Appendix A: Glossary of Traceability Terms (v1.0)

Orlena Gotel, Jane Cleland-Huang, Andrea Zisman, Jane Huffman Hayes, Alex Dekhtyar, Patrick Mäder, Alexander Egyed, Paul Grünbacher, Giuliano Antoniol, and Jonathan Maletic

Answer set – A known set of *trace links* derived prior to a *tracing* experiment, usually prepared by system experts.

Artifact – Something that is created or shaped by humans, either directly or indirectly via automation. In software and systems engineering contexts, the term refers to the products of the engineering process. See *trace artifact.*

Artifact type – See *trace artifact type.*

Assisted traceability – See *semi-automated traceability.*

Assisted tracing – See *semi-automated tracing.*

Association – An as yet unspecified connection between a pair of *artifacts*. Where augmented with semantics providing directionality, the *association* becomes traversable and is referred to as a *trace link*.

Atomic trace – A *trace* (noun sense) comprising a single *source artifact*, a single *target artifact* and a single *trace link*.

Attribute – A characteristic or property inherent in or ascribed to something. In software and systems engineering contexts, the term refers to the properties of *artifacts* and their *trace links*. See *trace attribute.*

Automated traceability – The potential for *automated tracing*.

Automated tracing – When *traceability* is established via automated techniques, methods and tools. Currently, it is the decision as to among which *artifacts* to create and maintain *trace links* that is automated.

Backward traceability – The potential for *backward tracing*.

Backward tracing – In software and systems engineering contexts, the term is commonly used when the *tracing* follows antecedent steps in a developmental path, which is not necessarily a chronological path, such as backward from code through design to requirements. Note that the *trace links* themselves could be used in either a *primary* or *reverse trace link direction*, dependent upon the specification of the participating *traces*.

Bidirectional trace link – A term used to refer to the fact that a *trace link* can be used in both a *primary trace link direction* and a *reverse trace link direction*.

Bidirectional traceability – The potential for *bidirectional tracing*.

Bidirectional tracing – When *tracing* can be undertaken in both a *forward* and *backward* direction.

Body of knowledge for traceability – See *Traceability Body of Knowledge (TBOK)*.

Candidate trace link – A potential, as yet unverified, *trace link*.

Center of Excellence for Software Traceability (CoEST) – A *traceability community* initiative. "Our goal is to bring together *traceability* researchers and experts in the field. We hope to encourage research collaborations, assemble a *body of knowledge for traceability*, and develop new technology to meet *tracing* needs." (Hayes et al., 2007.) See: *http://www.coest.org*.

Chained trace – A *trace* (noun sense) comprising multiple *atomic traces* strung in sequence, such that a *target artifact* for one *atomic trace* becomes the *source artifact* for the next *atomic trace*.

Continuous traceability maintenance – The update of impacted *trace links* immediately following changes to *traced artifacts*.

Creating traceability – See *traceability creation*.

Element – A fundamental constituent of a composite entity. In a *traceability* context, the term refers to the fundamental constituents of a *trace* (noun sense). See *trace element*.

Establishing traceability – Enacting those parts of the *traceability process* associated with *traceability creation* and *maintenance*, and in accordance with the *traceability strategy*.

Forward traceability – The potential for *forward tracing*.

Forward tracing – In software and systems engineering contexts, the term is commonly used when the *tracing* follows subsequent steps in a developmental path, which is not necessarily a chronological path, such as forward from requirements through design to code. Note that the *trace links* themselves could be used in either a *primary* or *reverse trace link direction*, dependent upon the specification of the participating *traces*.

Golden standard requirements traceability matrix – See *answer set*.

Grand Challenge of Traceability – A fundamental problem with *traceability* that members of the international research and industrial communities agree deserves attention in order to achieve a revolutionary advance in *traceability practice*. It is a problem with no point solution; its solution involves first understanding and tackling a myriad of underlying challenges, and so will demand the effort of multiple research groups over an extended time period.

Horizontal traceability – The potential for *horizontal tracing*.

Horizontal tracing – In software and systems engineering contexts, the term is commonly used when *tracing artifacts* at the same level of abstraction, such as: (i) *traces* between all the requirements created by "Mary", (ii) *traces* between requirements that are concerned with the performance of the system, or (iii)

traces between versions of a particular requirement at different moments in time. *Horizontal tracing* may employ both *forward tracing* and *backward tracing*.

Just in time tracing (JITT) – See *reactive tracing.*

Link – See *trace link.*

Link base – See *link set.*

Link semantics – The purpose or meaning of the *trace link*. The *link semantics* are generally specified in the *trace link type*, which is a broader term that may also capture other details regarding the nature of the *trace link*, such as how the *trace link* was created.

Link set – The totality of the *trace links* on a project.

Link type – See *trace link type.*

Maintaining traceability – See *traceability maintenance.*

Manual traceability – The potential for *manual tracing.*

Manual tracing – When *traceability* is established by the activities of a human *tracer*. This includes *traceability creation* and *maintenance* using the drag and drop methods that are commonly found in current *requirements management tools.*

Obsolete trace link – A pre-existing, and previously verified, *trace link* that is no longer valid.

On-demand traceability maintenance – A dedicated and overall update of the *trace set* (in whole or in part), generally in response to some explicit trigger and in preparation for an upcoming *traceability use.*

Post-requirements (specification) traceability – The potential for *post-requirements (specification) tracing.*

Post-requirements (specification) tracing – In software and systems engineering contexts, the term is commonly used to refer to those *traces* derived from or grounded in the requirements, and hence the *traceability* explicates the requirements' deployment process. The *tracing* is, therefore, forward from requirements and back to requirements. *Post-requirements (specification) tracing* may employ *forward tracing*, *backward tracing*, *horizontal tracing* and *vertical tracing.*

Pre-requirements (specification) tracing – The potential for *pre-requirements (specification) tracing.*

Pre-requirements (specification) traceability – In software and systems engineering contexts, the term is commonly used to refer to those *traces* that show the derivation of the requirements from their original sources, and hence the *traceability* explicates the requirements' production process. The *tracing* is, therefore, forward to requirements and back from requirements. *Pre-requirements (specification) tracing* may employ *forward tracing*, *backward tracing*, *horizontal tracing* and *vertical tracing.*

Primary trace link direction – When a *trace link* is traversed from its specified *source artifact* to its specified *target artifact*, it is being used in the primary direction as specified. Where *link semantics* are provided, they provide for a way to "read" the traversal (e.g., A implements B).

Proactive tracing – Initiating *trace capture* without explicit response to a stimulus to do so (i.e., *traces* are created in the background). Compare with *reactive tracing*.

Prospective tracing – See *trace capture*.

Reactive tracing* – Responding to a stimulus to initiate *trace capture* (i.e., *traces* are created on demand). Compare with *proactive tracing*.

Ready-to-use traceability – Where previously established *trace links* are maintained as a project evolves, generally in compliance with a *traceability information model (TIM)*, so that the *traceability* on a project is always ready to be used according to the intentions for a project. This may combine *continuous* and *on-demand traceability maintenance* as appropriate.

Reference set – See *answer set*.

Requirements management – The activity concerned with the effective control of information related to stakeholder, system and software requirements and, in particular, the preservation of the integrity of that information for the life of the system and with respect to changes in the system and its environment. *Requirements management* depends upon *requirements traceability* as its enabling mechanism.

Requirements management tools – Tools that support *requirements management*.

Requirements traceability – "The ability to describe and follow the life of a requirement in both a forwards and backwards direction (i.e., from its origins, through its development and specification, to its subsequent deployment and use, and through periods of ongoing refinement and iteration in any of these phases)." (Gotel and Finkelstein, 1994.)

Requirements traceability matrix (RTM) – See *traceability matrix*.

Retrospective tracing – See *trace recovery*.

Reverse trace link direction – When a *trace link* is traversed from its specified *target artifact* to its specified *source artifact*, it is being used in the reverse direction to its specification. The *link semantics* may no longer be valid, so a change from active to passive voice (or vice-versa) is generally required (e.g., if A replaces B then B is replaced by A).

Semi-automated traceability* – The potential for *semi-automated tracing*.

Semi-automated tracing* – When *traceability* is established via a combination of automated techniques, methods, tools and human activities. For example, automated techniques may suggest *candidate trace links* or *suspect trace links* and then the human *tracer* may be prompted to verify them.

Software traceability – See *requirements traceability*, extending the definition to encompass and interrelate any uniquely identifiable software engineering *artifact* to any other.

Source artifact* – The *artifact* from which a *trace* originates.

Stakeholder requirements for traceability – *Stakeholder requirements for traceability* comprise two parts: (i) why end users (i.e., people, organizations, etc.) need *traceability*; and (ii) what *tracers* need in order to establish and use this *traceability*. The latter form part of the *system requirements for traceability*.

Suspect trace link – A pre-existing, and previously verified, *trace link* that may no longer be valid.

System requirements for traceability – What the *traceability solution* needs to do to fulfill the *stakeholder requirements for traceability*. Note that the agent (human or automated) that establishes the *traceability* is part of the *traceability solution*.

Systems traceability – See *requirements traceability*, extending the definition to encompass and interrelate any uniquely identifiable systems engineering *artifact* to a broad range of systems-level components, such as people, processes and hardware models.

Target artifact* – The *artifact* at the destination of a *trace*.

Trace (Noun) – A specified triplet of *elements* comprising: a *source artifact*, a *target artifact* and a *trace link* associating the two *artifacts*. Where more than two *artifacts* are associated by a *trace link*, such as the aggregation of two *artifacts* linked to a third *artifact*, the aggregated *artifacts* are treated as a single *trace artifact*. The term applies, more generally, to both *traces* that are *atomic* in nature (i.e., singular) or *chained* in some way (i.e., plural).

Trace (Verb) – The act of following a *trace link* from a *source artifact* to a *target artifact* (*primary trace link direction*) or vice-versa (*reverse trace link direction*). See *tracing*.

Trace acquisition – See *trace creation.*

Trace artifact* – A *traceable* unit of data (e.g., a single requirement, a cluster of requirements, a UML class, a UML class operation, a Java class or even a person). A *trace artifact* is one of the *trace elements* and is qualified as either a *source artifact* or as a *target artifact* when it participates in a *trace*. The size of the *traceable* unit of data defines the *granularity* of the related *trace*.

Trace artifact type* – A label that characterizes those *trace artifacts* that have the same or a similar structure (syntax) and/or purpose (semantics). For example, requirements, design and test cases may be distinct *artifact types*.

Trace asset – See *trace element.*

Trace attribute* – Additional information (i.e., meta-data) that characterizes properties of the *trace* or of its individual *trace elements*, such as a date and time stamp of the *trace's creation* or the *trace link type*.

Trace capture* – A particular approach to *trace creation* that implies the creation of *trace links* concurrently with the creation of the *artifacts* that they associate. These *trace links* may be created automatically or semi-automatically using tools.

Trace creation* – The activity of *creating* a single *trace*, associating two *artifacts* via a *trace link*. The *trace link* may be created manually, automatically using tools or semi-automatically using some combination of tool and manual input. The terms of *trace capture*, *trace recovery* and *trace retrieval* lend connotations as to when a *trace link* is created, along with the technique used to create the *trace link* in the case of *trace retrieval*.

Trace data – See *trace element.*

Trace element* – Used to refer to either one of the triplets comprising a *trace*: a *source artifact*, a *target artifact* or a *trace link*.

Trace generation – A particular approach to *trace creation* that implies that the *trace links* are created automatically or semi-automatically using tools.

Trace granularity – The level of detail at which a *trace* is recorded and performed. The granularity of a *trace* is defined by the granularity of the *source artifact* and the *target artifact.*

Trace life cycle – A conceptual model that describes the series of activities involved in the life of a single *trace*, from initial conception, through creation, maintenance and use, through to eventual retirement. This is the *traceability process* from the perspective of a single *trace* flowing through the *traceability process*.

Trace link* – A specified *association* between a pair of *artifacts*, one comprising the *source artifact* and one comprising the *target artifact*. The *trace link* is one of the *trace elements*. It may or may not be annotated to include information such as the *link type* and other semantic *attributes*. This definition of *trace link* implies that the *link* has a *primary trace link direction* for *tracing*. In practice, every *trace link* can be traversed in two directions (i.e., if A tests B then B is tested by A), so the *link* also has a *reverse trace link direction* for *tracing*. The *trace link* is effectively *bidirectional*. Where no concept of directionality is given or implied, it is referred to solely as an *association*.

Trace link type* – A label that characterizes those *trace links* that have the same or similar structure (syntax) and/or purpose (semantics). For example, "implements", "tests", "refines" and "replaces" may be distinct *trace link types*.

Trace maintenance – Those activities associated with updating a single pre-existing *trace* as changes are made to the *traced artifacts* and the *traceability* evolves, *creating* new *traces* where needed to keep the *traceability* relevant and up to date.

Trace precision – A commonly used metric in *automated tracing* that applies to represent the fraction of retrieved *trace links* that are relevant. It is computed as: Precision = (Relevant *Links* ∩ Retrieved *Links*) / Retrieved *Links*.

Trace quality – A measurable property of a single *trace* at a particular point in time on a project, such as a confidence score depicting its correctness.

Trace query – A term often used in the process of generating or vetting *trace links*, where one high level *element* is regarded as the *trace query* for searching into an *artifact* collection to find *trace links* (as distinguished from *traceability-related queries*).

Trace recall – A commonly used metric in *automated tracing* that applies to represent the fraction of relevant *trace links* that are retrieved. It is computed as: Recall = (Relevant *Links* ∩ Retrieved *Links*) / Relevant *Links*.

Trace record – Persistent information that registers the triplet of *trace elements* constituting a *trace* and is subject to version control. The *trace record* can also refer to the entire *trace set*.

Trace recovery* – A particular approach to *trace creation* that implies the creation of *trace links* after the *artifacts* that they associate have been generated and manipulated. These *trace links* may be created automatically or

semi-automatically using tools. The term can be construed to infer that the *trace link* previously existed but now is lost.

Trace relation – All the *trace links* created between two sets of specified *trace artifact types*. The *trace relation* is the instantiation of the *trace relationship* and hence is a collection of *traces*. For example, the *trace relation* would be the actual *trace links* that associate the instances of requirements *artifacts* with the instances of test case *artifacts* on a project. The *trace relation* is commonly recorded within a *traceability matrix*.

Trace relationship – An abstract definition of a permissible *trace relation* on a project (i.e., *source artifact type*, *target artifact type* and *trace link types*), as typically expressed within a *traceability information model (TIM)*. Note that the *trace links* of the instances of the two *artifact types* may not necessarily have the same *trace link type*.

Trace retrieval – A particular approach to *trace creation* where information retrieval methods are used to dynamically create a *trace link*. This approach can be used for both *trace capture* and *trace recovery*.

Trace set – The totality of the *traces* on a project.

Trace sink artifact – See *target artifact*.

Trace source artifact – See *source artifact*.

Trace target artifact – See *target artifact*.

Trace use – Those activities associated with putting a single *trace* to use to support various software and systems engineering activities and tasks.

Traceability – The potential for *traces* to be established and used. *Traceability* (i.e., *trace* "ability") is thereby an *attribute* of an *artifact* or of a collection of *artifacts*. Where there is *traceability*, *tracing* can be undertaken and the specified *artifacts* should be *traceable*.

Traceability analyses – The analyses that can be undertaken following *traceability-related queries*.

Traceability benchmark – A standard measure or test against which approaches to various aspects of the *traceability process* can be evaluated and compared.

Traceability benchmark data – Datasets that contain two or more *artifact types* and validated *traceability matrices*, the latter serving as *answer sets* (i.e., reference sets), for evaluating experimental results.

Traceability Body of Knowledge (TBOK)* – A proposed resource for the *traceability community*, containing *traceability benchmarks*, good *traceability practices*, *traceability* experience reports, etc.

Traceability challenge – A significant problem with *traceability* that members of the international research and industrial communities agree deserves attention in order to achieve advances in *traceability practice*.

Traceability community – Those people who are *establishing* and *using traceability* in practice, or have done so in the past or intend to do so in the future. Also, those people who are active in *traceability* research or in one of its many interrelated areas.

Traceability configuration management – The process of identifying, defining, recording and reporting on *traces* as configuration items, also controlling both

the release of *traces* for *traceability use* and the changes that occur during *traceability maintenance*. *Traceability configuration management* depends upon *traceability version control*.

Traceability creation – The general activity of associating two (or more) *artifacts*, by providing *trace links* between them, for *tracing* purposes. Note that this could be done manually, automatically or semi-automatically, and additional annotations can be provided as desired to characterize *attributes* of the *traces*.

Traceability decay – The gradual disintegration and break down of the *traceability* on a project. This tends to result following ongoing *traceability evolution*.

Traceability-enabled activities and tasks – Those software and systems engineering activities and tasks that *traceability* supports, such as verification and validation, impact analysis and change management.

Traceability-enabled tasks and activities – See *traceability-enabled activities and tasks*.

Traceability end use – See *traceability use.*

Traceability end user – The human or system engaged in *traceability use*.

Traceability entropy – The inevitable and steady deterioration of *traceability* as a result of *traceability decay*.

Traceability evolution – The gradual change of the *traceability* on a project. It generally refers to the tendency for pre-existing *traces* to become outdated and/or obsolete over time as changes are made to the *traced artifacts*, unless the *traceability* is maintained sufficiently. Ongoing deterioration of the *traceability* may lead to *traceability decay*.

Traceability graph – A representation of the *trace set*, with *trace artifacts* depicted as nodes and *trace links* depicted as edges.

Traceability history – A record of the *traceability evolution* and the associated *traceability maintenance* that has taken place on a project.

Traceability information – Any *traceability*-related data, such as *traceability information models*, *trace artifacts*, *trace links* and other *traceability work products*.

Traceability information model (TIM)* – A graph defining the permissible *trace artifact types,* the permissible *trace link types* and the permissible *trace relationships* on a project, in order to address the anticipated *traceability-related queries* and *traceability-enabled activities and tasks*. The *TIM* is an abstract expression of the intended *traceability* for a project. The *TIM* may also capture additional information such as: the cardinality of the *trace artifacts* associated through a *trace link*, the *primary trace link direction*, the purpose of the *trace link* (i.e., the *link semantics*), the location of the *trace artifacts*, the *tracer* responsible for *creating* and *maintaining* the *trace link*, etc. (See Mäder et al. (2009) for more detail.)

Traceability intent – See *traceability information model (TIM).*

Traceability life cycle – A conceptual model that describes the series of activities associated with a full end-to-end *traceability process*.

Traceability link – A term often used in place of *trace link*. Arguably, while *traceability link* captures the enabling role of the *link* for *traceability* purposes, *trace link* emphasizes the fact that the *link* is a primary *element* of a *trace*.

Traceability link document – A document depicting *traces*, showing which pairs of *trace artifacts* are associated via *trace links*.

Traceability maintenance – Those activities associated with updating pre-existing *traces* as changes are made to the *traced artifacts* and the *traceability* evolves, *creating* new *traces* where needed to keep the *traceability* relevant and up to date.

Traceability management – Those activities associated with providing the control necessary to keep the *stakeholder* and *system requirements for traceability* and the *traceability solution* up to date during the life of a project. *Traceability management* is a fundamental part of *traceability strategy*.

Traceability matrix – A matrix recording the *traces* comprising a *trace relation*, showing which pairs of *trace artifacts* are associated via *trace links*.

Traceability meta-model – Defined constructs and rules related to the *trace artifact types* and *trace link types* for building *traceability information models (TIMs)*.

Traceability method – A prescription of how to perform a collection of *traceability practices*, integrating *traceability techniques* with guidance as to their application and sequencing.

Traceability metric – A measure for some property or aspect of the *traceability process*, either quantitative or qualitative, such as *trace recall* and *trace precision* for *trace recovery*.

Traceability model – See *traceability information model (TIM)*.

Traceability network – A *traceability graph* in which the directionality of the *trace links* is expressed (i.e., the *artifacts* are depicted as ordered pairs) and where the *trace links* are potentially weighted in some manner.

Traceability planning – Those activities associated with determining the *stakeholder* and *system requirements for traceability* and designing a suitable *traceability solution*. *Traceability planning* is a fundamental part of *traceability strategy*.

Traceability policy – Agreed principles and guidelines for *establishing* and *using traceability* in practice.

Traceability practices – Those actions and activities associated with *planning*, *managing*, *creating*, *maintaining* and *using traceability*.

Traceability process – An instance of a *traceability process model* defining the particular series of activities to be employed to establish *traceability* and render it usable for a particular project, along with a description of the responsibilities and resourcing required to undertake them, as well as their inputs and outputs. The *traceability process* defines how to undertake *traceability strategy*, *traceability creation*, *traceability maintenance* and *traceability use.*

Traceability process improvement – The activity of defining, analyzing and improving upon an existing *traceability process*.

Traceability process model – An abstract description of the series of activities that serve to establish *traceability* and render it usable, along with a description of the typical responsibilities and resourcing required to undertake them, as well as their inputs and outputs. Distinctive steps of the process comprise *traceability strategy*, *traceability creation*, *traceability maintenance* and *traceability use*.

Traceability product – See *traceability work products.*

Traceability quality – A measurable property of the overall *traceability* at a particular point in time on a project, such as a confidence score depicting its overall correctness, accuracy, precision, completeness, consistency, timeliness, usefulness, etc.

Traceability quality assessment – The activity of assessing the *traceability quality* on a project.

Traceability quality assurance – The activity of assuring that defined standards and processes for *traceability* are appropriate and applied on a project.

Traceability quality attribute – A measurable property of a single *trace link* or of a group of *trace links*, such as a confidence score depicting the likelihood that a recovered *candidate trace link* is correct or the usefulness of a particular *trace link* over time.

Traceability reference model – See *traceability information model (TIM).*

Traceability-related queries – Those questions that a software or systems engineer may pose to which *traceability* can help to retrieve answers, such as the percentage of the specified requirements that are *traceable* to test cases and the existence of any requirements that are not *traced* through to design *artifacts*.

Traceability scheme – See *traceability information model (TIM).*

Traceability solution* – The *traceability information model (TIM)* and *traceability process*, as defined, designed and implemented for a particular project situation, along with any associated *traceability tooling*. The *traceability solution* is determined as a core part of the *traceability strategy*.

Traceability stakeholders – Those roles (i.e., people or systems) that have something to gain or something to lose from either having or not having *traceability* on a project.

Traceability standard – Mandatory practices and other conventions employed and enforced to prescribe a disciplined and uniform approach to *traceability*, generally written down and formed by consensus.

Traceability strategy – Those decisions made in order to determine the *stakeholder* and *system requirements for traceability* and to design a suitable *traceability solution*, and for providing the control necessary to keep these requirements and solutions relevant and effective during the life of a project. *Traceability strategy* comprises *traceability planning* and *traceability management* activities.

Traceability system – See *traceability solution.*

Traceability technique – A prescription of how to perform a single *traceability practice*, such as *traceability creation*, along with a description of how to represent its *traceability work products*.

Traceability tool – Any instrument or device that serves to assist or automate any part of the *traceability process*.

Traceability use* – Those activities associated with putting *traces* to use to support various software and systems engineering activities and tasks, such as verification and validation, impact analysis and change management.

Traceability version control – *Tracking* changes to a particular *trace* over time. Each time a *trace* is changed in some way, a new version of the *trace* is effectively generated. This provides for an audit trail, and for parallel development and rollback possibilities.

Traceability work products* – Those *artifacts* produced as a result of *planning*, *managing*, *creating*, *maintaining* and *using traceability*, including the *trace set*.

Traceable – The potential for *artifacts* to be accessed and retrieved by following *trace links* (i.e., by undertaking *tracing*). *Traceable* (i.e., *trace* "able") is thereby an *attribute* of an *artifact* or of a collection of *artifacts*.

Traced – The *artifacts* that have been accessed by *tracing*, and so by having followed *trace links*.

TraceLab – A visual experimental workbench for designing and executing *traceability* experiments, providing *traceability* researchers with access to algorithms, datasets, experimental frameworks and benchmarking tools. *TraceLab* is a major component of the *Tracy project*.

Tracer – The agent engaged in the activity of *tracing*, where the agent can be a human or supporting tool.

Tracing – The activity of either *establishing* or *using traces*.

Tracing activity or task – A discrete and identifiable unit of work associated with the broader activity of *tracing*; an atomic activity of the *traceability process*.

Tracing benchmark – A clearly defined *tracing task*, with associated data sets and metrics that have been agreed upon by the *traceability community*, and which is used to evaluate different *traceability techniques* and *methods* comparatively.

Tracing contest – A clearly defined *tracing task* that has been identified by the *traceability community* as a critical *traceability practice* that warrants *traceability benchmarking*.

Tracing task or activity – See *tracing activity or task.*

Tracking – In software and systems engineering contexts, the term commonly applies to the act or process of following requirements and depends upon *requirements traceability*.

Tracy project – A National Science Foundation funded project designed to instrument the *traceability* research community, and to develop tools for facilitating the transfer of technology to industry and government organizations (Cleland-Huang et al., 2011).

True requirements traceability matrix – See *answer set.*

Using traceability – Enacting those parts of the *traceability process* associated with *traceability use*.

Value-based traceability – An approach to *traceability* that actively seeks to create, manage and measure either the monetary worth or utility worth of *traceability* on a project.

Vertical traceability – The potential for *vertical tracing*.

Vertical tracing – In software and systems engineering contexts, the term is commonly used when *tracing artifacts* at differing levels of abstraction so as to accommodate life cycle-wide or end-to-end *traceability*, such as from requirements to code. *Vertical tracing* may employ both *forward tracing* and *backward tracing*.

References

Cleland-Huang, J., Czauderna, A., Dekhtyar, A., Gotel, O., Huffman Hayes, J., Keenan, E., Leach, G., Maletic, J., Poshyvanyk, D., Shin, Y., Zisman, A., Antoniol, G., Berenbach, B., Egyed, A., Maeder, P.: Grand challenges, benchmarks, and TraceLab: Developing infrastructure for the software traceability research community. In: Proceedings of the 6th International Workshop on Traceability in Emerging Forms of Software Engineering, Honolulu, Hawaii, USA, 23 May 2011.

Gotel, O., Finkelstein, A.: An analysis of the requirements traceability problem. In: Proceedings of the 1st IEEE International Conference on Requirements Engineering, Colorado Springs, CO, USA, 18–22 Apr 1994, pp. 94–101.

Hayes, J.H., Dekhtyar, A., Cleland-Huang, J.: Charter (Business Plan) for the center of excellence for traceability. COET-CBP-07-02-1.0, 15 Feb 2007. (http://www.traceabilitycenter.org/).

Mäder, P., Gotel, O., Philippow, I.: Getting back to basics: Promoting the use of a traceability information model in practice. In: Proceedings of the 5th International Workshop on Traceability in Emerging Forms of Software Engineering, Vancouver, Canada, 18 May 2009.