirement |
| Conforms Relationship | The Conforms Relationship assumes that the realization has (or will) provided a satisfaction relationship from its constrained elements to each of the applicable requirements in the Component Specification. It can be thought of as an abstraction of the set of satisfy relationships.  The "conforms to" connection will be able to specify what part, or subset of requirements, of the specification are applicable. This can be done by identifying each requirement ID or by identifying one or more requirement groups IDs. See the definition for the satisfy relationship to see the distinction in the terms satisfy, conforms, realize and specify, as defined for this effort.   [1, created for SECM] | .Requirement |
| Constrained Element | This element is bound to one or more elements in the realization that is constrained by the requirement. The constrained element is used by the Constraint Evaluation to determine is the system satisfied the requirement as stated in the constraining element.   In a traditional textual requirement statement, this element is equivalent to the subject of the verb "shall" (or will, should, etc.).  [1, created for SECM] | .Requirement |
| Constraining Element | The element that defines the requirement is constraining. This element can apply to any model element including properties, behaviors, etc. [1, created for SECM] | .Requirement |
| Constraint Evaluation | Includes evaluation of satisfaction (assertion) and verification (evaluation):  1. Parent requirement is fully satisfied if logical combination of the following are satisfied: - child requirements are satisfied - derived requirements are satisfied - design elements satisfy each requirement 2. Can include margin  [1, created for SECM] | .Requirement |
| Context Element | An element that is referenced in the formal requirement statement that is contained within the context of the constraining element. [1, created for SECM] | .Requirement |
| Decompose Relationship | A decompose relationship is established from a requirement group to a requirement. The intent is to add constraints. There can be 1 or more decompose relationships from a requirement group. They are used to identify the set of requirements within a requirement group. [1, created for SECM] | .Requirement |
| Derive Relationship | A derived relationship imposes constraints to meet a higher level constraint. A derived relationship indicates a requirement has been added. It can be used in two ways; 1. A relationship between a higher architectural level requirement and one or more requirements derived in lower architectural components. 2. A relationship between 2 requirements on the same architectural level where one constraining element constrains another. [1, created for SECM] | .Requirement |
| Design Constraint | One of the potential category requirement selections available in the Requirement Type attribute. This selection identifies the requirement to be imposed during the design process. [1, created for SECM] | .Requirement |
| Formal Requirement Statement | A formal requirement captures all aspects of a requirement in a machine readable form, vs. text in a Textual requirement. This allows requirements to be used to automate tasks associated with validation of requirement information, verification of requirements and the use of the requirement parameters during system analysis.   Transformation of a textual requirement to a new formal requirement is one means of deriving a formal requirement. A textual view of a set of selected requirements is very useful to a user performing analysis or for a review. [1, created for SECM] | .Requirement |
| Functional Requirement | One of the potential category of requirement selections available in the Requirement Type attribute. This selection identifies the requirement to be a functional requirement. [1, created for SECM] | .Requirement |
| Interface Requirement | One of the potential category requirement selections available in the Requirement Type attribute. This selection identifies the requirement to be associated with an interface. [1, created for SECM] | .Requirement |
| Originator/Author Attribute | The originator/author is the person responsible for entering the requirement. [14, Guide Writing Requirements, 5.3.3] | .Requirement |
| Owner Attribute | Identifies the person or element of the organization that maintains the requirement, who has the right to say something about this requirement, approves changes to the requirement, and reports the status of the requirement. [14, Guide Writing Requirements, 5.3.5] | .Requirement |
| Performance Requirement | One of the potential category requirement selections available in the Requirement Type attribute. This selection identifies the requirement to be a performance requirement. [1, created for SECM] | .Requirement |
| Physical Requirement | One of the potential category of requirement selections available in the Requirement Type attribute. This selection identifies the requirement to be a physical related requirement. [1, created for SECM] | .Requirement |
| Precondition Expression | Something that must exist or happen before something else can exist or happen [11, Merriam-Webster on-line definition] Types of preconditions can include:  - Property values  - Events  - Textual Statements  - provided in external referenced documents [1, created for SECM] | .Requirement |
| Priority Attribute | This is how important the requirement is to the stakeholders. It may not be a critical requirement (that is, one the system must possess or it won’t work at all), but simply something that the stakeholder(s) hold very dear. Priority may be characterized in terms of a level (1, 2, 3 or high, medium, low). Priority may be inherited from a parent requirement. High priority requirements must always be met for the project to be successful; lower priority requirements may be traded off when conflicts occur or when there are budget or schedule issues. [14, Guide Writing Requirements, 5.3.20] | .Requirement |
| Process Requirement | One of the potential category requirement selections available in the Requirement Type attribute. This selection identifies the requirement to be imposed on a process that is to be used during a development, manufacturing, support or maintenance process. [1, created for SECM] | .Requirement |
| Rationale Relationship | A rationale relationship is used between two elements to establish traceability between the element being rationalized, the rationalized element, and the justifying element.   A rationalized element can be any type of element including elements such as blocks, requirements or relationships. Typical rationalized relationship elements include a derived or satisfy relationship.  The justifying element contains the rationalization, explanation or justification for rationalized element. Justifying elements can include comment containing a text statement or an analysis.  [1, created for SECM] | .Requirement |
| Reliability Requirement | One of the potential category of requirement selections available in the Requirement Type attribute. This selection identifies the requirement to be a reliability related requirement. [1, created for SECM] | .Requirement |
| Required/Desired | This is a property of the formal requirement statement that defines if the requirement is a mandatory (required) requirement or a desired requirement from a customer perspective. In textual requirement statements the verb "shall" or "will" are typically used, respectively, to do this. When a textual form of the formal requirement statement is automatically generated then this verb will be used in the textual statement. [1, created for SECM] | .Requirement |
| Requirement | This concept represents a usage of a Requirement Definition and is therefore typed by a Requirement Definition. [1, created for SECM] | .Requirement |
| Requirement Allocation Relationship | A relationship from a higher architectural level (n) requirement to one or more lower level (n-1) architectural component with the intent it will be satisfied.   It is used to maintain traceability from the source requirement that was ultimately used to derive requirements for lower architectural components.  During this allocation a value contained in the source requirement may be distributed to two or more components. See the Requirement Needs document for an explanation of where and how this relationship is used.   [1, created for SECM] | .Requirement |
| Requirement Attribute | An attribute is additional information included with a requirement statement, which is used to aid in the management of that requirement. [14, Guide Writing Requirements, Definitions) | .Requirement |
| Requirement Attribute Library | This library contains a set of requirement attributes and requirement types (categories) which are typically used by projects. This list should include all attributes and types defined in the INCOSE Requirements Writing Guide [14].  The intent is to make this library available to each organization and/or project to allow that organization or project to select which best fit their workflow needs. [1, created for SECM] | .Requirement |
| Requirement Definition | Statement that identifies a product\* or process operational, functional, or design characteristic or constraint, which is unambiguous, testable or measurable, and necessary for product or process acceptability. (ISO/IEC 2007)   \*includes product, service, or enterprise.[3, SEBoK Glossary] | .Requirement |
| Requirement Group | A grouping or organization of requirements. This can be an entity of a specification that contains a set of related requirements or it can any grouped set of requirements to facilitate any analysis task.   In a black box specification a requirement context could be a functional items, external interfaces, or topic areas such as security, safety, design constraints, etc. In a white box requirement a context could also be a sub-components.   The context can contain other related supporting information to help understand the requirements such as examples from other systems.  [1, created for SECM] | .Requirement |
| Requirement Refine Relationship | A requirement refinement relationship is used between any two requirements or between a requirement group and one or more requirements, where one or more requirements are used to re-express the first element more precisely and with less ambiguity. Uses for between two requirements to include; 1. When a text requirement statement is refined with a formal requirement statement.  2. When a text requirement statement is initially defined and a later version contains improved text or transformed to a formal requirement statement. A refine relationship can also be used to refine a compound requirement to two or more singular requirements. A compound requirement is a concatenation of 2 or more requirements. This compound requirement should be decomposed into its constituent part requirements. This relationship established a connection between a compound requirement and its constituent requirements.  A logical aggregation of the constituent requirements requirement is equivalent to the original compound parent requirement. (note that this could be defined as a kind of a Requirement Group) [1, created for SECM] | .Requirement |
| Requirement Status Attribute | This requirement attribute is intended to maintain the current status of the requirement. Typical values can include "draft", "ready for review", "accepted", "rejected", "implemented" and "verified".  A requirement can continue to change after being accepted, implemented and/or verified. This change control management is typically managed via the same change control process as other model elements. [1, created for SECM] | .Requirement |
| Requirement Type/Category | Each organization will define types or categories to which a requirement fits, based on how they may wish to organize their requirements. The Type/Category field is most useful because it allows the requirements database to be viewed by a large number of designers and stakeholders for a wide range of uses. [14, Guide Writing Requirements, 5.3.25] | .Requirement |
| Restricted Requirement Statement | A specific type of Textual Requirement Statement, specified by using a restricted/controlled natural language that puts restrictions on grammar (which can be realized by templates and patterns) and vocabulary (by using e.g., pre-defined keywords). Restricted Requirement Statements (RRS) strikes a balance between practicality and level of automation, bridges the gap from informal requirements specifications in natural language to formal, precise, and analyzable specifications.  [1, created for SECM] | .Requirement |
| Result Expression | This is the expression that determines the boundary used by the Constraint Evaluation. This boundary expression can be as simple as a text statement or define a volume in three dimensional space. It can presented in many forms including a table, equation, multidimensional graph or text. [1, created for SECM] | .Requirement |
| Risk Attribute | A risk value assigned to each requirement based on risk factors. Requirements that are at risk include requirements that fail to have the set of characteristics that all well-formed requirements must have: necessary, singular, conforming, appropriate, correct, unambiguous, complete, feasible, and verifiable. Risk can also address feasibility/attainability in terms of technology, schedule, and cost. If the technology needed to meet the requirement is new with a low maturity, the risk is higher than if using a mature technology you have used in other similar projects. The requirement can be high risk if the cost and time to develop a technology is outside what has been planned for the project. Risk may be inherited from a parent requirement. [14, Guide Writing Requirements, 5.3.22] | .Requirement |
| Safety Requirement | One of the potential category of requirement selections available in the Requirement Type attribute. This selection identifies the requirement to be a safety related requirement. [1, created for SECM] | .Requirement |
| Satisfy Relationship | A relationship between a requirement and the constrained element (realized element) or its context (the container for the realized element), which asserts the constraint evaluation is true between the constraining element and the constrained element.  There are other terms that are sometimes used for the term satisfy such as conforms, realize and specify. For this effort we have defined these terms as follows:  - The terms realize and conforms are synonymous. Both of these terms are essentially defined as an abstraction of the set of satisfy relationships between the constrained elements and the requirements in a component spec. See the definition for conforms relationship for more information. We chose to use conforms vs. realize.  - The term specify is the opposite of conforms and realize, i.e. converses of each other. Therefore if specify relationship did exist it would go from a Component Specification to the Component Realization. We chose to use one term not both and chose conforms vs specify. [1, created for SECM] | .Requirement |
| Security Requirement | One of the potential category of requirement selections available in the Requirement Type attribute. This selection identifies the requirement to be a security related requirement. [1, created for SECM] | .Requirement |
| Supportability Requirement | One of the potential category of requirement selections available in the Requirement Type attribute. This selection identifies the requirement to be a supportability related requirement. [1, created for SECM] | .Requirement |
| Textual Requirement Statement | The traditional "shall" textual statement used to state a requirement. [1, created for SECM] | .Requirement |
| Trace Relationship | A generalized relationship between a requirement and one or more source elements. Since a requirement is an element the source can be a requirements. It has sometimes been used prior to making a clear distinction of which relationship to use.   Is the trace relationship a real SE need assuming that all needs have been captured in the other listed relationships? If so we should state the need.  [1, created for SECM]  As defined from UML 2.5 Definition; Specifies a trace relationship between model elements or sets of model elements that represent the same concept in different models. Traces are mainly used for tracking requirements and changes across models. As model changes can occur in both directions, the directionality of the dependency can often be ignored. The mapping specifies the relationship between the two, but it is rarely computable and is usually informal. [15, UML] | .Requirement |
| Uncategorized Requirement | One of the potential category of requirement selections available in the Requirement Type attribute. This selection identifies that categorizing requirements is part of the organization's process but the task has not been completed. [1, created for SECM] | .Requirement |
| Unique Identifier | This is a unique identifier, which can be either a number or mixture of characters and numbers used to refer to the specific requirement. The unique identifier is not a paragraph number.  It can be a separate identifier or automatically assigned by whatever Requirement Management Tool (RMT) the organization is using. This identifier is used once and never reused.   A unique identifier is also needed to link requirements in support of the flow down of requirements (allocation), traceability, and to establish peer-to-peer relationships. Some organizations include in the unique identifier codes that relate to the SOI to which the requirement applies: e.g., [SOI] 1234. [14, Guide Writing Requirements, 5.3.1]  The ID typically includes some intelligence built into the number to help humans relate to its context (for example CR\_100 for a customer requirement, where CR\_ is a user-defined prefix unique to a requirement specification, and 100 is tool generated). [1, created for SECM] | .Requirement |
| Usability Requirement | One of the potential category of requirement selections available in the Requirement Type attribute. This selection identifies the requirement to be a user usability related requirement. [1, created for SECM] | .Requirement |
| User Defined Requirement Attribute | This is a requirement attribute that is not available in the Requirement Attribute Library but can be can be created and defined by a specific organization or project to meet the needs of their workflow. [1, created for SECM] | .Requirement |
| Verification | (1a) Confirmation, through the provision of objective evidence, that specified (system) requirements have been fulfilled. (ISO/IEC 2008, section 4.38)    (1b) Verification is a set of activities that compares a system or system element against the required characteristics. This includes, but is not limited to, specified requirements, design description and the system itself. The system was built right. (ISO/IEC/IEEE 2015, 1, Section 6.4.6)    (2) The evaluation of whether or not a product, service, or system complies with a regulation, requirement, specification, or imposed condition. It is often an internal process. Contrast with validation. (PMI 2013)  [3, SEBoK definition] | .Requirement |
| Verification Activity | A requirement applied to the means of establishing compliance of an end item with its specification requirements.  [1, created for SECM]  Suggested Change in Definition- Process for acquiring the data needed to evaluate compliance. | .Requirement |
| Verification Case | A structured scenario that describes a verification objective and individual steps representing the verification activities required. [1, created for SECM] | .Requirement |
| Verification Context | An environment context that supports the ability to ensure that requirements have been met. [1, created for SECM] | .Requirement |
| Verification Evaluation Activity | An activity that compares the verification data produced by the verification activity with the verification success criteria. [1, created for SECM] | .Requirement |
| Verification Method | The verification method for each requirement simply states the planned method of verification (inspection, demonstration, test, analysis, simulation). [10, INCOSE Handbook]  The Description property provides a textual description of the steps that will be taken in Verification Activity and Verification Evaluation Activity. [1, created for SECM]  The type of method may also include sampling and analogy. [2, SEBoK Verification] | .Requirement |
| Verification Outcome | Describes the data and any other results from performing the Verification Activity. [1, created for SECM] | .Requirement |
| Verification Requirement | A requirement applied to the means of establishing compliance of an end item with its specification requirements. [1, created for SECM] | .Requirement |
| Verification Result | The result of the Verification Evaluation. [1, created for SECM] | .Requirement |
| Verification Success Criteria | The success criteria is a subset of the requirement being verified (e.g. selected points in a flight test envelope). [1, created for SECM] | .Requirement |
| Verification System | An aggregation of enabling elements needed to perform verification activities. This includes the equipment, users and facilities used to perform the activity. [1, created for SECM] | .Requirement |
| Verification System Element | A system element used to stimulate and interact with the unit under verification during the execution of the verification case. [1, created for SECM] | .Requirement |
| Verify Relationship | A relationship between a requirement and a verification case that can be elaborated to specify how verification of the requirement is accomplished and to produce a result from the constraint evaluation. [1, created for SECM] | .Requirement |

# Bibliographic Citation List

References and citations are shown in the glossary definition text as a number within square brackets.

1. Created for SECM - This citation indicates that this text was created specifically for the SECM and no other reference is known. Enter [1, created for SECM] at the end of the text field.
2. INCOSE. 2011. INCOSE Systems Engineering Handbook, Version 3.2.2. San Diego, CA, USA: International Council on Systems Engineering (INCOSE), INCOSE-TP-2003-002-03.2.2.
3. BKCASE Editorial Board. 2015. The Guide to the Systems Engineering Body of Knowledge (SEBoK), v. 1.5. R.D. Adcock (EIC). Hoboken, NJ: The Trustees of the Stevens Institute of Technology. Accessed DATE. [www.sebokwiki.org](http://www.sebokwiki.org). BKCASE is managed and maintained by the Stevens Institute of Technology Systems Engineering Research Center, the International Council on Systems Engineering, and the Institute of Electrical and Electronics Engineers Computer Society.
4. International Standard - ISO/IEC 15288 and IEEE 15288 - 2008, Second Edition 2008-02-01, Systems and software engineering - System life cycle processes
5. ISO/IEC 2008. Systems and Software Engineering -- System Life Cycle Processes. Geneva, Switzerland: International Organization for Standardization / International Electromechanical Commissions. ISO/IEC/IEEE 15288:2008 (E).
6. Wikipedia: Safety: Mar 31, 2015: <http://en.wikipedia.org/wiki/Safety#Safety_measures>
7. Douglas, Bruce: Safety Analysis of UML Models
8. Wikipedia. Main Page. Mar 31, 2015. <http://en.wikipedia.org>
9. Roedler, G.J. and Jones, C. December 27, 2005. Technical Measurement, Version 1.0, Practical Software and Systems Measurement (PSM) and International Council on Systems Engineering (INCOSE). INCOSE-TP-2003-020-01
10. INCOSE (2015). Systems Engineering Handbook: A Guide for System Life Cycle Process and Activities (4th ed.) D. D. Walden, G. J. Roedler. K. J. Forsberg, R.D. Hamelin, and, T. M. Shortell (Eds.). San Diego, CA: International Council on Systems Engineering. Published by John Wiley & Sons, Inc.
11. Merriam Webster on-line dictionary
12. UML 4SE RFP. SE Definitions List, April 01 2003: <http://syseng.omg.org/UML%20for%20SE%20Definitions%20030401.xls>
13. Business Dictionary.com - <http://www.businessdictionary.com/>
14. INCOSE. 2015. Guide for Writing Requirements. Version 2, San Diego, CA, USA: International Council on Systems Engineering (INCOSE), INCOSE-TP-2010-006-02.
15. OMG Unified Modeling Language ™ (OMG UML), Version 2.5, March 2015, OMG Document Number - formal/2015-03-01
16. OMG Systems Modeling Language (OMG SysML™), Version 1.4, September 2014, OMG Document Number: formal/2015-06-03
17. ISO Online Browsing Platform (OBP), Terms and Definitions, <https://www.iso.org/obp/ui/#home>
18. Weilkiens, Tim: Variant Modeling with SysML, MBSE4U - Tim Weilkiens, Apr 12 2016, ISBN 978-3-9817875-4-2