Martin Glinz

# A Glossary of Requirements Engineering Terminology

With an English–German and German–English Dictionary of Terminology

Standard Glossary for the Certified Professional for Requirements Engineering (CPRE) Studies and Exam

IREB
International Requirements Engineering Board

University of Zurich UZH

**Department of Informatics**

Requirements Engineering Research Group
RERG

**About the Author**

Martin Glinz is a full professor of Informatics and head of the Requirements Engineering Research Group at the University of Zurich, Department of Informatics. His interests include requirements and software engineering – in particular, modeling, validation, and quality – and software engineering education.

He received a diploma in Mathematics and a Dr. rer. nat. in Computer Science, both from RWTH Aachen University. Before joining the University of Zurich, he worked in industry for ten years, where he was active in software engineering research, development, training, and consulting.

Martin Glinz has over 25 years of experience in Requirements Engineering, both academic and industrial. He is on the editorial board of the Requirements Engineering journal and chaired the steering committee of the IEEE International Requirements Engineering Conference from 2006-2009.

Version 1.1     May 2011

# Table of Contents

# The IREB Certified Professional for Requirements Engineering (CPRE)

In 2007, the International Requirements Engineering Board (IREB e.V.) was founded. It is composed of independent experts who all have a strong background in Requirements Engineering, covering a wide area of domains, including industry, consulting, research, and education. Many of them are known world-wide for their contributions to the field.

The members of the board have created a curriculum for the domain of requirements engineering and have developed a certificate, the Certified Professional for Requirements Engineering (CPRE), that is based on the curriculum. The goal is to establish standardized, high-quality instruction and continuing training in requirements engineering, thereby improving the practice of requirements engineering. The glossary presented in this document has been endorsed by IREB as the standard glossary of terms for the CPRE.

In 2007, the IREB started out very successfully in Germany, Austria, and Switzerland. Since the curriculum has been made available in English, the CPRE is now offered in more and more countries and has become truly international. Translations into French and Spanish are in progress.

Four main actors are involved in the certification process: the IREB, recognized training providers, certification authorities of the individual countries, and of course the participants in training courses and the examinees.

The IREB creates the curriculum, develops the corresponding examination questions, defines and governs the examination process, authorizes certification authorities to administer examinations, and recognizes training providers whose training courses conform to the curriculum for the Certified Professional for Requirements Engineering exam. In the individual countries, IREB-commissioned certification authorities administer the examinations for the certificate.

Formally, the IREB curriculum is similar to the curricula of other established instruction and continued training standards (e.g., ISTQB Certified Tester) and takes the pertinent standards of ISO and IEEE into consideration. The curriculum for the Foundation Level comprises the essential knowledge of requirements engineering, including requirements elicitation, documentation, validation, and management. The content the IREB certificate covers can be reviewed in the publicly available curriculum. Through its curriculum, the IREB provides a guideline for the amount of material to be covered during training, the training contents, and the time required for achieving the learning goals and carrying out practical exercises. The IREB syllabi are complemented by this glossary of Requirements Engineering terminology and by other, supplementary materials.

All information about the International Requirements Engineering Board (IREB e.V.) and about the Certified Professional for Requirements Engineering certification can be found on the IREB website:

http://www.certified-re.org/en

# Preface

When looking for definitions of terms in Requirements Engineering, one can find definitions for almost any term by searching the web. However, such searching requires effort and the quality of the results is unpredictable. Frequently, definitions found in different sources are inconsistent with each other. Existing glossaries in Requirements Engineering textbooks mostly focus on the topics covered in these books. Systematic translations of terminology into major languages other than English are missing completely.

This glossary aims at collecting the existing knowledge on Requirements Engineering terminology and defining the core terminology carefully and consistently. In cases where more than one definition is in use or where terms are defined differently when viewed from different perspectives, multiple definitions or perspectives are included. For terms having both a general meaning and a specific meaning in a Requirements Engineering context, both meanings are defined. Important terms are annotated with hints and additional information. Additionally, all terms are translated into German. Translations into other languages, in particular French and Spanish, are planned for the future.

This glossary complements the textbooks endorsed by the International Requirements Engineering Board (IREB). The definitions in the forthcoming textbook *Requirements Engineering Fundamentals* by Klaus Pohl and Chris Rupp and the definitions in this glossary have been aligned with each other.

The sources I used for defining the terms are listed in the references. I don't cite sources for individual definitions because I deliberately decided not to compile definitions from various existing sources just by copy-paste, but to carefully re-formulate all definitions consistently and according to today's use. Of course this doesn't exclude that some definitions or parts of them have been taken verbatim from one of the referenced sources – I don't intend to re-invent the world. Some definitions are a result of joint work with others. Having been in the field of Requirements Engineering for more than 25 years, another source, for which I can't cite any individual references, is my personal knowledge and experience of how terms are used both in academia and industry. Credits for definitions taken from other sources and for joint work with others are given in the *Sources* Section on page 20.

This is ongoing work. In the current state, 128 terms have been defined, covering the base terminology to a large extent. There are still some gaps with respect to the terminology related to processes, project management, and product management. Special terms, for example of specific techniques for requirements elicitation or conflict resolution, are also still missing. I plan to fill these gaps incrementally in the future.

The glossary consists of three parts: the definitions of terms, the English-German dictionary and the German-English one. I hope that both learners and professionals in Requirements Engineering will find this structure useful and consider consulting the glossary to be worthwhile.

I want to thank the IREB members, in particular, Karol Frühauf, Colin Hood, Klaus Pohl, Chris Rupp with her Sophist team, and Thorsten Weyer, for their contributions to this glossary and for numerous comments that helped improve its quality. I also gratefully acknowledge the patience of many people who waited for more than one year for this glossary to appear. Most of all, I thank my wife Angelika. Without her love, patience and understanding, most of my professional work, including this one, would not have been possible.

*Martin Glinz*
*Zurich, May 2011*

# Part One: Definitions of Terms

| | |
|---|---|
| **Acceptance** | The process of assessing whether a ↑system satisfies all its ↑requirements |
| **Acceptance test** | A test that assesses whether a ↑system satisfies all its ↑requirements. |
| **Activity diagram** | A diagram type in ↑UML which models the flow of actions in a ↑system or in a ↑component including data flows and areas of responsibility where necessary. |
| **Actor** | 1. Generally in RE: A person, a ↑system or a technical device in the ↑context of a system that interacts with the system.<br>2. Especially in goal-oriented RE: a person, a ↑system or a technical device that may act and process information in order to achieve some ↑goals. |
| **Adequacy** (of a requirement) | The degree to which a ↑requirement expresses the ↑stakeholders' true desires and needs (i.e., those they had actually in mind when stating the requirement). |
| **Application domain** | Those parts of the real world that are relevant for determining the ↑context of a ↑system. |
| **Artifact** | An intermediate or final result of ↑system development; for example, a ↑requirements specification. |
| **Attribute** | A characteristic property of an ↑entity. |
| **Baseline** | A stable, change-controlled ↑configuration of ↑artifacts.<br>Baselines serve for ↑release planning and release definition as well as for project management purposes such as effort estimation. |
| **Behavior model** | A ↑model describing the behavior of a ↑system or ↑component, e.g., by a ↑state machine. |
| **Bug** | → Defect |
| **Cardinality** | 1. In modeling: The minimum and maximum number of objects in a relationship. In ↑UML, the term multiplicity is used for cardinality.<br>2. In mathematics: The number of elements in a set. |
| **Change control board** | A committee of client and supplier representatives that decides on ↑change requests.<br>Abbreviation: CCB |
| **Change request** | In RE: A well-argued request for changing one or more ↑baselined ↑requirements. |
| **Changeability** (of an artifact) | The degree to which an ↑artifact enables a required modification of the artifact. |
| **Checking** (requirements) | Comprises ↑requirements ↑validation and checking requirements for qualities such as ↑unambiguity or comprehensibility.<br>Note that some sources define validation broader and consider the terms *checking* and *validation* to be ↑synonyms. |
| **Class** | Represents a set of objects of the same kind by describing the structure of the objects, the ways they can be manipulated and how they behave. |

**Class diagram**

| | |
|---|---|
| **Class diagram** | A diagrammatic representation of a ↑class model. |
| **Class model** | A model consisting of a set of classes and relationships between them. |
| **Completeness** (of requirements) | 1. For a single requirement: The degree to which a requirement contains all necessary information<br>2. For a requirements specification: The degree to which the specification contains all information which is necessary for developing a system that satisfies the ↑stakeholders' desires and needs. |
| **Compliance** | The capability of an ↑artifact to adhere to ↑standards, regulations, laws, or other formally imposed documents.<br>↑Systems frequently need to comply with standards, regulations, and laws constraining the domain where the system is deployed. Such compliance can only be ensured systematically if compliance checking already starts with the ↑requirements. |
| **Component** | 1. In general: A delimitable part of a ↑system.<br>2. In software architecture: An encapsulated set of coherent objects or ↑classes that jointly provide a service.<br>Note: When viewed in isolation, a component is a ↑system by itself. |
| **Configuration** | A consistent set of logically coherent units. The units are individually identifiable ↑artifacts or parts of artifacts (e.g., ↑requirements) in at most one version per unit. |
| **Conformity** (of requirements) | The degree to which a ↑requirements specification conforms to regulations given in some ↑standard. |
| **Consistency** (of requirements) | The degree to which a set of ↑requirements is free of contradicting statements. |
| **Constraint** | A ↑requirement that limits the solution space beyond what is necessary for meeting the given ↑functional requirements and ↑quality requirements. |
| **Context** | 1. In general: The network of thoughts and meanings needed for understanding phenomena or utterances.<br>2. Especially in RE: The part of a ↑system's environment being relevant for understanding the system and its ↑requirements.<br>Context in the second meaning is also called the ↑system context. |
| **Context boundary** | Boundary between the ↑context of a ↑system and those parts of the ↑application domain that are irrelevant for the ↑system and its ↑requirements.<br>The context boundary separates the relevant part of the environment of a system to be developed from the irrelevant part, i.e., the part that does not influence the system to be developed and, thus, does not have to be considered during requirements engineering. |
| **Context diagram** | 1. A diagrammatic representation of a ↑context model.<br>2. In ↑Structured Analysis, the context diagram is the root of the dataflow diagram hierarchy. |
| **Context model** | A ↑model describing a ↑system in its ↑context. |
| **Correctness** | The degree to which the information contained in an ↑artifact is provably true.<br>In RE, correctness is frequently used as a synonym for ↑adequacy. |

| | |
|---|---|
| **Customer** | A person or organization who receives a product or service.<br>Also see ↑stakeholder. |
| **Customer requirements specification** | A coarse description of the required capabilities of a ↑system from the ↑customer's perspective.<br>Usually supplied by the customer. |
| **Dataflow diagram** | A diagram modeling the ↑functionality of a ↑system or ↑component by *processes* (also called *activities*), *data stores* and *data flows*. Incoming data flows trigger processes which then consume the received data, transform them, read/write persistent data held in data stores and then produce new data flows which may be intermediate results that trigger other processes or final results that leave the system. |
| **Decision table** | A tabular, systematic representation of a complex decision that depends on multiple criteria. |
| **Defect** | A spot in an ↑artifact that is incorrectly described or crafted.<br>Synonym: fault, bug |
| **Domain** | A range of relevant things (for some given matter); for example, an ↑application domain. |
| **Effectiveness** | The degree to which something actually happens in the way it ought to happen.<br>In RE, typically the degree to which a ↑system actually enables its ↑users to achieve their ↑goals as stated in the system's ↑requirements. |
| **Efficiency** | The degree to which a result is achieved with minimum consumption of resources. |
| **Elicitation** (of requirements) | → Requirements elicitation |
| **End user** | → User |
| **Entity** | 1. In general: an element or set of elements that may stand for any conceivable item, e.g., a ↑system, a part of reality, a thing, an organization, a process, etc.<br>2. In entity-relationship-modeling: an individual object which has an identity and does not depend on another object. |
| **Entity-relationship diagram** | A graphic representation of an ↑entity-relationship model.<br>Abbreviation: ERD |
| **Entity-relationship model** | A ↑model of data that are relevant for a ↑system, or of the data of an ↑application domain. An ERM consists of a set of entity types that are each characterized by ↑attributes and linked by relationships.<br>Abbreviation: ERM, ER Model |
| **Error** | A discrepancy between an observed behavior or result and the specified behavior or result.<br>An error typically is a symptom for the existence of a ↑fault or ↑defect in some ↑artifact.<br>In colloquial English, there is sometimes no distinction between the notions of error and fault. |
| **Fault** | → Defect |

| **Fault Tolerance** | |
|---|---|
| **Fault Tolerance** | The capability of a ↑system to continue normal operation despite the presence of (hardware or software) ↑faults.<br>Fault tolerance may be stated as a ↑quality requirement. |
| **Feature** | A delimitable characteristic of a ↑system that provides value for ↑stakeholders.<br>Normally comprises several ↑requirements and is used for communicating with stakeholders on a higher level of abstraction and for expressing variable or optional characteristics. |
| **Functional requirement** | A ↑requirement concerning a result of behavior that shall be provided by a *function* of a ↑system (or of a ↑component or service). |
| **Functionality** | The capabilities of a ↑system as stated by its ↑functional requirements. |
| **Glossary** | A collection of definitions of terms that are relevant in some ↑domain. Frequently, a glossary also contains cross-references, ↑synonyms, ↑homonyms, acronyms, and abbreviations. |
| **Goal** | A desired state of affairs (that a ↑stakeholder wants to achieve).<br>Goals describe intentions of ↑stakeholders. They may conflict with one another. |
| **Goal model** | A ↑model that represents the ↑goals of something as an ordered structure of sub-goals. |
| **Homonym** | A term looking identical to another term, but having a different meaning.<br>For example, *bill* as a bank note and *bill* as a list (of materials) are homonyms. |
| **Inspection** | A kind of ↑review where the ↑artifact under review is inspected by a group of experts according to given criteria. The experts' findings are then collected and consolidated. |
| **Kind of requirement** | There are several kinds of ↑requirements. ↑Requirements Engineering is primarily concerned with ↑system requirements. Beyond that, there are *project requirements* and *process requirements*.<br>Requirements are typically sub-classified into ↑functional requirements, ↑quality requirements and ↑constraints. The latter two are also called ↑non-functional requirements. |
| **Language** | A structured set of signs for expressing and communicating information. Signs are elements that are used for communication: expressions in a language, symbols, gestures, etc. |
| **Maintainability** | The ease with which a software ↑system can be modified to correct ↑faults or adapt the system to changing needs.<br>Maintainability may be stated as a ↑quality requirement. |
| **Model** | An abstract representation of an existing reality or a reality to be created.<br>This definition covers the most frequent case in requirements engineering, but is a bit narrow. More generally speaking, a model is an abstract representation of an existing ↑entity or an entity to be created, where *entity* denotes any part of reality or any other conceivable set of elements or phenomena, including other models. With respect to a model, the entity is called the *original*.<br>In ↑Requirements Engineering, ↑requirements can be specified by models.<br>Note that ↑entity in this definition is used in its general meaning which is *different* from the one used in ↑Entity-relationship models. |

| | |
|---|---|
| **Modeling language** | A ↑language for expressing ↑models of a certain kind. May be textual, graphic, symbolic or some combination thereof. |
| **Multiplicity** | → Cardinality |
| **Non-functional requirement** | A ↑quality requirement or a ↑constraint. <br> ↑Performance requirements may be regarded as another category of non-functional requirements. In this glossary, performance requirements are considered to be a sub-category of quality requirements. <br> Synonym: Extra-functional requirement |
| **Performance requirement** | A ↑requirement describing a performance characteristic (timing, speed, volume, capacity, throughput...). <br> Is regarded in this glossary as a sub-category of ↑quality requirements, but can also be considered as a ↑non-functional requirements category of its own. |
| **Phrase template** | A template for the syntactic structure of a phrase that expresses an individual ↑requirement in natural ↑language. |
| **Portability** | The ease with which a ↑system can be transferred to another platform (while preserving its ↑functionality). <br> Portability may be stated as a ↑quality requirement. |
| **Priority** (of a requirement) | Documents the importance of a ↑requirement in comparison to other requirements according to given criteria. |
| **Process verb** | A verb characterizing the required action in a ↑requirement written in natural ↑language. |
| **Prototype** | 1. In manufacturing: a piece which is built prior to the start of mass production. <br> 2. In software engineering: An executable piece of software that implements critical parts of a ↑system in advance. <br> In ↑Requirements Engineering, prototypes are used as a means for requirements ↑elicitation and ↑validation. |
| **Quality** | The degree to which a set of inherent characteristics of an ↑entity fulfills ↑requirements. <br> The entity may be a ↑system, service, product, ↑artifact, process, person, organization, etc. An *inherent characteristic* is a distinguishing feature of or property of an entity which is inherent to the entity and has not been assigned explicitly. <br> This is the notion of quality that is generally used in industry. Note that quality in this definition just means fitness for intended use, as stated in the requirements. This is in contrast to the colloquial notion of quality which is typically connoted with *goodness* or *excellence*. |
| **Quality requirement** | A ↑requirement that pertains to a quality concern that is not covered by ↑functional requirements. |
| **Redundancy** | Multiple occurrence of the same information or resource. |
| **Release** | A ↑configuration that has been released for installation and use by ↑customers. |
| **Reliability** | The capability of a ↑system to maintain a specified level of ↑functionality and ↑performance when used under specified conditions. <br> Reliability may be stated as a ↑quality requirement. |

| | |
|---|---|
| **Requirement** | |

| | |
|---|---|
| **Requirement** | 1. A condition or capability needed by a ↑user to solve a problem or achieve an objective.<br><br>2. A condition or capability that must be met or possessed by a ↑system or system ↑component to satisfy a contract, standard, specification, or other formally imposed documents.<br><br>3. A documented representation of a condition or capability as in (1) or (2).<br><br>Note: The definition above is the classic one from IEEE Std 610.12 of 1990. Alternatively, we also give a more modern definition:<br><br>1. A need perceived by a ↑stakeholder.<br><br>2. A capability or property that a ↑system shall have.<br><br>3. A documented representation of a need, capability or property. |
| **Requirements analysis** | 1. Analysis of elicited ↑requirements in order to understand and document them.<br><br>2. Synonym for ↑requirements engineering. |
| **Requirements baseline** | A ↑baseline for a set of ↑requirements. |
| **Requirements discovery** | → Requirements elicitation |
| **Requirements document** | A document consisting of a ↑requirements specification.<br><br>Frequently used as a synonym for ↑requirements specification. |
| **Requirements elicitation** | The process of seeking, capturing and consolidating ↑requirements from available ↑requirements sources. May include the re-construction or creation of requirements.<br><br>Synonym: Requirements discovery |
| **Requirements engineer** | A person who – in collaboration with ↑stakeholders – elicits, documents, validates, and manages ↑requirements. |
| **Requirements Engineering** | A systematic and disciplined approach to the ↑specification and management of ↑requirements with the following ↑goals:<br><br>(1) Knowing the relevant ↑requirements, achieving a consensus among the ↑stakeholders about these ↑requirements, documenting them according to given standards, and managing them systematically,<br><br>(2) Understanding and documenting the ↑stakeholders' desires and needs,<br><br>(3) Specifying and managing ↑requirements to minimize the risk of delivering a ↑system that does not meet the ↑stakeholders' desires and needs.<br><br>Abbreviation: RE<br><br>Note: All three goals address important facets of RE: (1) process-orientation, (2) stakeholder focus, and (3) importance of risk and value considerations. |
| **Requirements management** | The process of managing existing ↑requirements and requirements related ↑artifacts. Includes particularly storing, changing and tracing of requirements (↑traceability). |
| **Requirements model** | A ↑model that has been created with the purpose of specifying ↑requirements. |
| **Requirements source** | The source from which a ↑requirement has been derived. Typical sources are ↑stakeholders, documents, existing ↑systems and observations. |
| **Requirements specification** | A systematically represented collection of ↑requirements, typically for a ↑system or ↑component, that satisfies given criteria.<br><br>In some situations we distinguish between a ↑customer requirements specification |

|  | (typically written by the customer) and a ↑system requirements specification or ↑software requirements specification (written by the supplier). Requirements specification may also denote the activity of specifying requirements. |
|---|---|
| **Requirements template** | A blueprint for the syntactic structure of individual ↑requirements. A ↑phrase template is a specific requirements template for requirements written in natural ↑language. |
| **Review** | A formally organized endeavor for checking an ↑artifact by a group of experts. Checking may be performed with respect to both contents and conformance. |
| **Risk** | An event that threatens the success of an endeavor, e.g., of developing or operating a ↑system. A risk is typically assessed in terms of its probability and potential damage. |
| **Safety** | The capability of a ↑system to achieve an acceptable level of probability that operating the system will not result in harming people, property or the environment. Safety requirements may be stated as ↑quality requirements or in terms of ↑functional requirements. |
| **Scenario** | 1. A description of a potential sequence of events that lead to a desired (or unwanted) result. 2. An ordered sequence of interactions between partners, in particular between a ↑system and external ↑actors. May be a concrete sequence (instance scenario) or a set of potential sequences (type scenario, ↑use case). 3. In UML: An execution trace of a ↑use case. |
| **Scope** (of a system) | The range of things that can be shaped and designed when developing a ↑system. |
| **Security** | The capability of a ↑system to protect (a) its data and resources against unauthorized use and (b) its legitimate ↑users against denial of service. |
| **Semantics** | The meaning of a sign or a set of signs in a ↑language. |
| **Semi-formal** | Something which is formal to some extent, but not completely. An ↑artifact is called semi-formal if it contains formal parts, but isn't formalized totally. Typically, a semi-formal artifact has a defined ↑syntax, while the ↑semantics is partially defined only. |
| **Sequence diagram** | A diagram type in ↑UML which models the interactions between a selected set of objects and/or ↑actors in the sequential order that those interactions occur. |
| **Software require-ments specification** | A ↑requirements specification pertaining to a software system. Abbreviation: SRS |
| **Source** (of a requirement) | → Requirements source |
| **Specification** | A systematically represented description of the properties of an ↑entity (a system, a device, etc.) that satisfies given criteria. It may be about required properties (↑requirements specification) or implemented properties (e.g., a technical product specification). |
| **Specification language** | An artificial ↑language that has been created for expressing specifications. |

| **Stakeholder** | |
| --- | --- |
| **Stakeholder** | A person or organization that has a (direct or indirect) influence on a ↑system's ↑requirements. |
| | Indirect influence also includes situations where a person or organization is impacted by the system. |
| **Standard** | A uniform regulation for perceiving, manufacturing or executing something. |
| **State machine** | A ↑model describing the behavior of a system or ↑component by a finite set of states and state transitions. State transitions are triggered by events and can in turn trigger actions and new events. |
| | Related terms: A state machine with atomic states is called a *finite state automaton*. State machines having states that are hierarchically and/or orthogonally decomposed are called *statecharts*. |
| **State-transition diagram** | A diagrammatic representation of a ↑state machine. |
| **Statechart** | A ↑state machine having states that are hierarchically and/or orthogonally decomposed. |
| **Steering committee** | A committee that supervises a project. |
| **Structured Analysis** | An approach for specifying the ↑functionality of a system based on a hierarchy of ↑dataflow diagrams. Data flows as well as persistent data are defined in a data dictionary. A ↑context diagram models the sources of incoming and the destinations of outgoing data flows. |
| **Supplier** | A person or organization who delivers a product or service to a ↑customer. |
| **Synonym** | A word having the same meaning as another word. |
| **Syntax** | The rules for constructing structured signs in a ↑language. |
| **System** | 1. In general: A principle for ordering and structuring. |
| | 2. In Informatics: A coherent, delimitable set of ↑components that – by coordinated action – provides services. |
| | ↑Requirements Engineering is concerned with the ↑specification of ↑requirements for systems consisting of software ↑components, technical elements (computing hardware, devices, sensors,...) and organizational elements (persons, positions, business processes,...). |
| | Note that a ↑system may comprise other systems. Therefore, ↑components or services of a system are also considered to be systems. |
| **System boundary** | The boundary between a ↑system and its surrounding ↑context. |
| | The system boundary separates the ↑system to be developed from its environment; i.e., it separates the part of the reality that can be modified or altered by the development process from aspects of the environment that cannot be changed or modified by the development process. |
| **System context** | The part of a ↑system's environment that is relevant for the definition as well as the understanding of the ↑requirements of a ↑system to be developed. |
| **System requirement** | A ↑requirement pertaining to a ↑system or to a ↑component of a system. |
| **System require-ments specification** | A ↑requirements specification pertaining to a ↑system. |
| | Frequently considered to be a synonym for ↑requirements specification. |

| | |
|---|---|
| **Tool** (in software engineering) | A (software) ↑system that helps develop, operate and maintain ↑systems.<br><br>In RE, tools support ↑requirements management as well as modeling, documenting, and validating ↑requirements. |
| **Traceability** (of requirements) | The ability to trace a ↑requirement (1) back to its origins, (2) forward to its implementation in design and code, (3) to requirements it depends on (and vice-versa). Origins may be ↑stakeholders, documents, rationale, etc.<br><br>Traceability of a requirement back to its origin is also called *pre-RS traceability*. Conversely, traceability of a requirement forward to its implementation in design and code is also called *post-RS traceability*. RS stands for requirements specification.<br><br>Sometimes, traceability to the rationale of a requirement is considered to be a traceability category of its own. |
| **UML** | Abbreviation for Unified Modeling Language, a standardized language for modeling problems or solutions. |
| **Unambiguity** (of requirements) | The degree to which a ↑requirement is expressed such that it cannot be understood differently by different people. |
| **Usability** | The capability of a system to be understood, learned, used, and liked by its ↑users.<br><br>Usability (or parts thereof) may be stated as ↑quality requirements. |
| **Use case** | A description of the interactions possible between ↑actors and a ↑system that, when executed, provide added value.<br><br>Use cases specify a ↑system from a ↑user's (or other external ↑actor's) perspective: every use case describes some functionality that the system must provide for the ↑actors involved in the use case. |
| **Use case diagram** | A diagram type in UML that models the ↑actors and the ↑use cases of a ↑system.<br><br>The boundary between the actors and the use cases constitutes the ↑system boundary. |
| **User** | A person who uses the ↑functionality provided by a ↑system. Also called *end user*. |
| **Validation** (of requirements) | The process of checking whether documented ↑requirements match the ↑stakeholders' needs.<br><br>Note that some sources define requirements validation broader by also including ↑checking requirements for qualities such as ↑unambiguity or comprehensibility, thus considering the terms *validation* and ↑*checking* to be ↑synonyms. |
| **Verifiability** (of requirements) | The degree to which the fulfillment of a ↑requirement by an implemented ↑system can be checked, e.g., by defining ↑acceptance test cases, measurements or inspection procedures. |
| **Version** (of an entity) | If an ↑entity exists in multiple, time-ordered occurrences, where each occurrence has been created by modifying one of its predecessors, every occurrence is a *version* of that entity. |
| **View** | An excerpt from an ↑artifact, containing only those parts one is currently interested in.<br><br>A view can abstract or aggregate parts of the artifact. |
| **Viewpoint** | A certain perspective on the ↑requirements of a ↑system.<br><br>Typical viewpoints are perspectives that a ↑stakeholder or stakeholder group has (for example, an end user's perspective or an operator's perspective). However, there can also be topical viewpoints such as a security viewpoint. |

| **Walkthrough** | |
| --- | --- |
| | Note that this definition is somewhat different from the definition of an architectural viewpoint in the international standard ISO/IEC42010: 2007 (IEEE Std 1471-2000). |
| **Walkthrough** | A kind of ↑review where the author of an ↑artifact under review walks a group of experts systematically through the artifact. The experts' findings are then collected and consolidated. |

# List of Abbreviations

| | |
| --- | --- |
| **CCB** | Change Control Board |
| **CPRE** | Certified Professional for Requirements Engineering |
| **ER** | Entity-Relationship |
| **ERD** | Entity-Relationship Diagram |
| **ERM** | Entity-Relationship Model |
| **IREB** | International Requirements Engineering Board |
| **RE** | Requirements Engineering |
| **SRS** | Software Requirements Specification |
| **UML** | Unified Modeling Language |

## Part Two: English–German Dictionary

| | | | |
|---|---|---|---|
| **Acceptance** | Abnahme | **Efficiency** | Effizienz |
| **Acceptance test** | Abnahmetest | **Elicitation** (of requirements) | Anforderungsermittlung |
| **Activity diagram** | Aktivitätsdiagramm | **End user** | Endbenutzer |
| **Actor** | Akteur | **Entity** | 1. Ein Etwas, 2. Gegenstand, Entität |
| **Adequacy** (of a requirement) | Adäquatheit (einer Anforderung) | **Entity-relationship diagram** | Entity-Relationship Diagramm |
| **Application domain** | Anwendungsbereich | **Entity-relationship model** | Entity-Relationship Modell |
| **Artifact** | Artefakt | **Error** | Fehler |
| **Attribute** | Attribut | **Fault** | Defekt |
| **Baseline** | Basislinie | **Fault Tolerance** | Fehlertoleranz |
| **Behavior model** | Verhaltensmodell | **Feature** | Merkmal (Feature) |
| **Bug** | Defekt | **Functional requirement** | Funktionale Anforderung |
| **Cardinality** | Kardinalität | **Functionality** | Funktionalität |
| **Change control board** | Change control board | **Glossary** | Glossar |
| **Change request** | Änderungsantrag | **Goal** | Ziel |
| **Changeability** (of an artifact) | Änderbarkeit (eines Artefakts) | **Goal model** | Zielmodell |
| **Checking** (requirements) | Prüfung (von Anforderungen) | **Homonym** | Homonym |
| **Class** | Klasse | **Inspection** | Inspektion |
| **Class diagram** | Klassendiagramm | **Kind of requirement** | Anforderungsart |
| **Class model** | Klassenmodell | **Language** | Sprache |
| **Completeness** (of requirements) | Vollständigkeit (von Anforderungen) | **Maintainability** | Wartbarkeit, Pflegbarkeit |
| **Compliance** | Befolgung, Einhaltung | **Model** | Modell |
| **Component** | Komponente | **Modeling language** | Modellierungssprache |
| **Configuration** | Konfiguration | **Multiplicity** | Multiplizität |
| **Conformity** (of requirements) | Konformität (von Anforderungen) | **Non-functional requirement** | Nicht-funktionale Anforderung |
| **Consistency** (of requirements) | Widerspruchsfreiheit, Konsistenz (von Anforderungen) | **Performance requirement** | Leistungsanforderung |
| **Constraint** | Randbedingung | **Phrase template** | Satzschablone |
| **Context** | Kontext | **Portability** | Portabilität, Übertragbarkeit |
| **Context boundary** | Kontextgrenze | **Priority** (of a requirement) | Priorität (einer Anforderung) |
| **Context diagram** | Kontextdiagramm | **Process verb** | Prozesswort |
| **Context model** | Kontextmodell | **Prototype** | Prototyp |
| **Correctness** | Korrektheit | **Quality** | Qualität |
| **Customer** | Kunde | **Quality requirement** | Qualitätsanforderung |
| **Customer requirements specification** | Lastenheft | **Redundancy** | Redundanz |
| **Dataflow diagram** | Datenflussdiagramm | **Release** | Release |
| **Decision table** | Entscheidungstabelle | **Reliability** | Zuverlässigkeit |
| **Defect** | Defekt | **Requirement** | Anforderung |
| **Domain** | Domäne | | |
| **Effectiveness** | Effektivität | | |

| | | | |
|---|---|---|---|
| **Requirements analysis** | Anforderungsanalyse | **State machine** | Zustandsmaschine |
| **Requirements baseline** | Anforderungsbasislinie | **State-transition diagram** | Zustandsdiagramm |
| **Requirements discovery** | Anforderungsermittlung | **Statechart** | Statechart |
| | | **Steering committee** | Lenkungsausschuss |
| **Requirements document** | Anforderungsdokument | **Structured Analysis** | Strukturierte Analyse |
| | | **Supplier** | Lieferant |
| **Requirements elicitation** | Anforderungsermittlung | **Synonym** | Synonym |
| | | **Syntax** | Syntax |
| **Requirements engineer** | Anforderungsanalytiker, Anforderungsingenieur | **System** | System |
| | | **System boundary** | Systemgrenze |
| **Requirements Engineering** | Requirements Engineering | **System context** | Systemkontext |
| | | **System requirement** | Systemanforderung |
| **Requirements management** | Anforderungs- management | **System requirements specification** | System-Anforderungs- spezifikation, Pflichten- heft |
| **Requirements model** | Anforderungsmodell | | |
| **Requirements source** | Anforderungsquelle | **Tool** (in software engineering) | Werkzeug (im Software Engineering) |
| **Requirements specification** | Anforderungs- spezifikation | **Traceability** (of requirements) | Verfolgbarkeit (von Anforderungen) |
| **Requirements template** | Anforderungs- schablone | **UML** | UML |
| **Review** | Review, Durchsicht | **Unambiguity** (of requirements) | Eindeutigkeit (von Anforderungen) |
| **Risk** | Risiko | **Usability** | Benutzbarkeit |
| **Safety** | Sicherheit (Nutzungs- sicherheit) | **Use case** | Anwendungsfall, Use Case |
| **Scenario** | Szenario | **Use case diagram** | Anwendungsfall- diagramm, Use Case Diagramm |
| **Scope** (of a system) | Systemumfang | | |
| **Security** | Sicherheit (Informationssicherheit) | **User** | Benutzer |
| | | **Validation** (of requirements) | Validierung (von Anforderungen) |
| **Semantics** | Semantik | | |
| **Semi-formal** | Teilformal | **Verifiability** (of requirements) | Prüfbarkeit (von Anforderungen) |
| **Sequence diagram** | Sequenzdiagramm | | |
| **Software requirements specification** | Software-Anforde- rungsspezifikation, Pflichtenheft | **Version** (of an entity) | Version (eines Gegenstands) |
| | | **View** | Sicht |
| **Source** (of a requirement) | Anforderungsquelle | **Viewpoint** | Gesichtspunkt, Standpunkt |
| **Specification** | Spezifikation | **Walkthrough** | Walkthrough, Durchsprache |
| **Specification language** | Spezifikationssprache | | |
| **Stakeholder** | Interesseneigner, Stakeholder | | |
| **Standard** | Norm, Standard | | |

# Part Three: German–English Dictionary
# Teil Drei: Begriffswörterbuch Deutsch–Englisch

| | | | |
|---|---|---|---|
| **Abnahme** | Acceptance | **Datenflussdiagramm** | Dataflow diagram |
| **Abnahmetest** | Acceptance test | **Defekt** | Defect, Bug, Fault |
| **Adäquatheit** (einer Anforderung) | Adequacy (of a requirement) | **Domäne** | Domain |
| | | **Durchsicht** (↑Review) | Review |
| **Akteur** | Actor | **Durchsprache** (↑Walkthrough) | Walkthrough |
| **Aktivitätsdiagramm** | Activity diagram | | |
| **Änderbarkeit** (eines Artefakts) | Changeability (of an artifact) | **Effektivität** | Effectiveness |
| | | **Effizienz** | Efficiency |
| **Änderungsantrag** | Change request | **Eindeutigkeit** (von Anforderungen) | Unambiguity (of requirements) |
| **Anforderung** | Requirement | **Einhaltung** (↑Befolgung) | Compliance |
| **Anforderungsanalyse** | Requirements analysis | | |
| **Anforderungs-analytiker** | Requirements engineer | **Endbenutzer** | End user |
| **Anforderungsart** | Kind of requirement | **Entität** | Entity |
| **Anforderungs-basislinie** | Requirements baseline | **Entity-Relationship Diagramm** | Entity-relationship diagram |
| **Anforderungs-dokument** | Requirements document | **Entity-Relationship Modell** | Entity-relationship model |
| **Anforderungs–ermittlung** | Requirements elicitation | **Entscheidungstabelle** | Decision table |
| **Anforderungs-gewinnung** (↑Anforderungs-ermittlung) | Requirements elicitation | **Feature** (↑Merkmal) | Feature |
| | | **Fehler** | Error |
| | | **Fehlertoleranz** | Fault Tolerance |
| | | **Funktionale Anforderung** | Functional requirement |
| **Anforderungs-ingenieur** | Requirements engineer | | |
| **Anforderungs-management** | Requirements management | **Funktionalität** | Functionality |
| | | **Gegenstand** | Entity |
| **Anforderungsmodell** | Requirements model | **Gesichtspunkt** | Viewpoint |
| **Anforderungsquelle** | Requirements source | **Glossar** | Glossary |
| **Anforderungs-schablone** | Requirements template | **Homonym** | Homonym |
| | | **Inspektion** | Inspection |
| **Anforderungs-spezifikation** | Requirements specification | **Interesseneigner** (↑Stakeholder) | Stakeholder |
| **Anwendungsbereich** | Application domain | **Kardinalität** | Cardinality |
| **Anwendungsfall** | Use case | **Klasse** | Class |
| **Anwendungs-falldiagramm** | Use case diagram | **Klassendiagramm** | Class diagram |
| | | **Klassenmodell** | Class model |
| | | **Komponente** | Component |
| **Artefakt** | Artifact | **Konfiguration** | Configuration |
| **Attribut** | Attribute | **Konformität** (von Anforderungen) | Conformity (of requirements) |
| **Basislinie** | Baseline | | |
| **Befolgung** (↑Einhaltung) | Compliance | **Konsistenz** (von Anforderungen) (↑Widerspruchsfreiheit) | Consistency (of requirements) |
| **Benutzbarkeit** | Usability | | |
| **Benutzer** | User | **Kontext** | Context |
| **Change control board** | Change control board | **Kontextdiagramm** | Context diagram |

| | | | |
|---|---|---|---|
| **Kontextgrenze** | Context boundary | **Satzschablone** | Phrase template |
| **Kontextmodell** | Context model | **Semantik** | Semantics |
| **Korrektheit** | Correctness | **Sequenzdiagramm** | Sequence diagram |
| **Kunde** | Customer | **Sicherheit** (Informationssicherheit) | Security |
| **Lastenheft** | Customer requirements specification | **Sicherheit** (Nutzungssicherheit) | Safety |
| **Leistungsanforderung** | Performance requirement | **Sicht** | View |
| **Lenkungsausschuss** | Steering committee | **Software-Anforderungsspezifikation** | Software requirements specification |
| **Lieferant** | Supplier | **Spezifikation** | Specification |
| **Merkmal** (↑Feature) | Feature | **Spezifikationssprache** | Specification language |
| **Modell** | Model | **Sprache** | Language |
| **Modellierungssprache** | Modeling language | **Standard** (↑Norm) | Standard |
| **Multiplizität** | Multiplicity | **Stakeholder** (↑Interesseneigner) | Stakeholder |
| **Nicht-funktionale Anforderung** | Non-functional requirement | **Standpunkt** (↑Gesichtspunkt) | Viewpoint |
| **Norm** (↑Standard) | Standard | **Statechart** | Statechart |
| **Pflegbarkeit** (↑Wartbarkeit) | Maintainability | **Strukturierte Analyse** | Structured Analysis |
| **Pflichtenheft**[1] | Software requirements specification (also: system requirements specification) | **Synonym** | Synonym |
| | | **Syntax** | Syntax |
| | | **System** | System |
| **Portabilität** (↑Übertragbarkeit) | Portability | **Systemanforderung** | System requirement |
| **Priorität** (einer Anforderung) | Priority (of a requirement) | **System-Anforderungsspezifikation** | System requirements specification |
| **Prototyp** | Prototype | **Systemgrenze** | System boundary |
| **Prozesswort** | Process verb | **Systemkontext** | System context |
| **Prüfbarkeit** (von Anforderungen) | Verifiability (of requirements) | **Systemumfang** | Scope of a system |
| **Prüfung** (von Anforderungen) | Checking (requirements) | **Szenario** | Scenario |
| **Qualität** | Quality | **Teilformal** | Semi-formal |
| **Qualitätsanforderung** | Quality requirement | **Übertragbarkeit** (↑Portabilität) | Portability |
| **Randbedingung** | Constraint | **UML** | UML |
| **Redundanz** | Redundancy | **Use Case** (↑Anwendungsfall) | Use case |
| **Release** | Release | | |
| **Requirements Engineering** | Requirements Engineering | **Use Case Diagramm** | Use case diagram |
| **Review** (↑Durchsicht) | Review | **Validierung** (von Anforderungen) | Validation (of requirements) |
| **Risiko** | Risk | **Verfolgbarkeit** (von Anforderungen) | Traceability (of requirements) |
| | | **Verhaltensmodell** | Behavior model |
| | | **Version** (eines Gegenstands) | Version (of an entity) |
| | | **Vollständigkeit** (von Anforderungen) | Completeness (of requirements) |
| | | **Walkthrough** (↑Durchsprache) | Walkthrough |
| | | **Wartbarkeit** (↑Pflegbarkeit) | Maintainability |

---

[1] *Pflichtenheft* hat im Deutschen mehrere mögliche Bedeutungen: 1. Vom Lieferanten erarbeitete Lösungsvorgaben für ein System, in der Regel auf der Basis eines ↑Lastenhefts; 2. Synonym für ↑Anforderungsspezifikation (in der Regel für ein softwarebasiertes System, erstellt vom Lieferanten); 3. Anforderungsspezifikation unter Einschluss der für den Kunden relevanten Teile des Projektplans. Es gibt kein englisches Wort mit einem vergleichbaren Bedeutungsspektrum. In der Regel ist *Software requirements specification* die angemessenste Übersetzung.

| | | | |
|---|---|---|---|
| **Werkzeug** (im Software Engineering) | Tool (in software engineering) | **Zustandsdiagramm** | State-transition diagram |
| **Widerspruchsfreiheit** (von Anforderungen) (↑Konsistenz) | Consistency (of requirements) | **Zustandsmaschine** | State machine |
| | | **Zuverlässigkeit** | Reliability |
| **Ziel** | Goal | | |
| **Zielmodell** | Goal model | | |

# Sources

As mentioned in the preface, I don't cite sources for individual definitions because I deliberately decided not to compile definitions from various existing sources just by copy-paste, but to carefully re-formulate all definitions consistently and according to today's use. Nevertheless, I want to give credit for some definitions that have been taken verbatim from a source or that are joint work with others. The copyright for cited definitions lies with the authors of the cited work. The copyright for joint work lies jointly with the author of this glossary and the persons mentioned.

| Term | Reference |
| --- | --- |
| Constraint | Joint work with Klaus Pohl, Chris Rupp, and Thorsten Weyer, based on definitions in my course notes on Requirements Engineering I |
| Context boundary | Joint work with Klaus Pohl, Chris Rupp, and Thorsten Weyer |
| Functional requirement | Joint work with Klaus Pohl, Chris Rupp, and Thorsten Weyer |
| Model | Base definition taken from [Pohl and Rupp 2011] |
| Quality requirement | Joint work with Klaus Pohl, Chris Rupp, and Thorsten Weyer, based on definitions in my course notes on Requirements Engineering I |
| Requirement | First part of definition taken from IEEE Std 610.12-1990 |
| Requirements Engineering | Joint work with Klaus Pohl, Chris Rupp, and Thorsten Weyer |
| Requirements specification | Base definition taken from [Pohl and Rupp 2011] |
| Requirements template | Base definition taken from [Pohl and Rupp 2011] |
| Stakeholder | Joint work with Klaus Pohl, Chris Rupp, and Thorsten Weyer |
| System boundary | Joint work with Klaus Pohl, Chris Rupp, and Thorsten Weyer |
| System context | Joint work with Klaus Pohl, Chris Rupp, and Thorsten Weyer |

# References

Gause, D.C., G.M. Weinberg (1989). *Exploring Requirements: Quality before Design.* New York: Dorset House.

Glinz, M. (2010). Course notes on *Informatik II (Modeling)*, *Requirements Engineering I*, and *Software Engineering*. http://www.ifi.uzh.ch/rerg/teaching/

Glinz, M. (2007). On Non-Functional Requirements. *Proceedings of the 15th IEEE International Requirements Engineering Conference*, Delhi, India. 21-26.

Glinz, M., R. Wieringa (2007). Stakeholders in Requirements Engineering (Guest Editors' Introduction). *IEEE Software* **24**, 2. 18-20.

Gotel, O., A. Finkelstein (1994). An Analysis of the Requirements Traceability Problem, *Proceedings of the First International Conference on Requirements Engineering*, Colorado Springs. 94-101.

IEEE (1990). *Standard Glossary of Software Engineering Terminology.* IEEE Std 610.12-1990.

IEEE (1993). *IEEE Recommended Practice for Software Requirements Specifications*. IEEE Standard 830-1993.

IREB (2010). *Certified Professional for Requirements Engineering Foundation Level Syllabus*, Version 2.1. http://www.certified-re.de/en/syllabi/foundation-level.html

ISO/IEC (2001). *Software Engineering—Product Quality—Part 1: Quality Model*. ISO/IEC Standard 9126-1:2001, International Organization for Standardization.

ISO/IEC (2007). *Systems and Software Engineering — Recommended Practice for Architectural Description of Software-Intensive Systems*. ISO/IEC Standard 42010: 2007 ( equal to IEEE Std 1471-2000) International Organization for Standardization.

ISO (2005). *Quality Management Systems—Fundamentals and Vocabulary*. ISO Standard 9000:2005, International Organization for Standardization.

Mylopoulos, J. (2006). *Goal-Oriented Requirements Engineering: Part II.* Presentation slides of keynote talk at the 14th IEEE International Requirements Engineering Conference (RE'06), Minneapolis, USA.

Pohl, K. (2007). *Requirements Engineering: Grundlagen, Prinzipien, Techniken*. Heidelberg: dpunkt.

Pohl, K. (2010). *Requirements Engineering: Fundamentals, Principles, and Techniques*. Berlin-Heidelberg: Springer.

Pohl, K., Rupp, C. (2010). *Basiswissen Requirements Engineering.* 2. Auflage. Heidelberg: dpunkt.

Pohl, K., Rupp, C. (2011). *Requirements Engineering Fundamentals*. Santa Barbara, Ca.: RockyNook.

Robertson, S., Robertson, J. (2006). *Mastering the Requirements Process*. 2nd edition, Addison-Wesley.

Rupp, C. et al. (2009). *Requirements Engineering und –Management: Professionelle, iterative Anforderungs-analyse für die Praxis*. 5. Auflage. München: Hanser.

Stachowiak, H. (1973). *Allgemeine Modelltheorie.* Wien: Springer.

Wikipedia. http://de.wikipedia.org und http://en.wikipedia.org. Visited Jan-Mar 2010.

Zowghi, D., C. Coulin (2005). Requirements Elicitation: A Survey of Techniques, Approaches, and Tools. In A. Aurum, C. Wohlin: *Engineering and Managing Software Requirements*. Berlin: Springer. 19-46.