The Grand Challenge of Traceability (v1.0)

Center of Excellence for Software Traceability
Technical Report #CoEST-2011-001
June 14, 2011

Orlena Gotel[1], Jane Cleland-Huang[2], Jane Huffman Hayes[3], Andrea Zisman[4], Alexander Egyed[5], Paul Grünbacher[6], Alex Dekhtyar[7], Giulio Antoniol[8] and Jonathan Maletic[9]

Abstract This technical report offers a vision for traceability in software and systems engineering and outlines eight challenges that need to be addressed in order to achieve it. One of these challenges is referred to as the *grand challenge of traceability* because making traceability *ubiquitous* in software and systems development (traceability challenge eight) demands progress with all seven other challenges. A model of a generic traceability process is used as a framework through which the goals and requirements of each challenge are expressed. For each requirement, the current status of the traceability research and practice is summarized, and areas of promise are highlighted. This systematic analysis is used to articulate eight major research themes for the traceability community, along with a number of underlying research topics and positive adoption practices for industry. This work is a snapshot of an ongoing and collaborative effort between traceability researchers and practitioners within the Center of Excellence for Software Traceability (CoEST)[10]. It is a major update to the draft Problem Statement and Grand Challenges document[11], and is intended to form a structured agenda for traceability research and practice, a basis for classifying research contributions and a means to track progress in the field.

1 Introduction

As software systems permeate our society, we must entrust many of them with the lives of everyday people on a daily basis. For example: a commuter on a train trusts that the switching software correctly routes the trains, an airline passenger trusts that the developers of air traffic control software and aviation

[1] Independent Researcher, New York City, USA (olly@gotel.net)
[2] De Paul University, Chicago, USA (jhuang@cs.depaul.edu)
[3] University of Kentucky, Kentucky, USA (hayes@cs.uky.edu)
[4] City University London, London, UK (a.zisman@soi.city.ac.uk)
[5] Johannes Kepler University Linz, Austria (alexander.egyed@jku.at)
[6] Johannes Kepler University Linz, Austria (paul.gruenbacher@jku.at)
[7] Cal Poly State University San Luis Obispo, California, USA (dekhtyar@calpoly.edu)
[8] École Polytechnique de Montréal, Québec, Canada (giulio.antoniol@gmail.com)
[9] Kent State University, Ohio, USA (jmaletic@cs.kent.edu)
[10] *http://www.coest.org*
[11] Cleland-Huang, J., Hayes J.H. and Dekhtyar, A. (Eds.) Center of Excellence for Traceability: Problem Statements and Grand Challenges (v0.1). Center of Excellence for Traceability Technical Report COET-GCT-06-01-0.9, 10 September 2006.

flight control software have built the system correctly, the grocery shopper purchases produce that they trust has been found to be safe and can be tracked back to the farm using software developed to U.S. Food and Drug Administration (FDA) standards, and patients in a hospital are monitored remotely by software systems that many parties trust will work as intended. The ability to attain a requisite level of trust in these everyday examples is enabled through some form of traceability.

Requirements traceability, defined as "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 on-going refinement and iteration in any of these phases)" (Gotel and Finkelstein 1994) is a critical element of any rigorous software and systems development process. For example, the U.S. FDA states that traceability analysis must be used to verify that the software design implements all of the specified software requirements, that all aspects of the design are traceable to software requirements, and that all code is linked to established specifications and established test procedures (FDA 2002). Similarly, the U.S. Federal Aviation Administration (FAA) has established DO-178B (FAA 1992) as the accepted means of certifying all new aviation software, and this standard specifies that at each and every stage of development "software developers must be able to demonstrate traceability of designs against requirements." Software process improvement standards that are being adopted by many organizations, such as the Capability Maturity Model Integration (CMMI Product Team 2010), require similar traceability practices.

Although there have been significant advances since the early processes and tools to support traceability were introduced in the 1970s (Pierce 1978), it is unfortunate that there is still almost universal failure across both industry and government projects to implement successful and cost-effective traceability (Egyed et al. 2007). For example, one global corporation working toward achieving CMMI level-three compliance was thwarted in this plan primarily because it was unable to successfully meet the traceability requirements for its legacy software products. In another organization governed by the U.S. FAA, developers of a software control system for a well-known airplane struggled to trace each line of code back to requirements and were finally able to accomplish this only through reverse engineering a large number of requirements[12]. These difficulties have been broadly attributed to problems associated with creating, maintaining and using requirements traceability matrices and other enabling techniques, and also attributed to the perception by many developers that the effort of establishing traceability exceeds the benefits it returns (Gotel and Finkelstein 1994; Lindvall and Sandahl 1996; Bianchi et al. 2000; Ramesh and Jarke 2001; Arkley and Riddle 2005).

The challenges of traceability are significant; however, the payoffs for getting it right are also considerable. Over the past two decades, traceability researchers have been systematically addressing the challenges in an attempt to alleviate the traceability problem experienced by practitioners, and to better understand how to create and maintain cost-effective, accurate and meaningful traceability that is fit-for-purpose. Because of the difficulty in accomplishing these goals, a number of international researchers gathered in a series of two workshops funded by NASA and the NSF (respectively held at NASA's IV&V facility in the Summer of 2006, and in Lexington, Kentucky in the Spring of 2007) with the specific intention of determining the state of the practice and research in traceability, and of identifying the significant challenges that need to be addressed. The participants represented academic, government, and industrial researchers and practitioners, and they brought a wealth of experience to the working sessions. This series resulted in the creation of a draft Problem Statement and Grand Challenges (v0.1) document (Cleland-Huang et al. 2006).

This technical report follows on from these workshop discussions and draft document, and it is a community effort among members of the Center of Excellence for Software Traceability. It is a reformulation of the material so as to give grounding, cohesion and structure to the challenges, and to articulate a single grand challenge for traceability as opposed to forty, along with a smaller set of supporting challenges[13].

[12] Both of these accounts were provided first hand to one of the authors of this technical report.

[13] A traceability matrix, one that maps this new reformulation of The Grand Challenge of Traceability (v1.0) to the draft Problem Statement and Grand Challenges (v0.1) document (Cleland-Huang et al. 2006), is provided in Figure 6 of Section 12 of this technical report.

The technical report first presents a vision of what traceability makes possible twenty-five years into the future, by describing a hypothetical software and systems development scenario in 2035, and then outlines the assumptions that are necessary to make this vision a reality. These assumptions constitute the revised and updated set of traceability challenges, and they are eight crosscutting concerns – traceability that is purposed, cost-effective, configurable, trusted, scalable, portable, valued and ubiquitous. The last challenge is elevated to the status of the *grand challenge of traceability* since it demands progress with the other seven. The objective of this reformulation is to provide a structured and motivated research agenda for the traceability community, and a basis upon which to classify and track this research going forward. It therefore highlights eight major research themes to tackle the challenges and delineates their underlying research topics.

The technical report is a complement to existing survey work in the area, notably two comprehensive surveys of the traceability landscape (von Knethen and Paech 2002; Winkler and von Pilgrim 2010), as well as more focal surveys on traceability relations (Spanoudakis and Zisman 2005) and requirements interdependencies (Dahlstedt and Persson 2005).

The technical report is organized as follows. Section 2 presents a traceability vision for 2035 and summarizes the traceability assumptions underlying this vision. These assumptions form the eight traceability challenges. Section 3 describes the framework that was used to explore each of the challenges in more detail, and to derive the major research theme associated with each challenge and its underlying research topics. Sections 4 through 10 present the first seven challenges of traceability in turn – traceability that is purposed, cost-effective, configurable, trusted, scalable, portable and valued. Section 11 presents the eighth and grand challenge of traceability – traceability that is ubiquitous. Section 12 explains the approach to evaluation that is in progress and the intended future use of the traceability challenges by researchers and practitioners. Section 13 concludes, and reiterates the challenges and major research themes for the traceability community.

2 Traceability Vision

The vision for traceability revolves around the software and systems development practice that traceability will help to make possible in the year 2035: the problems traceability solves, the questions it answers, and the overall software and systems engineering experience it enables. Given that there are likely to be many concomitant advances in the processes and technologies that are used for software and systems development over the next fifteen years, this vision is grounded in what is envisaged will be a typical working environment in 2035. A Utopian scenario from this future is outlined in Section 2.1, the traceability it demands is summarized in Section 2.2 and the assumptions needed to achieve this traceability are elaborated in Section 2.3.

2.1 Utopian Traceability Scenario — Vestigia Sine Lacrimis[14]

The software systems engineer highlighted the five key stakeholder types that she knew were interested in the new flying solar car for which she was developing the controller software. She dragged their avatars into the requirements task area of her application lifecycle tool with a wave of her pointer finger. Three flashing red alerts appeared:

- `One potential stakeholder type is missing. The impact of their exclusion or inclusion has been analyzed and the results are ready to examine.`

[14] ***Tracing without tears*** – with thanks to Dr. Robert Natelson for the Latin version of this motto (*http://www.umt.edu/law/faculty/natelson.htm*).

- `High priority requirement 55 of stakeholder type 'Police officer' conflicts with high priority requirement 33 of stakeholder type 'disabled citizen'. Stakeholder representatives have been identified and the resolution process is ready to proceed.`
- `The software demands safety certification. Policy regulations have been retrieved and safety requirements have been determined from related systems in the requirements knowledgebase. Confirm to inspect and integrate.`

"I overlooked all that," she muttered as she pulled up the impact analyzer, conflict resolver and requirements integrator all with a snap of her left hand. A few minutes later, green check marks then appeared with the message:

`All identified requirements have been negotiated and validated with relevant parties. There are no current conflicts, inconsistencies or known omissions, and change management procedures have been established for this requirements baseline. Prioritized requirements with associated test cases are now ready for design and initial architectural options have been retrieved.`

The engineer said aloud: "Let's see the options then," and the design process engaged. A series of questions then appeared to the engineer:

- `Is usability more important than reliability?`
- `Is reliability more important than maintainability?`
- …

The engineer worked through the design goal parameters diligently, pulling up visual design aids and assessing the requirements change impact as needed. Having balanced the design attributes with cost parameters as the requirements evolved further, the engineer shipped off the results to the hardware analyst and other specialists to ensure that there were no lurking issues before proceeding further.

Eventually, working in this manner, the engineer had a fully tested software release ready to integrate into the flying solar car system. With the latest set of requirements verified, the necessary safety certificate was issued. After integration, system testing and launch, the engineer moved on to focus on her next project.

A few days later, the flying solar car project appeared on the engineer's pursetop with this note:

`A new stakeholder requirement has been identified for project flying solar car following end-user feedback. Please review the impact of this addition and of an inconsistency that has been identified if this is to be accommodated.`

The engineer clicked on the warning message and projected a rendering of the relationships between the new requirement request and the existing requirements, design, code and tests. Walking through the virtual project environment, the engineer could see that a similar requirement already existed due to the color of the requested requirement's visual path and that of an existing requirement. After a discussion with holographic avatars representing the stakeholders affected, the engineer pressed the 'dupe requirement' icon with her pinkie finger. Confident that the detailed rationale underlying this decision from the virtual discussion would be assembled and sent to the project manager and requesting end-users, and added into the requirement's record for future reference, she moved back to concentrate on her current project.

Eleven months after the deployment of the software and first production run of the flying solar car an alert arrived on the engineer's prototype smart cashmere sweater sleeve:

`ALERT! The license for the navigation software used by the flying solar car project expires in one month. Renew at $22 million per annum or substitute with one of the following new software services: (a) Nav-U-Like at $11.5 million or (b) Never-Get-Lost at $11.75 million. No negative impact of either code substitution has been identified during simulation and a benefit is projected for each option. Option (a)`

```
implements a requirement that would address open bug report 686 of priority 2, and
requires a small design change estimated to take Bob one working week to fully inte-
grate and test. An analysis of the multimedia materials accompanying option (b) in-
dicates that it satisfies requirements that align with a forthcoming change to world
policies on open skies flight that is scheduled to take effect in three months time
and negate our current safety certificate. The impacted components will take the
full team in Johannesburg two weeks to re-align the software and re-verify the re-
quirements.
```

"Let's plan ahead and go with Never-Get-Lost," she decides as she taps option (b) on her sweater sleeve.

2.2 Traceability in 2035

In 2035, traceability will be purely in the background and simply expected to be there. It will be accurate and trusted by all project stakeholders. Traceability will be seamless to software and systems engineering tasks, and something that underlies many of the techniques and technologies that engineers use habitually. With the disappearance of traceability as a primary concern, the engineer and other project stakeholders will be free to focus on those activities and decisions that utilize their skills and knowledge fully.

Traceability will facilitate tasks in all phases of the software and systems engineering lifecycle, providing for both productivity and quality gains. In particular, it will help with the definition of requirements through reuse at the requirements level, retrieving associated design, code and test cases, along with all the underpinning traceability. It will also help to identify services to satisfy those requirements and to monitor the violations of service-centric systems. It will help to discover discrepancies and inconsistencies in requirements perspectives by identifying connections between disparate requirements, in real-time, by following their trace links to assess the implications. It will also help to assess requirements completeness and satisfaction, and is the mechanism through which certificates of assurance will be issued.

In summary, traceability will be the thread that weaves data together on a project to tell a myriad of stories, from the rationale underlying decisions through to the underlying social network that came together to make these decisions and is, therefore, best able to change them. Traceability will be completely requirements-driven in 2035.

2.3 Assumptions of the Vision -> Traceability Challenges

To achieve this vision of traceability, advances will be required in a number of areas, ranging from everyday communication devices and visual displays through to the manner in which requirements are described and organized. Based upon progress over the past twenty-five years, it is likely that the technologies mentioned in the Utopian scenario will be historic by 2035, but the changes demanded in software and systems engineering practice will remain ambitious. The assumptions demanded of the traceability practice are highlighted below.

To provide for the level of engineering support envisaged, the results of the traceability must be amenable to use and fully trusted, echoing the theme of the examples provided in the introduction to this technical report. The starting point for securing this trust will be buy-in, accompanied by accurate and up to date underlying data to trace, along with timely and meaningful linkages between these data. Much will depend upon the quality of these data, be they business goals, requirements, design ideas or code, whatever the representation or medium used. In an ideal world, there would be elaborate trace links between all of these differing data, and these would be established on demand and cost-effectively, as needed, to satisfy end-user needs. An engineer might adjust some default settings, as far as the type of trace link to generate or when each one should be generated, so that traces would be created at the level of granularity appropriate to support the context and intent of a specific traceability-enabled activity or task. Furthermore, these trace links would be maintained in an accurate state by monitoring changes to the software system, at all locations at which it is distributed around the globe. Provisional traceability updates would be generated automatically as and when the system evolves.

In 2035, the traceability is assumed to be:

1 ***Purposed.*** Traceability is fit-for-purpose and supports stakeholder needs (i.e., traceability is requirements-driven).

2 ***Cost-effective.*** The return from using traceability is adequate in relation to the outlay of establishing it.

3 ***Configurable.*** Traceability is established as specified, moment-to-moment, and accommodates changing stakeholder needs.

4 ***Trusted.*** All stakeholders have full confidence in the traceability, as it is created and maintained in the face of inconsistency, omissions and change; all stakeholders can and do depend upon the traceability provided.

5 ***Scalable.*** Varying types of artifact can be traced, at variable levels of granularity and in quantity, as the traceability extends through-life and across organizational and business boundaries.

6 ***Portable.*** Traceability is exchanged, merged and reused across projects, organizations, domains, product lines and supporting tools.

7 ***Valued.*** Traceability is a strategic priority and valued by all; every stakeholder has a role to play and actively discharges his or her responsibilities.

8 ***Ubiquitous.*** Traceability is always there, without ever having to think about getting it there, as it is built into the engineering process; traceability has effectively "disappeared without a trace."

These eight assumptions constitute the eight traceability challenges and are examined in turn in Sections 4 through 11. The framework used to explore and discuss each challenge is described in Section 3.

Note that traceability challenge eight, traceability that is *ubiquitous*, is referred to as the *grand challenge of traceability* because its realization depends upon having made significant progress with each of the seven other challenges. Traceability challenge eight is longer term and all-encompassing.

3 Challenges Framework

The vision for traceability was created as a result of a brainstorming effort among the authors of this technical report, following on from the Kentucky workshop. The concept was to describe what software and systems development would be like in 2035 if the traceability problem were solved. Based upon the Utopian scenario, the assumptions that would need to hold true of the traceability to realize the vision were then determined, also in an iterative manner. This led to agreement upon eight crosscutting concerns that now form the eight traceability challenges. In the subsequent sections of this technical report, each challenge is elaborated according to the following framework:

- ***Link to Vision.*** The challenge is anchored in the Utopian scenario of the vision via a short description.
- ***Problems Addressed.*** The current problems with traceability that realization of the challenge would help to address are summarized. This provides additional rationale and motivation for prioritizing and addressing the challenge.
- ***Dream Process.*** To explore the traceability process that would be needed to realize each challenge, the authors developed a model of a generic traceability process. This model was developed in an iterative manner and is described fully in Section 3.1. In summary, the key activities are: traceability planning and management (strategy), traceability creation and maintenance, and traceability use. The dream ap-

proach to each traceability process activity is described for the challenge. A pre-requisite for this community activity was the need to reach consensus on the use of traceability terminology. This led to the development of the glossary found at the end of this technical report.

- ***Goals.*** The high-level goals that would need to be satisfied to achieve the challenge are listed. These goals are given a unique identifier in order to track progress toward their satisfaction, using the following format: <challenge name> <goal number> (e.g., Purposed G 1, Purposed G 2, Valued G 1, etc.).
- ***Requirements.*** The goals suggest and decompose into a number of requirements. The requirements relevant to each of the traceability process activities (i.e., strategy, creation and maintenance, and use) are examined in turn for each challenge. For each requirement that is defined, a brief review of the current status of the research and practice is provided, and areas of promise are highlighted[15]. These requirements are given a unique identifier in order to track progress toward their satisfaction, using the following format: <challenge name> <requirement number> (e.g., Purposed Req 1, Purposed Req 2, Valued Req 1, etc.). Each requirement is cross-referenced to the goals it supports.
- ***Recommended Research.*** Based upon the prior analysis, a major research theme was identified for each challenge. This was then decomposed into a number of supporting research topics. These research topics are given a unique identifier in order to track future progress with the research, using the following format: <challenge name> <research topic number> (e.g., Purposed RT 1, Purposed RT 2, Valued RT 1, etc.). Each research topic is cross-referenced to the requirements it addresses.
- ***Positive Adoption Practices for Industry.*** The framework ends with a list of practices that, if implemented in industrial settings, would facilitate and / or begin to show satisfaction of the requirements, and so progress toward the realization of the challenge. These industry practices are also given a unique identifier in order to subsequently track the progress, using the following format: <challenge name> <industry practice number> (e.g., Purposed IP 1, Purposed IP 2, Valued IP 1, etc.).

The intention of creating a framework for exploring the challenges was to provide a systematic structure for directing and tracking future traceability research and practice. The details provided in the following sections serve to highlight salient points to assist with this objective, arising from working discussions among the authors of this technical report; they do not provide an exhaustive review of the entire traceability field. The reader is referred to a number of existing surveys for such traceability review material (von Knethen and Paech 2002; Dahlstedt and Persson 2005; Spanoudakis and Zisman 2005; Winkler and von Pilgrim 2010).

3.1 A Generic Traceability Process Model[16]

Figure 1 depicts a generic *traceability process model*. It shows the essential activities that are required to bring *traces* into existence and to take them through to eventual retirement. Traces are *created*, *maintained* and *used*, all within the context of a broader *traceability strategy*. This *strategy* provides the detail of stakeholders' needs, decisions regarding mechanism and automation, and also chains *atomic traces* in some agreed way to enable required activities and tasks. Continuous feedback is a critical aspect of the entire process to enable the *traceability strategy* to evolve over time. The four key activities of this *generic traceability process model* are described in the following sub-sections.

---

[15] Please note that this technical report is a result of community workshops and discussions; the objective is to highlight general points about the state of the art and the practice in traceability, not to provide an exhaustive set of references to projects and publications. The reader is referred to the website of the Center of Excellence for Software Traceability (CoEST) for such materials: *http://www.coest.org*.

[16] All the italicized terms in this section are defined in the glossary at the end of this technical report.

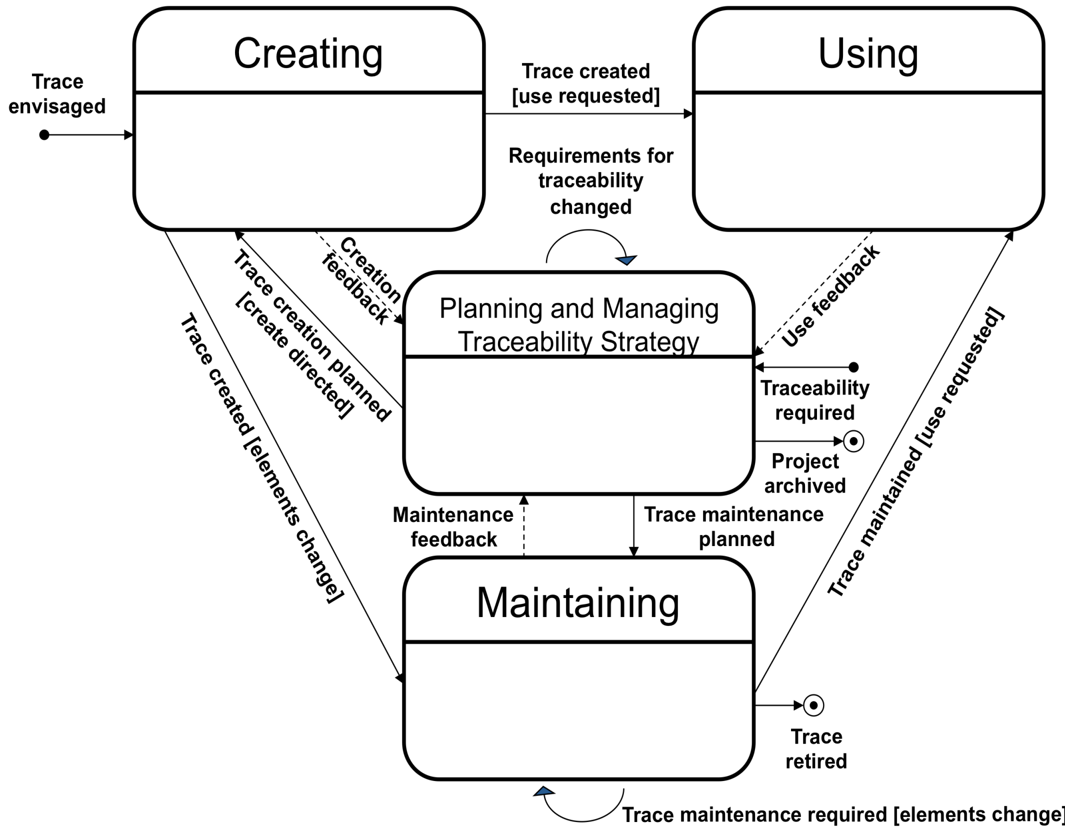

Fig. 1 A generic traceability process model

3.1.1 Traceability Strategy

Effective *traceability* rarely happens by chance or through ad hoc efforts. Minimally, it requires having retained the *artifacts* to be *traced*, having the capacity to establish meaningful *links* between these *artifacts* and having procedures to interrogate the resulting *traces* in a goal-oriented manner. Such simple requirements conceal complex decisions as to the *granularity*, categorization and storage of assorted multi-media *artifacts*. It also conceals choices as to the approach for generating, classifying, representing and then maintaining their inter-artifact and intra-artifact linkages. Additional questions need to be answered, such as: Which of these *tracing* activities should be *manual*? Which should be *automated*? Where should the responsibilities for these activities lie? When should they be undertaken? There are many decisions that need to be made and, therefore, an enabling *traceability strategy* needs to be built into the engineering and management practices from day one on a software and systems engineering project. Figure 2 outlines the typical high-level activities associated with planning and managing a *traceability strategy*.

Traceability is concerned with the provisioning of information to help in answering project-specific questions and in undertaking project-directed activities and tasks; it is thus a supporting system rather than a goal in its own right. This perspective demands understanding those stakeholders who may need the potential for *traceability*, what for and when? Acquiring clear-cut answers to these questions at the start of a project is not straightforward, as both stakeholders and their task needs will change. Even if these could be articulated exhaustively, building a *traceability solution* to service all needs is unlikely to be cost-effective, as resources are generally limited in some finite way. Determining whose needs to satisfy, and so which *traceability-enabled activities and tasks* to facilitate, is a value decision that lies at the heart of a *traceability strategy*; determining needs and resourcing constraints is a precursor to any discussion about *trace artifacts*, *trace links* and mechanism.

Ensuring that the *traceability* is then *established* as planned, and yet can adapt to remain effective as needs evolve and as a project's *artifacts* change, is also the province of *traceability strategy*. Determining how the *traceability* will be provisioned such that the requisite quality can be continuously assured further demands analysis, assessment and potential modification of the current *traceability solution*. Assessing the quality and the execution of the *traceability solution*, and implementing a feedback loop to improve it, is a critical part of the *traceability strategy* for a project; it needs to develop and leverage historical *traceability information*.

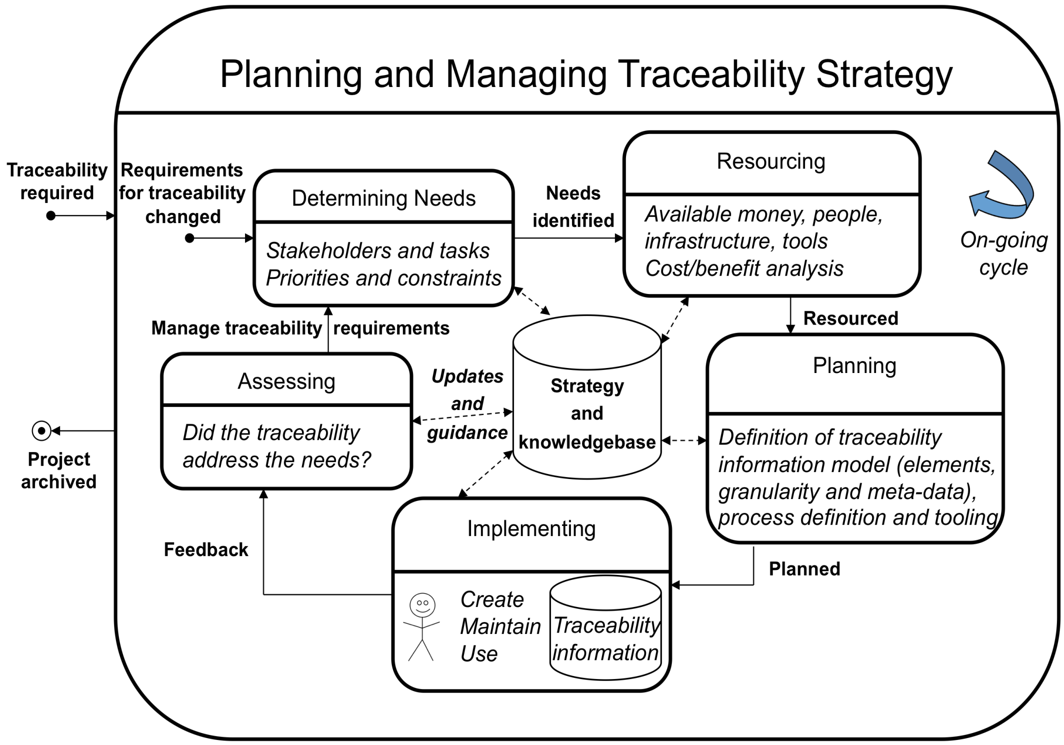

Fig. 2 Planning and managing a traceability strategy

Within the context of a broader *traceability strategy*, the *creation*, *maintenance* and *use* of individual *traces* and their constituent *elements* all need to be defined and managed. Given that *atomic traces* comprise *source*, *target* and relational *elements*, these data requirements need to be identified. This includes decisions as to meta-data to associate, dependent upon what kinds of *traceability-enabled activities and tasks* the *trace* is anticipated to participate in and support. Resourcing, planning and implementation decisions may hence vary on a trace-by-trace basis; for instance, it is quite possible that a particular *trace* is not *created* or *maintained* until its use is actually required. *Traces* thereby inhabit independent lifecycles, the constituent activities of which are examined in the following sub-sections.

3.1.2 Traceability Creation

When creating a *trace*, the *elements* of the *trace* have to be acquired, represented and then stored in some way, as illustrated in Figure 3. *Reference models* and classification schemes characterizing different types of *trace link* and *trace artifacts* drive the *traceability creation* process, as usually defined within the *traceability information model* of the overarching *traceability strategy*.

While project *artifacts* are generally pre-existing on a project, the *links* between them may not yet be defined. Techniques to support the *creation* of *trace links* can range from *manual* to *automated* approaches, each with differing degrees of efficiency and effectiveness. The differentiating factor is often whether the *trace links* are created concurrently with the forward engineering process (i.e., *trace capture*) or at some point later (i.e., *trace recovery*). Validation is therefore critical to the viability of the *traceability creation* process, regardless of how *trace links* are initially created, as it is concerned with determining and assuring the credibility of the *trace* as a whole.

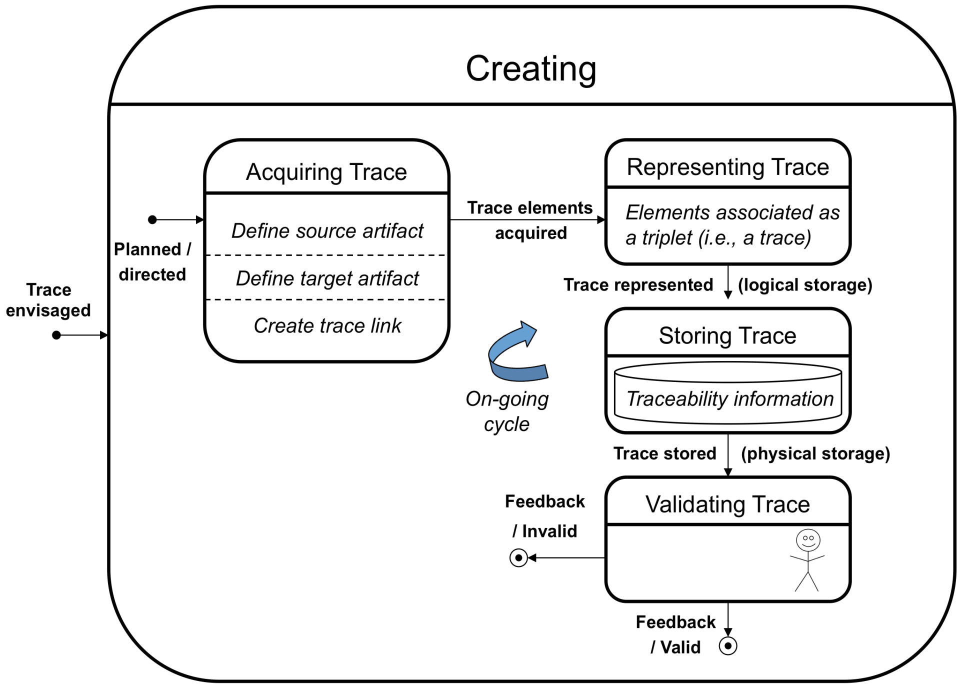

Fig. 3 Traceability creation

3.1.3 Traceability Maintenance

An *association* made between two *artifacts* at a moment in time to serve a particular purpose does not automatically mean that the resulting *trace* will have a persistent, useful life. The need for *maintenance* on a *trace* can be triggered by changes to any of the *trace*'s *elements* that, in turn, can be triggered by changes to *elements* within a *chain*. *Traceability maintenance* can also be required following changes to the requirements and constraints that drive the overarching *traceability strategy*.

To maintain a *trace*, it needs to be retrieved and the nature of the change analyzed to determine what update is necessary, as illustrated in Figure 4. This may necessitate the propagation of changes and / or the *creation* of entirely new *traces*. Updates need to be performed, where applicable, recorded and verified. Feedback on the *maintenance* process is also essential for evolving the overarching *traceability strategy*. As per *traceability creation*, *traces* can be *maintained continuously* or *on-demand*.

3.1.4 Traceability Use

The availability and usefulness of *traces* has to be ensured to allow for their ongoing use throughout the software and systems development lifecycle, potentially in a myriad of configurable ways. Here, it is helpful to distinguish between short-term *traceability use* during initial product development and long-term *traceability use* during subsequent product maintenance. Typical short-term uses for *traceability* include requirements completeness analysis, requirements trade-off analysis or requirements-to-acceptance-test mapping for final acceptance testing. Typical examples of long-term uses for *traceability* include the determination of effects of changes to a software system or the propagation of changes during its evolution.

Any *atomic trace* is likely to play a role in the context of many use contexts. To use a *trace* in isolation, or as a part of a *chain*, it needs to be retrieved and rendered visible in some task-specific way, as suggested in Figure 5. An important component of the *use* process is assessing the *quality of the traceability* that is provided in terms of the fitness for purpose with respect to the task or activity for which the *traceability* is required. Such information provides a feedback loop to improve the overall *traceability strategy*.

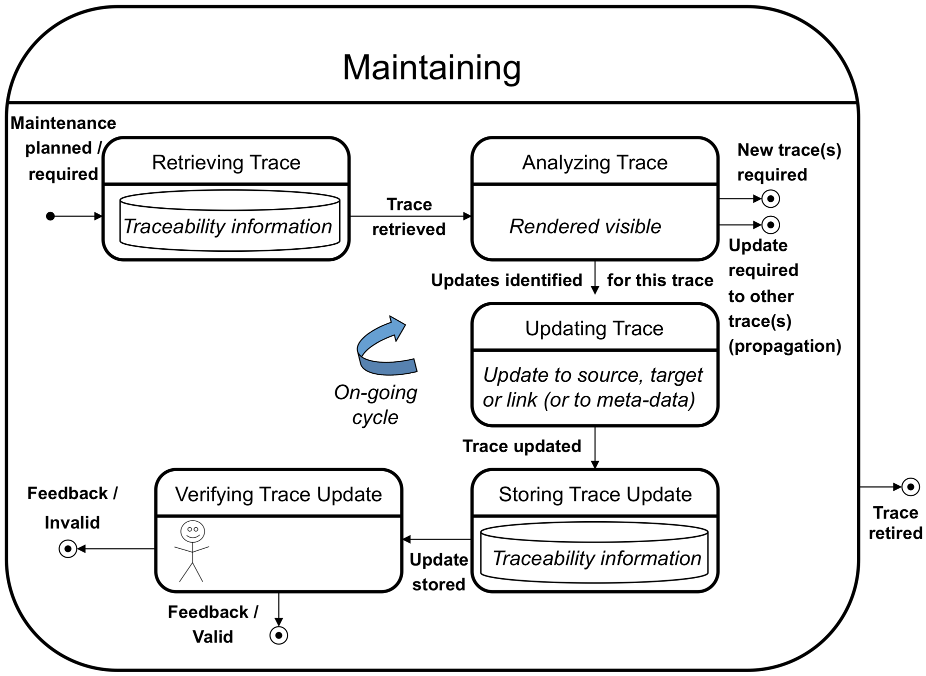

Fig. 4 Traceability maintenance

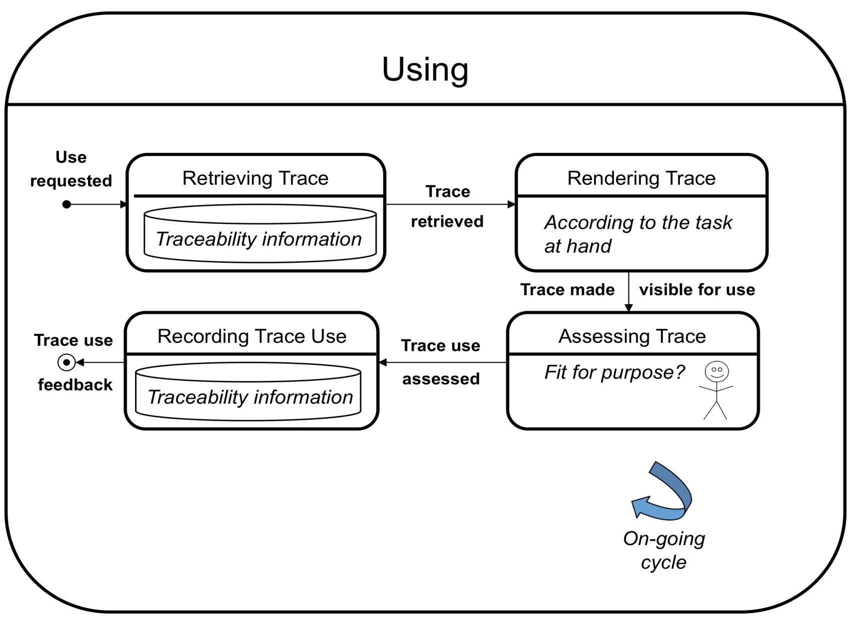

Fig. 5 Traceability use

4 Traceability Challenge 1: Traceability that is Purposed

Traceability is fit-for-purpose and supports stakeholder needs (i.e., traceability is requirements-driven).

4.1 Link to Vision (Purposed)

In the vision scenario, traceability helps the engineer to detect those stakeholders to involve during requirements elicitation, to identify missing and conflicting requirements, and to demonstrate compliance to regulatory codes. Traceability also helps the engineer to see the impact of new and modified requirements, and facilitates the requirements negotiation and validation process with appropriate stakeholders. Traces are used to retrieve the context and rationale for decisions, to examine costs and to verify compliance to product requirements. Traceability supports the engineer explicitly in all aspects of her daily work over the course of the project. The traceability is fit-for-purpose.

4.2 Problems Addressed (Purposed)

Traceability will not be implemented and used in practice unless it is perceived as useful or is mandated. Currently, there is poor understanding of what people need traceability for and how people actually use traceability over time. Further, traceability will not be created or maintained effectively if the required tasks to do so are themselves not understood and supported. Currently, there is poor understanding of what individuals and teams need to do to create and maintain traces. This distinction between satisfying the requirements of those stakeholders who establish traceability and those stakeholders who use traceability lies at the heart of many traceability problems, for these roles are not necessarily overlapping. The stakeholder community for establishing and using traceability is potentially vast and dynamic, and the skills and incentives of these stakeholders vary widely. Tools are frequently purchased to enable traceability but, because they are often insufficiently configured to support these specific stakeholder requirements for traceability, they do not support their processes nor adapt to their changing needs; therefore, the tools rarely realize their potential.

4.3 Dream Process (Purposed)

- ***Traceability Strategy.*** The initial stakeholder requirements for traceability on a project will be selected from profiles and templates, and the integrated development environment used on the project will handle all the details necessary to design and implement a traceability solution to satisfy them. The effectiveness of this solution will be measured over time as the requirements evolve and are accommodated.
- ***Traceability Creation and Maintenance.*** All traces on a project will be demonstrably created based upon specified stakeholder requirements for establishing traceability, accounting for the nature of the artifacts to be traced in different environments. Once created on a project, the traces will be maintained such that changing stakeholder requirements for establishing traceability are continuously and demonstrably satisfied.
- ***Traceability Use.*** The traceability provided will fit the end-users' contexts and needs. A feedback-driven learning system will adapt the traceability that is established to fully address its end-users' evolving task contexts and needs.

4.4 Goals (Purposed)

Purposed G 1 — Prototypical stakeholder requirements for traceability use are understood, defined and shared by the software and systems research and development communities.

Purposed G 2 — Prototypical stakeholder requirements for creating and maintaining traceability are understood, defined and shared by the software and systems research and development communities.

Purposed G 3 — Stakeholder requirements for traceability drive, and are demonstrably satisfied in, traceability solutions.

Purposed G 4 — The effectiveness of the traceability in end-use is measured and drives traceability process improvement.

Purposed G 5 — The effectiveness of the traceability creation and maintenance process is measured and drives traceability process improvement.

Purposed G 6 — Executed trace queries provide value beyond simply retrieving a set of artifacts; they actively support specific software and systems engineering tasks.

4.5 Requirements (Purposed)

4.5.1 Traceability Strategy (Purposed)

<u>*Purposed Req 1*</u> — To understand and define the full range of stakeholders to be supported in and by a comprehensive traceability solution. [Purposed G 1, G 2]

- *Status:* Little attention has been paid to the full set of stakeholders for traceability. The focus of both research and practice has been on partial views of restricted constituents and their tasks, and this knowledge has not been consolidated in one place for the traceability community. Many traceability stakeholders are typically forgotten about during strategy formulation, such as the downstream consumers of traceability (e.g., subcontractors). Stakeholders also have both near and long-term needs for traceability, and this is rarely distinguished in the strategy.
- *Promise:* Traceability personas are being developed by the Tracy project to explore stakeholder requirements for traceability tools (Cleland-Huang et al. 2011). Characterizing personas and their requirements for establishing and using traceability (i.e., standard role models) would be a natural and valuable extension of this work, and would begin to address the current lack of requirements focus in and by the traceability community.

<u>*Purposed Req 2*</u> — To understand and characterize the contextual factors that constrain and shape options for traceability solutions, such as the project type, organizational type, regulatory demands, domain, etc. [Purposed G 1, G 2]

- *Status:* The contextual factors shaping the traceability provided in various domains are generally explained in any case study reporting. Various classification schemes also exist to characterize the nature of projects, organizations, etc. However, there is as yet no agreed upon classification scheme and the demands that such factors place on traceability solutions have not yet been examined systematically.
- *Promise:* Such classification material could be consolidated so as to begin to be more methodical about defining the 'context' for traceability solutions. Empirical studies could then be framed by comparable expressions of their contexts.

<u>*Purposed Req 3*</u> To understand and define the numerous properties required of traceability for it to be considered effective for supporting the various stakeholder tasks and contexts, such as different demands on trace quality, completeness and granularity. [Purposed G 1, G 2, G 4, G 5, G 6]

- *Status:* Given the lack of systematic attention to stakeholder identification and requirements determination for traceability, there has been a corresponding lack of attention to what would be required of a traceability solution to satisfy them; these criteria for assessing the effectiveness of any supporting traceability solutions are rarely articulated.
- *Promise:* Better definition of stakeholder requirements for traceability, along with their contexts and acceptance criteria, will help in designing and assessing potential solution options for traceability more comprehensively.

<u>*Purposed Req 4*</u> To design traceability solutions that are driven by, and traceable to, stakeholder requirements and contexts for traceability, providing access to the rationale for strategic decisions. [Purposed G 3]

- *Status:* There are few practical guidelines for practitioners as to good practices for designing and implementing an effective and traceable traceability solution for their project, one that is driven by the stakeholder requirements for traceability and project context. There is much reliance on past experience and informal knowledge sharing at present.
- *Promise:* The creation of a Traceability Body of Knowledge (TBOK), a resource proposed by the traceability community for the community, is essential to disseminate good traceability practices and to advance them. The development of such a resource is part of the impetus behind the formation of the Center of Excellence for Software Traceability (Hayes et al. 2007).

<u>*Purposed Req 5*</u> To tailor traceability solutions to accommodate key and potentially changing stakeholder requirements and contexts for traceability, and to evolve the overarching traceability strategy as needed. [Purposed G 3]

- *Status:* Once the traceability solution has been designed for a project context, the strategy is generally to fix this solution for the duration of the project. It can be a non-trivial exercise to reconfigure the entire approach mid-project.
- *Promise:* The growth in the use of agile approaches to software and systems development, coupled with more focus on the use of services to satisfy requirements, is necessitating the development of lightweight, lean and dynamic traceability solutions. Such solutions are emerging.

<u>*Purposed Req 6*</u> To agree upon measures of effectiveness with respect to organizational and business needs for a traceability strategy and its component aspects. [Purposed G 4, G 5]

- *Status:* There are no agreed upon measures for assessing the effectiveness of competing traceability strategies in different organizational and business contexts. Reporting on the effectiveness of an overarching traceability strategy, and its underlying models, processes and tools is largely the remit of qualitative industrial case studies at present.
- *Promise:* The number of industrial case studies and traceability experience reports has been growing in recent years and there would be value in more systematic cross-comparison of this work.

4.5.2 Traceability Creation and Maintenance (Purposed)

<u>Purposed Req 7</u> To understand and define the requirements and constraints of those stakeholders who create and / or maintain traces (i.e., the creators' and maintainers' requirements for traceability). [Purposed G 2]

- *Status:* The goal of research is to simplify the task of traceability creation and maintenance by reducing the human effort required. However, this research has focused more on the study of techniques, methods and tools than on the people creating and maintaining the traceability and their needs. As a consequence, there is little real appreciation as to what may be gained or what may be lost by the move to increasing automation in these processes, such as the tacit role that a manual creator or maintainer plays, and the implicit development and maintenance knowledge gained by humans from doing the work.
- *Promise:* Empirical studies of humans undertaking various traceability tasks are beginning to emerge from research on trace automation, and this will lead to more understanding of the underlying activities and provide baselines for performance comparison.

<u>Purposed Req 8</u> To develop a model of the general process of traceability creation and maintenance that depicts the generic workflow and component activities of the process and articulates the lifecycle of a single trace within this process. [Purposed G 2]

- *Status:* There is no fine-grained description of traceability creation and maintenance processes, along with how these fit into a wider traceability process. The various steps and activities involved in creation and maintenance of a single trace are neither articulated nor agreed.
- *Promise:* Initial work on a generic traceability process model by the traceability community deconstructs the traceability creation and maintenance processes into their fundamental activities, and examines the workflow needed to create and maintain a single trace. Understanding and agreeing upon the underlying specifics of these processes will potentially help to identify process bottlenecks, and then guide and improve the support in these areas.

<u>Purposed Req 9</u> To use the creators' and maintainers' requirements for traceability, in conjunction with a generic traceability process model, to guide and support the traceability creation and maintenance process. [Purposed G 3]

- *Status:* While there has been focus on the need to define the traceability process to be enabled by traceability techniques, methods and tools, the support to actually define this process on a project and then to implement this process in a team setting is not always readily available to practitioners.
- *Promise:* Guidance for traceability process definition is provided in some leading commercial tools or supported via consulting arrangements. Ideally, creators and maintainers would be provided with the means to define and configure their own working processes.

<u>Purposed Req 10</u> To agree upon measures of effectiveness with respect to traceability creation and maintenance. [Purposed G 3]

- *Status:* Researchers have conducted some initial studies to compare the effectiveness of fully automated, semi-automated and manual approaches to traceability creation and maintenance, resulting in well-accepted measures of trace recall and trace precision. While these measures focus on the quality of the trace links, they do not account for the quality of their end-use by stakeholders. They also do not account for the impact of using various traceability creation and maintenance techniques, methods and tools on the wider development tasks.
- *Promise:* Any discussion on the effectiveness of the traceability creation and maintenance process needs to be tied to the effectiveness of the traces in end-use. Even where trace links are well crafted, this does

not imply that the creation process was effective. Promise lies in a more sophisticated understanding and analysis of ‘effectiveness’ and its associated measures.

<u>*Purposed Req 11*</u> To gather data on and monitor the process of traceability creation and maintenance, using agreed measures of effectiveness, so as to continuously improve the process. [Purposed G 3, G 5]

- *Status:* Researchers have conducted some initial studies to compare the effectiveness of fully automated, semi-automated and manual approaches to traceability creation and maintenance, though this has not yet matured to using these data to then evolve the process of creation and maintenance.
- *Promise:* More comparative studies of manual processes for traceability creation and maintenance with semi and fully automated settings are emerging, along with baselines for comparisons upon which to improve. Benchmark experiments and data sets will shape the future research direction and practical uptake strategies in these areas.

<u>*Purposed Req 12*</u> To understand the paradigm used to develop the software or system (e.g., object-oriented, agent-oriented, service-oriented, product line, etc.), the nature of the artifacts involved and the domain specifics, so as to contextualize support for the traceability creation and maintenance processes within the wider software and systems development workflow. [Purposed G 3]

- *Status:* Approaches to traceability vary across development type and domain. However, there has been little systematic effort to articulate those project characteristics that impact the choices made for the approach to traceability creation and maintenance.
- *Promise:* Understanding what approaches to traceability creation and maintenance work best in different situations, and blending approaches as needed.

<u>*Purposed Req 13*</u> To collect and use data about traceability evolution, such as intermediate versions of traces, to improve the initial traceability creation process and subsequent maintenance tasks. [Purposed G 5]

- *Status:* Trace maintenance data has been under-utilized to date.
- *Promise:* Historical traceability data may reveal useful insight into both traceability creation and maintenance process improvement areas.

4.5.3 Traceability Use (Purposed)

<u>*Purposed Req 14*</u> To understand and define the full range of stakeholders who use the end products of traceability (i.e., its end-users), their task needs, their constraints and their contexts of use. [Purposed G 1, G 6]

- *Status:* To date, the focus of the traceability community has been more on the processes and software needed to support the mechanics of traceability than on the needs of the consumers of the traces. Where the needs of the end-users is a concern, research has focused mostly on using traceability to support the tasks of a specific subset of stakeholders, such as independent validation and verification analysts, and representatives from regulatory bodies, rather than on the full range of end-users.
- *Promise:* End-user stakeholder requirements for traceability are discussed in a fragmented way across various publications, often in terms of high-end users and low-end users of traceability. Typical end-user requirements for traceability in different projects, organizations and domains could be consolidated and classified as a definitive resource for the traceability community to draw upon.

Purposed Req 15 — To provide guidelines to determine and prioritize which traces are needed on a project, by whom, for what purposes, when, how, at what level of granularity, under what constraints, etc. [Purposed G 1]

- *Status:* Actual traceability use in various domains is patchy, as engineering professionals do not always recognize that traceability is needed or could save money or lives. Traceability need assessment is quite coarse and little active support is provided to do this.
- *Promise:* Practitioners are beginning to publish more experience reports of traceability in use to the wider traceability community. However, there is the issue of confidentiality that restricts progress. When organizations implement traceability techniques and methods that do not work as intended, they do not always publish the results. This makes it very hard for the traceability community to find out what does and does not work over time. Better ways to anonymize, sanitize and incentivize such reporting are sorely needed.

Purposed Req 16 — To agree upon measures of effectiveness with respect to traceability in end-use. [Purposed G 4]

- *Status:* There are no proposed or routinely used measures to assess traceability effectiveness in end-use in different organizational and business contexts. Traceability metrics tend to focus on the effectiveness of the actual trace links (i.e., is it a real one?) and so support assessment for traceability creation and maintenance purposes only. Researchers have no hard statistics to confirm whether traceability actually enables what it sets out and purports to do.
- *Promise:* Researchers advocate the use of traceability information models that capture decisions about the anticipated traceability-related queries that the traceability solution should support, and describe the trace artifacts and the trace links needed to support those queries. This requirements and task-directed approach is promoted in researcher-led training sessions and there has been some initial uptake in practice. The obvious next step is to track the effectiveness of the solution in satisfying these queries from an end-user perspective. The metrics component now needs more consideration.

Purposed Req 17 — To gather data on and monitor traceability end-use against stakeholder requirements for traceability, using agreed measures of effectiveness, to evolve the end-user requirements and the capacity for their satisfaction in traceability solutions. [Purposed G 3, G 4]

- *Status:* If practitioners have end-use effectiveness data, it is rarely shared within the traceability community, for the reasons described above. Practitioners primarily rely upon word of mouth (externally) or tool-generated traceability-related reports (internally) to get feedback about traceability end-use for process improvement purposes.
- *Promise:* Anonymous feedback, ranking and rating systems are now common when distributing information on websites. There could be potential in examining similar strategies for evaluating traceability end-use, focusing on measures less reliant on the concept of ‘traffic’ or ‘throughput’, to assess whether the results of traceability are used as intended and are actually useful in practice.

4.6 Recommended Research (Purposed)

The major research theme to achieve purposed traceability is ***to define and instrument prototypical traceability profiles and patterns***. These would comprise typical stakeholder requirements for traceability, a way to characterize the wider project context, and recognized approaches for their accommodation and satisfaction in traceability solutions. Supporting research topics are listed below.

Research ID	Description	Req ID
Purposed RT 1	Develop a profile of prototypical role, task and context-based stakeholder requirements for traceability, including scenarios of end-use for traceability.	Purposed Req 1, 7, 8, 14, 15
Purposed RT 2	Develop a classification scheme to define the context of a traceability need, such as salient properties of projects, organizations and domains.	Purposed Req 2, 12, 15
Purposed RT 3	Develop patterns for traceability implementations associated with traceability profiles and contexts.	Purposed Req 3, 4, 9, 15
Purposed RT 4	Instrument a mechanism to both use and evolve this resource of profiles, contexts and patterns, integrating feedback from practice and experience.	Purposed Req 5, 9, 11, 13, 17
Purposed RT 5	Propose and agree upon metrics to measure effectiveness in all areas of the traceability process.	Purposed Req 3, 6, 10, 16
Purposed RT 6	Perform empirical studies to determine whether the various stakeholder types find traceability techniques, methods and tools fit-for-purpose.	Purposed Req 3, 6, 10, 11, 16, 17
Purposed RT 7	Develop a Traceability Body of Knowledge (TBOK) to define the traceability terminology, profiles, contexts, patterns, practices, techniques, methods and tools, and to include resources on metrics, case studies, lessons, experts, benchmarking, baselines, etc. Careful attention will need to be paid to the contribution process for the credibility and sustainability of such a resource.	Purposed Req 1-17

4.7 Positive Adoption Practices for Industry (Purposed)

Purposed IP 1 — Practitioners consult, use and contribute to an evolving Traceability Body of Knowledge (TBOK).

Purposed IP 2 — Practitioners draw upon prototypical traceability profiles, contexts and patterns when designing and implementing a traceability solution for their project, organization and domain.

Purposed IP 3 — Practitioners routinely measure the effectiveness of all aspects of their traceability process, evolve their solution accordingly and contribute these data to the Traceability Body of Knowledge (TBOK).

5 Traceability Challenge 2: Traceability that is Cost-effective

The return from using traceability is adequate in relation to the outlay of establishing it.

5.1 Link to Vision (Cost-effective)

By establishing traceability automatically and early in the vision scenario, the engineer is alerted to product requirements that she overlooked in the initial stages of the engineering process, avoiding the need for costly rework later. Such knowledge has been accrued over a myriad of projects thanks to traceability analyses. The engineer is able to focus on her job, and on those analyses that demand her expertise and decision-making skills, and is not distracted by building in traceability support continuously as she works. Moreover, by having the opportunity of creating or maintaining the traceability on-demand later, the engineer does not have to worry now about having a traceability problem in the future; she knows that any missing traceability can always be established cost-effectively if and when needed, based upon tried and tested best-of-breed techniques, methods and tools.

5.2 Problems Addressed (Cost-effective)

Complete traceability is often impractical, expensive to establish and not always necessary. Too much time can be invested in establishing traceability that may never be used or useful on a project, such as the provision of rich link semantics that are not actually exploited in traceability-related queries or analyses. It is difficult to know what is 'just enough' traceability for each project situation because these situations themselves are often poorly expressed. The costs incurred in establishing traceability are also perceived to come too early on in a project, which leads to delays in implementing traceability, or in implementing it only under crisis mode; but traceability is not something that can be retrofitted with ease later. Because there is little sharing of good practices and heuristics for traceability, costs can further escalate as well known mistakes are made. Furthermore, there is inadequate understanding of the costs incurred during the entire traceability lifecycle, so the approximate return on investment from traceability is not readily known or knowable at present. Together, these issues give traceability a bad reputation financially and present a real dilemma, as industry is reluctant to take on new approaches emerging from research without more data on the full costs and anticipated returns.

5.3 Dream Process (Cost-effective)

- ***Traceability Strategy.*** Interactive and intelligent planning models, decision support tools and return on investment simulators will illustrate the business impact of spend decisions on traceability solution options.
- ***Traceability Creation and Maintenance.*** Traceability will only be created when it is needed, at exactly the quality needed – no more, no less – and each trace will be created in the most economical way possible to serve its intended purpose. Just enough traceability will always be maintained, and each trace will be maintained in the most economical way possible to continue to serve its intended purpose. Traces will be archived and discarded once they are no longer needed to avoid unnecessary maintenance costs.

- ***Traceability Use.*** The end-user will always be effectively supported in his or her task. The costs for establishing this traceability will only be incurred at the point of end-use, which will be proportional to the benefits obtained, and these data will be known ahead of time for planning purposes.

5.4 Goals (Cost-effective)

Cost-effective G 1	The total cost of traceability throughout a project's life is computed, along with the projected return on investment, and it is available to assess the potential effectiveness of competing traceability solutions.
Cost-effective G 2	Just enough traceability is provided, balancing the stakeholder requirements for traceability with the resource constraints.
Cost-effective G 3	The perfect middle ground between creating and maintaining traceability early and creating and maintaining traceability on demand is attained, so that the time, effort and money that are expended in establishing traceability are in balance with the resourcing profile of the project and the required quality in end-use.
Cost-effective G 4	Lessons learned are captured, shared and capitalized upon, so that the cost and effectiveness of various traceability techniques, methods and tools are known and improved upon.
Cost-effective G 5	Intuitive user interfaces and interaction mechanisms enable process-related cost decisions to be explored and altered at all stages of the traceability process. The factors that influence traceability cost-effectiveness at different stages of the project lifecycle are hence monitored and the traceability process can be adapted as needed.

5.5 Requirements (Cost-effective)

5.5.1 Traceability Strategy (Cost-effective)

<u>*Cost-effective Req 1*</u> To provide support to get the right traceability (how good) at acceptable cost (how much) at the appropriate time (when) during traceability planning. [Cost-effective G 2, G 3, G 5]

- *Status:* There are no traceability-specific planning techniques and tools that help the practitioner to balance stakeholder requirements for traceability against its implementation costs. Practitioners tend to rely upon more traditional project management techniques and tools to assist their traceability planning at present. Furthermore, the resulting strategies are unlikely to vary over time.
- *Promise:* A better understanding and definition of what traces are needed, when and where, at what levels of quality, and for what duration on a project (i.e., progress with traceability challenge one) will assist with progress on this challenge. Research on value-based traceability is already underway and is needed for making strategy decisions on the traceability that is needed, leading to viable and mixed approaches in the future, and to more sophisticated visual planning aids.

<u>*Cost-effective Req 2*</u> To agree upon metrics for measuring the traceability return on investment on a project, informing those data to collect, and those mechanisms to put in place to obtain these data and measures. [Cost-effective G 1]

- *Status:* Few agreed upon return on investment metrics are available for traceability, let alone used routinely, when planning and making strategic traceability decisions.
- *Promise:* Value-based approaches could lead to the situation where every trace that is created and maintained, manually, semi-automatically or fully automatically, is routinely tagged with data on both the price to create and maintain the trace, and the expected return in terms of the anticipated need it will satisfy. The cost to achieve this crude metric would itself need to be balanced against the benefits of so doing.

Cost-effective Req 3 To understand the fixed and variable costs for a lifecycle-wide traceability solution. [Cost-effective G 1]

- *Status:* Currently, there is little examination as to where the various traceability costs actually lie across the entire software and systems development lifecycle. Furthermore, there is little understanding as to the essential costs and the optional costs, such as those specific to particular project characteristics.
- *Promise:* Models of the traceability process are beginning to decompose the underlying activities of traceability, thus providing a structure to investigate and delineate the cost profile. This needs to be superimposed on to development lifecycle models and the wider cost profile.

Cost-effective Req 4 To understand the costs and benefits of establishing traceability at different times on a project, and at varying levels of granularity. [Cost-effective G 3]

- *Status:* There are two extreme strategies for establishing traceability: (1) Early, by people who are familiar with the software or system. While this may produce quality traces at little cost per trace, the traces may never be used or useful; (2) On-demand, by people who may lack intricate knowledge of the software or system. While the speed and quality of the traces may be lacking in this approach, the traces that are produced are actually needed and used. No single strategy is perfect and a balance is now being sought.
- *Promise:* Proposals to distribute the cost of traceability across the whole project lifecycle, and to mix strategies such as (1) and (2) over this lifecycle, have been made and now need to be developed further.

Cost-effective Req 5 To capitalize upon historical return on investment measures and cost-benefit analyses when setting up a traceability strategy. [Cost-effective G 1, G 2, G 4]

- *Status:* Many of the benefits resulting from traceability may be realized only after the delivery of a product. This is hard to factor into fixed budgeting strategies without historical evidence of such. Practitioners share data on traceability practices, and rely upon past experiences when formulating traceability strategies, but this knowledge sharing may be restricted to personal networks or internal to organizations at present.
- *Promise:* There is a growing body of practitioner experience reports that are beginning to disseminate knowledge on traceability results and successful practices among the traceability community. More quantitative data on the costs and benefits now need to be gathered.

5.5.2 Traceability Creation and Maintenance (Cost-effective)

Cost-effective Req 6 To establish benchmarks to compare and contrast the cost-effectiveness of the various traceability creation and maintenance techniques, methods and tools. [Cost-effective G 4]

- *Status:* There is little comparative data available on the cost-effectiveness of various traceability techniques, methods and tools. Researchers have focused on measuring disparate aspects of individual approaches. There is no simple mechanism for the practitioner to measure the cost-effectiveness of the total trace creation and maintenance effort on a project because the cost-effectiveness of creating and maintaining even a single trace link is not measured at present.

- *Promise:* Benchmarking has become a priority topic within the traceability research community. The Tracy project is developing TraceLab as an environment within which to facilitate the development and use of such benchmarks for experimental studies. This should lead to the availability of more comparative data in the near future.

<u>*Cost-effective Req 7*</u> To provide a mix of continuous and on-demand approaches to traceability creation and maintenance to balance the costs throughout a project's life. This may include traces that are discovered, created and maintained only when needed. [Cost-effective G 3]

- *Status:* In practice, traces are often created that are never used, mostly manually, which is costly in terms of the time, effort and money expended. The true costs of this expenditure are rarely measured and known. The research focus on automated traceability creation and maintenance seeks to reduce the costs and errors that occur when this process is performed manually. The emphasis to date has been on exploring continuous versus on-demand approaches to traceability creation and maintenance, and the effectiveness of these techniques and methods, not on their respective costs.
- *Promise:* The promise lies in the potential to mix and match from a portfolio of complementary traceability creation and maintenance approaches, so as to balance needs with the available resources. To do this effectively, the various options will need to have cost profiles.

<u>*Cost-effective Req 8*</u> To develop more cost-effective techniques, methods and tools for traceability creation and maintenance. [Cost-effective G 4]

- *Status:* With a research focus on the automated creation of trace links, to save on the costs of initial traceability creation and the costs of ongoing maintenance, the emphasis has been on the effectiveness of these techniques, methods and tools in creating actual trace links. The costs incurred and the savings made in using these, in relation to manual processes, are still under investigation. Moreover, it is the traces that are used in practice that are more likely to be maintained, whereas those that are not used are left to decay. There has been no research to date on whether this is an effective strategy.
- *Promise:* The focus on benchmarks for traceability, establishing frameworks for experimentation and baselines to improve upon, will provide the needed comparative data to assess and improve upon individual techniques, methods and tools.

5.5.3 Traceability Use (Cost-effective)

<u>*Cost-effective Req 9*</u> To reduce the cost and increase the performance of retrieving and displaying traces for end-use. [Cost-effective G 1, G 4, G 5]

- *Status:* The costs for undertaking each activity in the traceability process are rarely quantified at present. The assumption is that retrieving and displaying traces can be a performance bottleneck and a deterrent to end-use where it distracts the end-user from their primary task.
- *Promise:* Ongoing improvements in performance with regard to information retrieval and data visualization will negate this issue over time, leading to the potential for real-time immersive trace data to facilitate end-user tasks more seamlessly.

<u>*Cost-effective Req 10*</u> To configure and adapt traces to support end-user tasks dynamically, creating new traces on-demand as needed, rather than hardwiring them in upfront just in case they are needed. [Cost-effective G 3]

- *Status:* In practice, many trace links are created and maintained that are either never used or never used effectively, partly because they are not needed and partly because the associated traces required to support a complete end-user task are missing. Research has not identified the optimal set of traces, partly because it does not have a thorough understanding of stakeholder needs (traceability challenge one).
- *Promise:* Progress in automated traceability creation could lead to dynamically generating traceability to support end-user tasks, if low cost. This would need to be coupled with a greater understanding of task-specific needs, and a way for end-users to articulate these needs both dynamically and non-intrusively.

<u>*Cost-effective Req 11*</u> To provide visualizations and interaction mechanisms for end-users to navigate and access traces, so as to render traces more effective for task-supported end-use. [Cost-effective G 5]

- *Status:* The artifacts that are related on a project are generally presented to practitioners in ways that do not always support their end-user tasks explicitly, such as via textual lists or traceability matrices. So, while traceability may be present on a project, it is not guaranteed that the practitioner can and will use it. This means that the return from the effort expended may never be realized. Little research attention has been paid to the usability and effectiveness of the results of traceability in end-user tasks or to improvement thereof.
- *Promise:* Researchers are beginning to propose interesting visualizations for traceability, but these tend to depict the trace links so as to support their validation rather than to support end-user tasks. Human-computer-interface researchers and practitioners, interaction designers and visual artists are enhancing many aspects of software and systems development practice. Their contributions are essential to make traceability end-use more intuitive and amenable to task support.

5.6 Recommended Research (Cost-effective)

The major research theme to achieve cost-effective traceability is ***to develop cost-benefit models for analyzing stakeholder requirements for traceability and associated solution options at a fine-grained level of detail.*** Supporting research topics are listed below.

Research ID	**Description**	**Req ID**
Cost-effective RT 1	Agree upon metrics for measuring traceability cost-effectiveness.	Cost-effective Req 2
Cost-effective RT 2	Understand the typical cost profile of traceability outlay on a project.	Cost-effective Req 3, 9
Cost-effective RT 3	Develop the means to associate a cost and a benefit profile with every trace that is brought into existence and maintained.	Cost-effective Req 7, 10
Cost-effective RT 4	Create decision support tools and impact analysis tools for making traceability return on investment decisions, such as a mechanism to globally and locally optimize the traceability solution based upon stakeholder requirements for traceability, the available resources and	Cost-effective Req 1, 4, 5, 7

	the return on investment required.	
Cost-effective RT 5	Gather and disseminate benchmark empirical studies for researchers to demonstrate the cost-effectiveness (or not) of various traceability processes, techniques, methods and tools, as part of the Traceability Body of Knowledge (TBOK).	Cost-effective Req 3, 5, 6
Cost-effective RT 6	Decrease the costs and improve the effectiveness of the techniques, methods and tools supporting all activities of the traceability process.	Cost-effective Req 8, 11

5.7 Positive Adoption Practices for Industry (Cost-effective)

Cost-effective IP 1 — Practitioners consult the Traceability Body of Knowledge (TBOK) to understand the cost-effectiveness of existing and new techniques, methods and tools when making traceability strategy decisions.

Cost-effective IP 2 — Practitioners use decision support tools and impact analysis tools to explore the cost-effectiveness of employing various and mixed traceability strategies on a project, and to help adapt the strategy over time.

Cost-effective IP 3 — Practitioners track the return on investment from traceability on a project and contribute these data routinely to the Traceability Body of Knowledge (TBOK).

[continued…]

6 Traceability Challenge 3: Traceability that is Configurable

Traceability is established as specified, moment-to-moment, and accommodates changing stakeholder needs.

6.1 Link to Vision (Configurable)

Traceability is established and used with consistency across the distributed teams in the vision scenario, to suit the particular needs of the engineer's project, organization and domain. As the engineer walks through a virtual project environment to explore the impact of a new requirement on the project, the paths and discussions that are taken are simultaneously packaged as traceable rationale for any decisions implemented, according to the project's and organization's potentially changing requirements for traceability. There is a real-time intention for traceability on the project, which is specified and complied with at all times by all team members.

6.2 Problems Addressed (Configurable)

The traceability solution is generally fixed upfront for a project and rigid thereafter. Once a traceability information model and an enabling process have been agreed to on a project (if at all), it can be problematic to change the particulars mid-project. Even when the traceability process is pre-defined and agreed upon, it is often implemented inconsistently in and across teams, irrespective of whether the team is co-located or distributed. Furthermore, when the stakeholder requirements for traceability change or the implementation specifics change, not all of the stakeholders may be notified. With time, the manner in which the traceability is established on a project can drift from the specified intent. A typical concern that is a common barrier to technology transfer of new traceability techniques, methods and tools in industry is whether research-initiated techniques can actually be configured to fit real project needs and circumstances as they emerge.

6.3 Dream Process (Configurable)

- ***Traceability Strategy.*** A traceability planning and management tool will automatically create a project-specific traceability solution with an underlying traceability information model and process that reflects stakeholder requirements for traceability. It will also provide an interactive traceability dashboard that will allow this all to be re-configured in real-time.
- ***Traceability Creation and Maintenance.*** Traces will be identified and created based upon a project's traceability information model and its actual artifacts, and they will be compliant with this definition of traceability intent. Traces will then be self-maintained such that they align with what is defined in a project's traceability information model at all times.
- ***Traceability Use.*** Semantically rich traceability will be personalized to satisfy individual needs for end-use at all times, by dynamically reconfiguring and re-purposing existing traces as needed.

6.4 Goals (Configurable)

Configurable G 1 — The intended traceability is defined for a project, using rich semantics for trace links, and any changes to these intentions are reflected.

Configurable G 2 — The traceability solution on a project complies with the definition of intent, accommodating diverse and potentially changing needs at all times.

Configurable G 3 — Proactive prediction provides support for determining and accommodating future stakeholder requirements for traceability, adapting the specification of intended traceability, updating the pre-existing traceability solution and reconfiguring existing traces over time as needed.

Configurable G 4 — Levels of compliance are defined so as to either relax or tighten the traceability that is established on a project, thereby configuring the extent to which it is necessary to comply with the intended traceability at different times, for differing artifacts or by differing stakeholders.

6.5 Requirements (Configurable)

6.5.1 Traceability Strategy (Configurable)

<u>*Configurable Req 1*</u> To define the intended traceability for a project as an integral part of the traceability solution. [Configurable G 1]

- *Status:* Researchers advocate that the intended traceability for a project be defined within a semantically rich traceability information model or meta-model. Such a model defines the trace artifact types and their associated trace link types based upon the analyses made possible by traversing these traces. The state of the practice is that traceability information models, if built, are typically rudimentary and their trace links are rarely semantically typed. The potential of using rich semantics is thus seldom exploited in traceability-related queries and end-use. While there are some domain-specific traceability information models, it appears that many practitioners have yet to be convinced of their value. High-level goals tend to be provided to explain the purpose of traceability information models, rather than actual guidance in their construction and use.
- *Promise:* Research has emphasized simple and pragmatic traceability information models recently, so some flow-through to industry is expected. The Tracy project further proposes to include a downloadable traceability information model tool for practitioners to configure and use, potentially facilitating uptake.

<u>*Configurable Req 2*</u> To define variable levels of granularity in the intended traceability, to accommodate different stakeholders and artifacts, and to account for differing parts of a system at different times in a project's life (i.e., heterogeneous solutions to heterogeneous needs). [Configurable G 1, G 4]

- *Status:* Traceability solutions are typically designed to be homogeneous (i.e., one size fits all). Research has not addressed variability in the traceability solution, so tools rarely support this. Traceability information models, where created, rarely come in a heterogeneous and partitioned form either.
- *Promise:* Finer-grained and parameterized traceability information models, tailored to different project contexts and needs, may enable variability. Individual requirements may demand different levels of traceability based upon their value and volatility, so risk-driven provisioning may be worth investigating.

<u>*Configurable Req 3*</u> To use the definition of the intended traceability to provide traceability process guidance, and to undertake compliance and consistency checks in the actual implementation of the traceability process across team members and other project constituents. [Configurable G 2, G 4]

- *Status:* A number of commercial tools offer assistance to define the intended traceability on a project and then to enforce compliance and consistency in its implementation. Process compliance and consistency management is already a mature topic in other branches of software and systems engineering.
- *Promise:* Process-aware integrated development environments that monitor the current state of a project and, when coupled with a well-defined traceability information model, provide guidance and feedback on the traceability that is implemented in real-time. Using a definition of the intended traceability on a project more habitually would enable such compliance checking and consistency management.

<u>*Configurable Req 4*</u> To adapt the definition of the intended traceability, and any associated process, to accommodate changing contexts and needs. [Configurable G 1, G 2, G 3]

- *Status:* Traceability information models, where defined and used, seldom come in an evolvable form. They can, therefore, be difficult to change retrospectively. Research has not addressed subsequent changes to the traceability information model and process, so tools rarely support this evolution.
- *Promise:* The concepts underpinning self-managing and adaptive systems, along with techniques from autonomic computing, are likely to play an important role in the required re-configurability of traceability solutions.

6.5.2 Traceability Creation and Maintenance (Configurable)

<u>*Configurable Req 5*</u> To create and maintain traces that comply with the intended traceability for a project, whenever, however and wherever these traces are established. [Configurable G 2, G 4]

- *Status:* Research proposes defining traceability information models to guide the creation of valid traces. Such models help to check the validity of the trace links that have been created, and tools can enforce this checking, but they do not readily help in capturing the trace links in the first place. Semantics may be attached to trace links in practice, by putting attributes on trace links in leading requirements management tools, but these semantics are often minimal, inconsistently applied and not always subsequently exploited in traceability end-use.
- *Promise:* Using a definition of the intended traceability on a project, as specified in a semantically rich traceability information model, to guide the actual discovery and creation of trace links, and then to guide ongoing trace maintenance activities.

<u>*Configurable Req 6*</u> To assess whether there is a need to remove and re-create existing traces when the definition of the intended traceability changes on a project, as an alternative to maintaining versions of existing traces. [Configurable G 3]

- *Status:* Where the context of a project changes, such as the introduction of new audit requirements in an industry or the reuse of an existing project's artifacts and associated traceability in a completely new project, the traceability remains relevant to the prior context. No research has investigated switches of context mid-project or in reuse situations for its ramifications with respect to trace validity and ongoing trace maintenance.
- *Promise:* In theory, the established traceability can be checked against its traceability information model at any time, where one exists on a project, and any discrepancies can either be noted or rectified. In prac-

tice, such models are infrequently used in this way beyond initial trace creation and then for ongoing maintenance, but this support would be a simple and natural progression.

6.5.3 Traceability Use (Configurable)

Configurable Req 7 To use models of the end-user, the wider end-use process and end-user traceability-related queries to guide the fine-grained definition of the intended traceability on a project. [Configurable G 1, G 3]

- *Status:* Researchers advocate that traceability information models be constructed that reflect and enable the answering of end-user traceability-related queries. But, because there is an incomplete understanding of the various end-users of traceability at present, their task queries are not routinely used to define traceability information models in practice. However, studies of how users use traces and models of the end-use process are both emerging.
- *Promise:* Operational profiles indicate where to focus the testing effort in software and systems development. A similar profile of intended end-use could lead to defining a profile for the traceability focus on a project, allowing for variation in both its specification and implementation over time and contexts.

Configurable Req 8 To monitor end-use to predict future needs and re-configure the definition of the intended traceability as needed. [Configurable G 3]

- *Status:* Since there is seldom a feedback loop from traceability in actual use back to the original intentions, the traceability that is created and maintained is rarely adjusted moment-to-moment. It is not clear whether this would even be a cost-effective approach.
- *Promise:* Data collected on end-use, both historical and real-time, may provide insights into likely future needs and enable the development of probabilistic end-use models. There may also be some scope for end-users to define and manipulate their own traceability needs and models.

Configurable Req 9 To adapt pre-existing traces to address end-user requirements for traceability dynamically. [Configurable G 3]

- *Status:* Where implemented, trace links are generally hard-wired to provide support for particular predefined uses in practice and are rarely reconfigurable to support new contexts of traceability use.
- *Promise:* With advances in monitoring, and in autonomic techniques and technologies, traces could be self-aware and adapt to changing demands. Smart traces would assist with the reuse and repurposing of traces for new end-uses.

6.6 Recommended Research (Configurable)

The major research theme to achieve configurable traceability is ***to use dynamic, heterogeneous and semantically rich traceability information models (or similar specifications of the intended traceability) to guide the definition and provision of traceability.*** Supporting research topics are listed below.

Research ID	Description	Req ID
Configurable RT 1	Provide better ways to define the traceability that is required on a project, accommodating varying levels of granularity and rich se-	Configurable Req 1, 2, 7

	mantics to account for differing tracing needs, artifacts and stages of the project lifecycle. This could be via traceability information models or other specification concepts.	
Configurable RT 2	Provide a mapping from the traceability information model (or similar specification concept) to its instantiation on a project, so as to support change and enable compliance checks and consistency management in its implementation.	Configurable Req 3, 4, 5
Configurable RT 3	Investigate techniques to automatically propose traceability information models (or similar specification concept) based upon an analysis of stakeholders' requirements for traceability and the projected project artifacts in various organizations and domains.	Configurable Req 7, 8
Configurable RT 4	Investigate how to reconfigure or re-purpose a pre-existing set of traces to accommodate changes in the definition of the traceability information model (or similar specification concept) – i.e., smart trace links.	Configurable Req 6, 9

6.7 Positive Adoption Practices for Industry (Configurable)

Configurable IP 1 Practitioners use a traceability information model (or similar specification concept) to define and update their traceability intentions for a project. This definition and use process will be supported and form an integral part of the traceability solution.

Configurable IP 2 Practitioners work on global and distributed projects establishing traceability consistently and as intended (which may not mean homogeneously) irrespective of locale.

Configurable IP 3 Practitioners change their particular approach to traceability as their needs and context dictate, yet comply with the traceability of other practitioners.

[continued…]

7 Traceability Challenge 4: Traceability that is Trusted

All stakeholders have full confidence in the traceability, as it is created and maintained in the face of inconsistency, omissions and change; all stakeholders can and do depend upon the traceability provided.

7.1 Link to Vision (Trusted)

In the vision scenario, the engineer is confident in making decisions based upon the options presented to her. She trusts the results of the traceability and expects the associated analyses it enables to be accurate and up to date at all times. The engineer is alerted to the impact on traceability of potential changes in the requirements and their implementation, and any necessary traceability updates for the changes that are implemented are made proactively, meaning that this confidence in the traceability is retained. The traceability simply self-repairs and evolves at all times without the engineer having to do anything explicit. The engineer is also comfortable in delegating any ensuing tasks that will impact the traceability, as she trusts that the overall traceability will not be jeopardized by others' actions or inactions. The traceability is always dependable; it is 'ready-to-use' by the engineer and even 'ready-to-wear' on her sweater sleeve.

7.2 Problems Addressed (Trusted)

The traceability that is established on many projects often has a dubious provenance, impacting how much trust can be placed in the analyses it facilitates, as well as its longevity. People establishing traceability make mistakes that go undetected and the impact of such mistakes are rarely known. Traces decay unless they are cultivated, but the useful life and quality of the trace links is usually also unknown. The traced artifacts can themselves expire and this can remain unknown, with unforeseeable consequences. Without effort, there is traceability entropy over time. This is a vicious cycle for both establishing and using traceability – why keep the traceability current if it is already flawed and why use it? Practitioners are not going to invest in something that they do not find trustworthy or that demands inordinate housekeeping effort from them to keep it dependable and credible.

7.3 Dream Process (Trusted)

- ***Traceability Strategy.*** An up to date quality profile for all the traces established and used on a project will be planned for and made available at any moment in time.
- ***Traceability Creation and Maintenance.*** Every trace that is created will have associated quality metrics. Once created, every trace will be guaranteed to a defined quality level and strive to retain its own ongoing integrity, despite changes in the system and artifacts, and its quality metrics will be updated accordingly if necessary.
- ***Traceability Use.*** Only trusted traces will be used to support different traceability-enabled tasks on a project. The end-user will trust the traceability and depend upon its analyses.

7.4 Goals (Trusted)

Trusted G 1	The factors that impact the quality of the traceability process and product are known and factored into traceability strategies.
Trusted G 2	The quality of the traceability is measured on a project, at an individual trace level and at a trace set level, and this information is provided to all stakeholders.
Trusted G 3	Degrees of confidence in the analyses provided by the traceability are calculated and this information is provided to all stakeholders.
Trusted G 4	The traceability is self-healing, so its quality is preserved in the face of change, or updated where adjusted.

7.5 Requirements (Trusted)

7.5.1 Traceability Strategy (Trusted)

<u>*Trusted Req 1*</u> To agree upon metrics to define the quality, both required and actual, of all aspects of the traceability process and product. [Trusted G 2, G 3]

- *Status:* Research on automated trace recovery and trace capture has made wide use of a number of quality metrics common in the information retrieval discipline, such as for the recall and precision of trace links. Other than these and their associated metrics, there are few agreed upon measures for traceability quality.
- *Promise:* Precision and recall metrics are only a start, and quantitative measures of traceability process and product quality will only take us so far. Qualitative and probabilistic measures of traceability quality will need to be added to provide for a mix of measures.

<u>*Trusted Req 2*</u> To account for levels of completeness, correctness, consistency, etc. in the various trace elements when planning and managing a traceability solution. [Trusted G 1, G 2]

- *Status:* The artifacts to be traced are seldom ‘perfect’. Researchers have focused on the quality of the trace links, more so than the quality of the trace artifacts to date, but the quality of the overall traceability is part determined by the quality of those artifacts being linked and traced. There is rarely any discussion on artifact quality and its ramifications on the traceability, and little ‘cleaning’ of the artifacts to be traced takes place in practice.
- *Promise:* If you link garbage you retrieve garbage. Those artifacts being traced need to be of an acceptable quality standard (i.e., accurate, complete, up to date, consistent, etc.). Or, where artifact quality is lacking, their quality attributes need to be understood and taken into account. Improvements in development practices, coupled with agreed upon quality metrics for traceability, will be important here. Advances will come from more focus on writing better requirements and by improving the other engineering artifacts to be traced, and by providing real-time feedback on their potential traceability at the time at which these artifacts are created.

<u>*Trusted Req 3*</u> To measure all aspects of the traceability process for completeness, correctness, consistency, etc., based upon agreed metrics. [Trusted G 1, G 2]

- *Status:* The quality of the traceability process itself is even less examined than the quality of the elements forming the traces. Process quality measures are not routinely integrated into the traceability strategy, limiting the potential for informed traceability process improvement.
- *Promise:* The use of process data and quality measures to advance the quality of the trace product, as is common practice in general process improvement, would provide a mechanism for traceability process improvement. Levels for such improvement could also be defined along the lines of the more general capability maturity models.

<u>Trusted Req 4</u> To understand the nature and impact of human vulnerability on all aspects of the traceability process, and to build in suitable mitigation strategies to address them. [Trusted G 1, G 2, G 3, G 4]

- *Status:* When creating and maintaining traceability manually, humans can err in their decisions, actions and inactions. When traces are created automatically, humans may not always trust the process that was used to create the traces, impacting their likelihood to use them. Furthermore, when performing certain traceability-enabled tasks in practice (such as impact analysis where it is essential to discover each and every impacted component), any incompleteness or error in the traces created (either manually or automatically) may lead the end-users to mistrust other traces created in the same manner, especially where they are led to believe that the traces will be complete. Little attention has been paid to the impact of human involvement and trace confidence levels in all aspects of the traceability process.
- *Promise:* Models of human involvement in the traceability process are needed to gain a greater understanding of the potential value humans add to the process and the bottlenecks they present. Researchers are now beginning to look at the 'humans in the loop' and more studies of this nature are essential.

<u>Trusted Req 5</u> To use the traceability itself to understand and strengthen the quality of the traceability on a project (i.e., traceability bootstrapping). [Trusted G 2, G 4]

- *Status:* The traceability that is already established on a project can itself be used to help identify some quality attributes, such as the completeness of the traceability via an examination of missing artifacts. Researchers are also beginning to study what can be learned about the traceability from both the presence and the absence of traces.
- *Promise:* Using traceability analyses to advance traceability quality may present some interesting opportunities for traceability bootstrapping.

7.5.2 Traceability Creation and Maintenance (Trusted)

<u>Trusted Req 6</u> To define and agree upon standards to create and maintain quality traces. [Trusted G 2, G 4]

- *Status:* Researchers informally agree upon what would be acceptable values for potential traceability quality metrics, such as for the recall and precision associated with automated traceability creation. In an attempt to reach such quality targets, recent research combines automated techniques to identify candidate trace links with voting-based mechanisms to improve and bolster the confidence in the quality of the traces created.
- *Promise:* Reaching agreement upon how the quality of a trace and its component elements are defined, and establishing benchmark experiments and datasets to compare techniques, methods and tools for their creation and maintenance against baselines.

<u>*Trusted Req 7*</u> To gather requisite data for both traceability quality assessment and the future upkeep of this quality at the time of a trace's creation. [Trusted G 2, G 4]

- *Status:* Researchers have paid much attention to boosting the confidence levels with automated traceability creation, using the concept of 'candidate links' and by setting thresholds for selecting among them.
- *Promise:* Providing suitable semantics and meta-data to clarify the quality attributes of a trace at the point of its initial creation and at every stage in the traceability maintenance process. This relies upon gaining progress, more generally, with agreeing upon quality metrics for traceability.

<u>*Trusted Req 8*</u> To understand the impact of the familiarity of the stakeholders who establish the traceability with the artifacts under trace (i.e., where stakeholders are less familiar with the code, there may be less trust in their ability to trace the design to the code). [Trusted G 1]

- *Status:* The quality of the traceability is, in part, determined by the person or the tool doing the tracing, and that topic has received limited attention to date. Equally, with the emergence of more automated approaches, researchers have not yet determined whether people trust automatically created traces more than manually created ones. With automatically created traces, practitioners still need to take the time to approve the candidate trace links to assure confidence in the trace link. This means that automated approaches still necessitate human skills in the loop at present.
- *Promise:* Empirical studies of the role of human involvement in the traceability process are emerging and more such studies are needed.

<u>*Trusted Req 9*</u> To monitor for any kind of change that impacts the quality of the traceability. [Trusted G 1, G 4]

- *Status:* In practice, the validity of traces expires and becomes obsolete, and this is not always accounted for in practice. This leads to a degradation of trust in the traceability over time. Research into the automated maintenance of traces assigns a status of 'suspect' to previously created trace links that change and in which confidence has been lost. Each suspect trace link demands user confirmation on subsequent actions to perform, while unambiguous updates can only sometimes be performed in the background.
- *Promise:* Techniques that identify potentially obsolete trace links, along with support to update, version and archive these trace links, are needed to retain the traceability quality. This includes the propagation of updates to ensure that the overall traceability remains credible. Initial work based upon event-based and rule-based maintenance is promising.

<u>*Trusted Req 10*</u> To understand the process of traceability decay, and to predict and measure the useful life of a trace. [Trusted G 1, G 4]

- *Status:* It is currently a costly proposition to maintain all the traces previously created during a development project. It is also not really known how the quality of each individual trace impacts the overall traceability quality and so it is uncertain as to which trace links really deserve the attention.
- *Promise:* The life expectancies of different traces probably vary and it may not be necessary to maintain and preserve them all. A triage-based approach to traceability maintenance would identify those traces that can be thrown away and those that can maintain themselves satisfactorily, relieving time to focus on those that need to be maintained more explicitly and on a case-by-case basis.

7.5.3 Traceability Use (Trusted)

<u>*Trusted Req 11*</u> To define the necessary and acceptable quality for different traceability-enabled end-user tasks. [Trusted G 3]

- *Status:* The quality required of the traceability to support the various end-users and their tasks is rarely articulated in practice, chiefly because the tasks themselves have not been specified (traceability challenge one).
- *Promise:* The required traceability quality is unlikely to be a fixed value across people, projects, tasks and time, so this needs to be articulated. This depends upon progress with traceability challenge one.

<u>*Trusted Req 12*</u> To present confidence levels for the traceability and the analyses it enables to the end-users, with respect to its suitability for different tasks. [Trusted G 3]

- *Status:* All trace links are usually presented as equal in practice (i.e., they either exist or they do not exist). It can also be difficult to assess whether a trace link is up to date or not. Some trace visualizations explore the use of color to suggest the age and likely relevance of trace links to assist in their end-use.
- *Promise:* Further visual mechanisms to render the quality of the traceability visible to end-users, and to indicate the suitability for various tasks, are needed. Traces are sometimes going to be less than perfect, so the promise also lies in making the best use of such traces and ensuing that the risks of this use is made visible.

<u>*Trusted Req 13*</u> To retrieve the most current trace with respect to an end-user query, reflecting real-time dependencies between the latest artifacts. [Trusted G 3, G 4]

- *Status:* To boost the quality and credibility of trace analyses, these must be based upon up to date trace artifacts and trace links, unless the analyses are historical in nature. Version control systems allow for such fine-grained control of artifacts and their dependencies.
- *Promise:* Version control systems are mature technologies and improvements in this area will be of continued value to traceability advances.

<u>*Trusted Req 14*</u> To accommodate or repair breaks in the traceability record, so that the quality status of the traceability is always made evident to the end-user. [Trusted G 2, G 4]

- *Status:* Automated techniques enable traceability to be recovered afresh on end-use request if traces are missing or problematic, but the difficulty lies in identifying that either a trace is missing or has been compromised in the first place.
- *Promise:* More attention to monitoring the quality of traces over time is essential. This relies upon quality metrics and knowledge of the quality levels required to support various end-user tasks. The quality could be repaired dynamically during end-use if any issues are encountered. Requirements monitoring research is already in evidence and could lend insight here.

<u>*Trusted Req 15*</u> To provide a link to those people who have contributed traced artifacts or have created trace links, to enable the end-user to assess whether they are trusted entities and to do further checks on quality concerns (in person) when needed. [Trusted G 1, G 2, G 3]

- *Status:* Practitioners often infer trust in traced artifacts and trace links based upon who created and who maintained them (i.e., the quality of a product is a reflection of the process and the people undertaking the process). Equally, where the traceability provided at the point of end-use is confusing or deficient in some way, sometimes the only resort in practice is to talk to the people who established the traces. Some research has proposed tying in the social production network underlying the traceability network to enable this support, such as by modeling the social network underlying the creation and maintenance of traceability (e.g., contribution structures).

- *Promise:* Further integration of social network modeling approaches and analyses into the traceability process is desirable here.

Trusted Req 16 To provide a way for end-users to exchange data about the perceived and actual quality of a trace and of the analyses provided following the end-use of a trace. [Trusted G 2, G 3]

- *Status:* There is little research into those mechanisms to help identify and alert end-users to mistakes or problems in the traceability (i.e., incorrect or missing traces), in turn to provide experiential quality data to factor into traceability analyses. However, most contemporary development environments now include integrated emailing and chat capabilities, discussion forums, etc. for developers to communicate about the development process, and sometimes these are being used to support traceability in these ways.
- *Promise:* Further exploitation of integrated communication capability within integrated tooling holds promise, to enable all stakeholders to report on quality issues in both establishing and using traceability.

7.6 Recommended Research (Trusted)

The major research theme to achieve trusted traceability is ***to perform systematic quality assessment and assurance of the traceability.*** Supporting research topics are listed below.

Research ID	**Description**	**Req ID**
Trusted RT 1	Develop a model of the vulnerabilities in the traceability process, including human error in both manual and automated approaches, and develop suitable techniques to reinforce their reliability.	Trusted Req 4, 8, 15, 16
Trusted RT 2	Formulate metrics for traceability quality assessment, especially for the traces that are created and maintained.	Trusted Req 1, 3, 6
Trusted RT 3	Gain improvements in the quality of both manual and automatically created and maintained trace links.	Trusted Req 2, 5, 6, 7, 9, 13, 14
Trusted RT 4	Provide ways of inferring trust in the traceability based upon how the trace links are established and used, and by whom, and upon the useful life expectancy of traces.	Trusted Req 4, 8, 10, 15, 16
Trusted RT 5	Create a visual dashboard for displaying and examining traceability quality attributes on a project.	Trusted Req 2, 3, 7, 9, 12, 15, 16
Trusted RT 6	Catalogue the quality required of the traceability for supporting different end-user tasks within the Traceability Body of Knowledge (TBOK).	Trusted Req 11

Trusted RT 7	Gather empirical evidence as to the quality of traceability techniques, methods and tools with respect to the quality of the traces they enable within the Traceability Body of Knowledge (TBOK).	Trusted Req 2, 3, 6, 7, 8, 16
Trusted RT 8	Advance the run-time monitoring of traceability quality with validated error detection models for trace links.	Trusted Req 7, 9
Trusted RT 9	Apply concepts from autonomic computing to explore self-healing traceability techniques, methods and tools, covering diagnosis, repair actions and propagation, to apply at both the individual trace and trace set levels.	Trusted Req 5, 14

7.7 Positive Adoption Practices for Industry (Trusted)

Trusted IP 1 Practitioners routinely specify acceptable levels for traceability quality attributes for their end-user tasks.

Trusted IP 2 Practitioners are provided with the data they need to determine whether they can trust the traceability techniques, methods and tools that they use and the analyses that are based upon their end-use.

Trusted IP 3 Practitioners supply feedback on the quality of the traceability unobtrusively and as part of its creation, maintenance and end-use.

[continued…]

8 Traceability Challenge 5: Traceability that is Scalable

Varying types of artifact can be traced, at variable levels of granularity and in quantity, as the traceability extends through-life and across organizational and business boundaries.

8.1 Link to Vision (Scalable)

The engineer has an enormous quantity of data that is rendered traceable in the vision scenario: eleven months of fine-grain project artifacts, links back to past archives containing other project artifacts, full records of project rationale and context, etc. The traceability that the engineer makes use of accounts for a myriad of artifact types, such as requirements, live links to stakeholders and contributors, test cases and government regulations. The engineer can rely upon the traceability having been established from the onset of her development project, through its transition into a maintenance project, to the eventual project closure and system retirement.

8.2 Problems Addressed (Scalable)

Traceability is often an afterthought on projects and established when it is needed, rather than from the first days in which project artifacts begin to accumulate. Pre-requirements artifacts can therefore be missed and remain untraceable. Likewise, traceability can erode over time unless the transition of traceability from a development project into its maintenance phase is also planned for. It is often difficult to account for the entirety of the artifacts relevant to development in the traceability, notably multimedia and unstructured informal artifacts. The traceability can become complex to depict and hence unusable over time. Some datasets are intrinsically difficult to trace due to inconsistencies in terminology, the nature of the artifact types, the lack of structure and heterogeneous formats. Non-functional requirements that have a global impact on the system are also notoriously difficult to trace. Traceability processes, techniques and methods tend to break down with scale in its various dimensions (e.g., the quantity of traceable artifacts or trace links, and time). Practitioners are reluctant to use new and emerging techniques, methods and tools without evidence of scalability in these multiple dimensions. The issue of scale can be compounded where customers mandate traces without discerning attention to their intended end-use.

8.3 Dream Process (Scalable)

- ***Traceability Strategy.*** Full lifecycle and all-embracing traceability will be planned for and managed, and any scale issues will be reduced via auto-completion tools.
- ***Traceability Creation and Maintenance.*** Trace creation will be as fast in large projects as it is in small ones, linking anything within its scope without a performance hit. Traceability maintenance will also be as fast in large projects as it is in small ones, and the traceability will not entropy over time.
- ***Traceability Use.*** End-users will only see what they need to see from among a mass of project artifacts when they use traceability, and they will switch between coarse-grain and fine-grain traceability routinely.

8.4 Goals (Scalable)

Scalable G 1	There are no practical limits to the quantity of traceable artifacts and trace links that can be created and maintained in a project.
Scalable G 2	All media and artifact types serve as potentially traceable artifacts.
Scalable G 3	Traceable artifacts are 'zoomed' into as required, to trace at varying levels of granularity.
Scalable G 4	Full project lifecycle traceability coverage and longevity of this coverage is provided throughout a system's life, extending across organizations and business entities.

8.5 Requirements (Scalable)

8.5.1 Traceability Strategy (Scalable)

Scalable Req 1 To plan and manage traceability from the first day of a project until the last day of the project. [Scalable G 1, G 2, G 4]

- *Status:* Traceability is sometimes not implemented in practice until it is needed, or it is truncated to cover a period of a project's life, such as from requirements to design, or from requirements to code. This is often a side effect of the disparate tools being used.
- *Promise:* Late or restricted implementation of traceability is often a consequence of the investment required upfront on a project, coupled with unclear cost-benefit studies. Progress here will depend upon progress with traceability challenge two.

Scalable Req 2 To set up an open system to accommodate multiple types of trace artifacts and trace links. [Scalable G 2, G 4]

- *Status:* Traceability is often planned for in a homogenous manner on projects, irrespective of the project artifacts and project size, so many artifacts can thus be excluded from traceability support. Traceability is primarily planned for and applied on code, textual descriptions (e.g., natural language requirements) and UML (Unified Modeling Language) artifacts at present. Nevertheless, industrial researchers are piloting the traceability of heterogeneous artifacts in very large projects with some success.
- *Promise:* Designing approaches to traceability based upon traceability abstractions, rather than concrete artifacts types, which can accommodate all the artifacts that are likely to arise in the life of a project.

Scalable Req 3 To specify the concept of granularity, formally, to provide a way to define and retrieve the levels of granularity required for traceability on a project. [Scalable G 3]

- *Status:* Researchers have proposed establishing macro and micro levels of traceability to accommodate diverse media types, promoting the concept of granularity layers in the traceability provided, but this has not yet been fully developed or adopted in practice. Granularity remains an informally defined concept, with no real consensus on what is actually meant by fine-grain and coarse-grain, and all the levels in between. At present, a trace artifact accounts for both a full requirements document and an individual word within a requirement statement.

- *Promise:* Trace link semantics have received a great deal of attention by the research community and more use of such will find its way into practice via rich links. A more discerning ontology for specifying trace artifacts is now equally needed.

Scalable Req 4 — To understand how traces are needed and used across organizations and business entities (i.e., accounting for subcontractors, etc.) and accommodating broader needs in the traceability strategy. [Scalable G 4]

- *Status:* Stakeholder requirements for traceability are poorly understood at present (see traceability challenge one).
- *Promise:* Progress with traceability challenge one is essential to progress here. However, this needs to take care to examine additional stakeholders beyond the obvious candidates to examine the breath of artifacts to be traced.

Scalable Req 5 — To apply traceability practices and processes to large, distributed, multi-person, multi-year projects. [Scalable G 1, G 4]

- *Status:* To support distributed contexts requires that the traceability does not decay as changes are made to interrelated and externally maintained artifacts over time. In practice, a lack of seamless bi-directionality of the traceability across all the possible tools that produce and hold the traced artifacts can compound the update of traceability following changes made to any associated external artifacts. This issue is usually addressed where projects use a single and shared application lifecycle tool.
- *Promise:* The growing interoperability of tools and data offers promise here because standardization on a single tool across a distributed multi-organizational setting may not always be viable. There are also dependencies that can reduce the scalability problem to a smaller, more manageable problem, such as exploiting the transitivity properties among trace links. If one trace can partially or fully imply another trace, then this can be reasoned about and be potentially supported by auto-completion strategies.

Scalable Req 6 — To understand the particular scale issues associated with tracing the global properties of systems, such as non-functional requirements, and with tracing in the context of systems of systems. [Scalable G 1, G 4]

- *Status:* The traceability of non-functional requirements is receiving attention in the research community, as the perception is that the traceability of such global properties is more complex and difficult to handle. Differentiating the particular nuances of systems of systems development, for tracing purposes, has received less attention to date.
- *Promise:* Early industry and government adopters of automated trace recovery techniques have made datasets available for research into the issues associated with the scalability of traceability techniques, methods and tools. The issues of local and global traceability could be examined in such contexts to gain a clearer understanding of the different issues with scale in these two increasingly important dimensions.

8.5.2 Traceability Creation and Maintenance (Scalable)

Scalable Req 7 — As scale grows, to maximize the use of automated traceability creation and maintenance. [Scalable G 1]

- *Status:* Pilot studies have been conducted in large industrial projects to examine the scalability of automated trace recovery techniques with promising results. The recall measure for trace links recovered via automated techniques is now generally acceptable, even on large datasets. The validation of automated maintenance techniques is still mostly restricted to small datasets at present.

- *Promise:* To provide for a viable approach to completely automated traceability creation and maintenance in large projects over time, the method of automation may need to be differentiated according to the criticality of the artifacts. For example, traces might be created as a by-product of formal specifications for highly critical components, while traces might be created using trace retrieval methods for less critical components.

Scalable Req 8 To create and maintain trace links between artifacts of different types, in terms of their media, formality, level of structure, etc., and at any level of granularity [Scalable G 2, G 3]

- *Status:* The focus of traceability creation has been from requirements through to code to date (i.e., post-requirements traceability). There has been limited research on the indexing and retrieval of informal, unstructured and multimedia artifacts in a software and systems development context, so they are often not included as potential traceable artifacts in traceability solutions that adopt automated traceability creation techniques and methods. Traceability maintenance techniques and methods also deal primarily with structured textual artifacts, UML diagrams and code. In general, there has been less focus on accounting for pre-requirements artifacts in traceability solutions by researchers or practitioners, though some recent industry attention has been on tracing back to regulatory codes.
- *Promise:* In theory, any artifact that can be indexed can be traced, so more attention needs to be paid to developing ontologies for describing different types of traceable artifact. Navigating and presenting the resulting traces also demands rendering these artifacts in some way, so this requires progress with trace visualization.

Scalable Req 9 To prune the growing mass of traceable artifacts and trace links to keep trace maintenance and trace retrieval manageable. [Scalable G 1]

- *Status:* Research has focused on accumulating trace links rather than on pruning them. Trace links are rarely retired in practice, potentially impeding future traceability as they grow in number. Trace links need versioning and garbage collection if the traceability is to scale, and well-known versioning systems are increasingly a core component of many traceability solutions.
- *Promise:* The versioning and garbage collection techniques common to other areas of software and systems engineering need to be applied more widely within traceability solutions.

8.5.3 Traceability Use (Scalable)

Scalable Req 10 To retrieve and filter trace artifacts, potentially represented as diverse media types, to address traceability-related queries. [Scalable G 1, G 2]

- *Status:* There has been limited analysis on how to exploit artifacts of different media types in trace retrieval algorithms, so presenting traces containing multiple media artifacts is not standard. There can be performance issues associated with using traceability in large datasets in practice, an issue compounded by the presence of rich media artifacts.
- *Promise:* Ongoing improvements in multimedia search, retrieval and filtering will make media-rich traces increasingly feasible in the future. However, the actual need for and value of media-rich traces requires more empirical study.

Scalable Req 11 To provide visual mechanisms to augment large-scale traceability in end-use, switching between coarse-grain views of traceability (i.e., broad) and fine-grain views of traceability (i.e., deep) with ease. [Scalable G 1, G 3]

- *Status:* End-users often need to untangle a mass of trace links in order to make use of them in practice. Commonly used visual mechanisms, like traceability matrices, while wholly appropriate for many traceability-enabled activities and tasks, do not scale. Researchers are beginning to focus on visualizations for trace links, to overcome their complexity in actual use, mostly appearing in prototype tools at present. However, there are few usability studies on the use of such emerging visuals, particularly for handling the traceability of large datasets.
- *Promise:* The improved visualization of traces will facilitate their end-use and make the resulting analyses more accessible to end-users. Layered approaches to traceability, building on similar concepts to those seen in computer-aided design tools, where layers can be turned on or off depending on need, would help to provide filters and so address some of the issues associated with scale.

8.6 Recommended Research (Scalable)

The major research theme to achieve scalable traceability is ***to provide for levels of abstraction and granularity in traceability techniques, methods and tools, facilitated by improved trace visualizations, to handle very large datasets and the longevity of these data.*** Supporting research topics are listed below.

Research ID	Description	Req ID
Scalable RT 1	Obtain industrial datasets from various domains to enable researchers to investigate scalability issues, and the potential of techniques, methods and tools to address them, both systematically and comparatively.	Scalable Req 5, 6, 7, 9
Scalable RT 2	Develop effective search, filtering and visual mechanisms to navigate and query large numbers of trace artifacts and trace links, of varying media types.	Scalable Req 2, 5, 10, 11
Scalable RT 3	Develop an abstract model of the traceability process and its component activities, to enable pluggable techniques, methods and tools that apply to differing process activities and differing layers of abstraction to be created, located and used.	Scalable Req 1, 2, 3, 5
Scalable RT 4	Develop a cost-benefit model to assess granularity decisions that impact subsequent scale issues with respect to traceability.	Scalable Req 1, 3, 8
Scalable RT 5	Provide techniques to evaluate the traceability potential of various datasets and media assets, and to guide in setting up a suitable traceability strategy to accommodate them.	Scalable Req 1, 8
Scalable RT 6	Gain improvements in performance for the real-time automated recovery and capture of trace links to account for scale.	Scalable Req 5, 7
Scalable RT 7	Gain improvements in performance for the real-time retrieval and rendering of traces to account for scale.	Scalable Req 5, 10, 11
Scalable RT 8	Define ontologies for software and systems development artifacts, and investigate the need for and value of integrating the various artifact types and media into traceability end-use.	Scalable Req 2, 8

Scalable RT 9	Explore the unique scalability issues associated with tracing non-functional requirements, and develop effective techniques, methods and tools for this context.	Scalable Req 6
Scalable RT 10	Explore the unique scalability issues associated with tracing within and across systems of systems, and across organizational and business boundaries, and develop effective techniques, methods and tools for this context.	Scalable Req 4, 6

8.7 Positive Adoption Practices for Industry (Scalable)

Scalable IP 1	Practitioners establish traceability from the onset of a project, along with the house-keeping procedures that are needed to keep the traceability use viable through to project completion.
Scalable IP 2	Practitioners take a multi-pronged approach to establish traceability, to account for all project artifacts over time, but the unique details remain hidden behind a simpler and more abstract treatment of the artifacts.
Scalable IP 3	Practitioners switch seamlessly between 2D and 3D visualizations as they walkthrough multimedia-rich traces at varying levels of granularity.
Scalable IP 4	Practitioners contribute datasets to enable researchers to examine scalability issues with emerging traceability techniques, methods and tools.

[continued…]

9 Traceability Challenge 6: Traceability that is Portable

Traceability is exchanged, merged and reused across projects, organizations, domains, product lines and supporting tools.

9.1 Link to Vision (Portable)

Traceability is merged across all components of the full flying solar car system in the vision scenario, where software is but one component of the system, and requirements from related projects are reused, along with their entire traceability networks. Interrogating the traceability networks of external software systems and services aids the engineer's decision making regarding procurement. The engineer integrates a new service into the existing system with the confidence that the traceability back to the requirements will facilitate both the uncoupling of the expired software and the integration of the new service, and so provide the team in South Africa with all the information that they need to complete the update. The entire traceability history is always available for use and reuse, irrespective of where the actual traces were created and the tools that were used to create them.

9.2 Problems Addressed (Portable)

Traceability is often legacy and locked into projects and tools, so it is rarely extractable and reusable across projects or components therein. It is also typically project, organization and person-specific, so difficult to reconcile in a timely manner. Standards are rarely used across more than locales and, where they are used, they can be applied somewhat inconsistently such that problems are not recognized until the traceability is needed and found wanting. In reality, it can be tricky to trace to artifacts created by other people and in other organizations, or to use others' trace links; much of the contextual knowledge needed to interpret and understand the traceability is often missing.

9.3 Dream Process (Portable)

- ***Traceability Strategy.*** Projects and organizations across the globe will use industry agreed upon standards, policies, representations and terminology for traceability, not because of mandate, but due to the obvious benefits and value of so doing.
- ***Traceability Creation and Maintenance.*** Where traceability is pre-established within a set of artifacts, it will be extracted, reused and integrated with the traceability of other artifacts with ease, irrespective of the tooling. Where traceability is integrated across a set of artifacts with their own traceability networks, this newly created traceability network will be maintained with ease.
- ***Traceability Use.*** Traceability will be retrieved such that it draws upon wider traceability networks to support any end-user traceability-related query or application need.

9.4 Goals (Portable)

Portable G 1	An industry agreed policy for traceability serves to define the minimal conditions under which any traceability solution and any resulting traceability network will integrate with any other.
Portable G 2	Comprehensive traceability information, comprising traceability information models, trace artifacts and trace links, are expressed in a common way, and retained and reused for full projects or for components therein.
Portable G 3	The traceability associated with individual projects and components is reconciled and merged seamlessly when reused across projects, product lines, organizations and domains. The traceability information models, processes and tools supporting the traceability are designed to enable this integration.
Portable G 4	Where traceability is reused or re-purposed for new contexts, multiple traceability networks are maintained as the trace elements change.
Portable G 5	Traceability is established dynamically, reaching out to incorporate previously unconnected artifacts within its scope as the search space for traceability widens.

9.5 Requirements (Portable)

9.5.1 Traceability Strategy (Portable)

Portable Req 1 — To standardize key aspects of the traceability process. [Portable G 1]

- *Status:* Traceability policies and standards are few at present, focus mostly on single project or organizational processes, and are rarely used in other than regulated industries and domains (e.g., military and aerospace standards).
- *Promise:* A loose framework of guiding policies, as common in some other industries requiring tracing (e.g., the food industry), supported by defined roles and responsibilities, may provide for a more flexible and less burdensome way to address the need for wider standardization in traceability processes.

Portable Req 2 — To agree upon and use a common representation to express the intended and actual traceability on a project. [Portable G 2]

- *Status:* While the traceable artifact types and trace link types may be listed in a requirements management plan, there is no agreed upon way to describe a traceability information model in research or practice, or even to describe a single trace that is created and maintained. While there have been numerous proposals as to the semantics of trace links, there is yet to be an agreement upon their classification and use. The traceability information models that show the full traceability intent for a project need to be examinable and the semantic meaning needs to be consistent to assess trace compatibility across projects. Likewise, the traces created and maintained need themselves to be examinable, consistent and extractable if they are to be shared and reused.
- *Promise:* A unified representation for expressing traceability information models, traces and other interchangeable traceability information would offer promise.

Portable Req 3 — To monitor and assure compliance to the policies, standards, representations and language used for traceability. [Portable G 1, G 2]

- *Status:* In regulated industries, the compliance of the traceability is generally assessed and assured via third parties. Automated techniques and tools are also beginning to assist in this space. This is less widely practiced in non-regulated industries. A related issue is the fact that the traceability terminology is not yet shared within the traceability community.
- *Promise:* Compliance will become easier to assess when the policies and representations for traceability have themselves have become better defined and their use is integrated into practice. A glossary of traceability terminology accompanies this technical report and may help to foster future agreement in the use of traceability terms by the community.

<u>*Portable Req 4*</u> To examine the integration potential of existing traces when they are to be merged and / or reused from across distributed project settings, and the subsequent potential for their maintainability. [Portable G 3, G 4]

- *Status:* There are no explicit mechanisms to assess the potential integration of traceability that has been pre-established for different artifact sets and is held in different tools, nor of the likely issues for subsequent traceability maintenance. Most of the research focus and practical implementation has been on inner product traceability, so there is no agreed upon standard for extracting and sharing the traceability across products over time. This is often the case even within organizations. Recent attention has been paid to reusing traceability between variants in a product line, and support for this is maturing in practice in some industries (e.g., in the automotive industry). This trace reuse is carefully built into the engineering practice, through an examination of variability, and is not determined post-hoc.
- *Promise:* The traceability work that is emerging from product line engineering contexts may have wider applicability to broader traceability reuse.

<u>*Portable Req 5*</u> To develop reconciliation tactics to accommodate specific project and organizational needs when merging and reusing previously disparate or legacy traceability networks. [Portable G 3]

- *Status:* Legacy projects can have their traceability recovered via automated techniques with some success, though research has not yet looked into wider traceability integration and reconciliation of traces across multiple traceability networks. Reconciling traces that have been created by other people in other projects and organizations is a relatively open area, but one that will become increasingly relevant with the service-oriented provisioning of software and systems. The attention to incorporating institutional knowledge about traceability may facilitate such sharing and reuse.
- *Promise:* The provision of appropriate contextual information, alongside the traces that are used or reused, may ease the understanding and merging of myriad trace elements by humans. Useful trace metadata to support automatic reconciliation also needs to be explored.

9.5.2 Traceability Creation and Maintenance (Portable)

<u>*Portable Req 6*</u> To provide supporting mechanisms to facilitate tracing to artifacts created by other people in other projects and organizations, perhaps held in diverse toolsets. [Portable G 1, G 2, G 3]

- *Status:* Dedicated requirements management tools offer varying levels of support for incorporating artifacts created outside of the tool within a traceability network, often via pre-processing, though support for bi-directional traceability to these other tools can be variable and impede future maintenance of the traceability once incorporated. There are prototype tools that have demonstrated the creation of trace links across heterogeneous CASE tools at distributed locations though. A common method for supporting trace portability between artifacts in disparate tools is indirectly via XML (Extensible Markup Language). Such trace links may also be held in one place and as a separate artifact to ease extraction and reuse.

- *Promise:* Decoupling the representation of the trace links from the trace artifacts, irrespective of where the trace elements are physically stored, will further aid trace extraction, portability and reuse.

<u>Portable Req 7</u> To monitor and identify changes in trace-related artifacts, irrespective of their storage location, and to propagate the necessary traceability updates to those traceability networks in which they participate. [Portable G 4]

- *Status:* It can be problematic to maintain traceability in dedicated requirements management tools if other third party tools have been used for different stages of the software and systems development lifecycle. This requires clear protocols for the interchange and interoperability of data between the tools. Application lifecycle management tools ameliorate the problem as they are fully integrated tools and, as such, can propagate traceability changes internally. The repurposing of artifacts in multiple traceability networks may happen routinely within a single project and tool, but extending this to their inclusion within additional project and tooling contexts is not routine at present.
- *Promise:* Maintaining reused traces will be less problematic where the reused trace artifacts and trace links are initially created and maintained, and then subsequently reused, within fully integrated and interoperable toolsets. Where this is not the case, differentiating live reuse (i.e., where updates to artifacts impact the traceability) from copied reuse (i.e., where updates to artifacts does not impact the traceability) may be important to investigate here.

9.5.3 Traceability Use (Portable)

<u>Portable Req 8</u> To understand and use the trace artifacts and trace links established by third parties in traceability-related queries. [Portable G 5]

- *Status:* Traceability-related queries are generally targeted to an associated set of artifacts for which the traceability has been explicitly defined and created. Research has not looked at how this could be extended to incorporate additional artifacts, opportunistically, within its remit, and whether this would even add value as a concept. This may become more important as systems are developed from pre-existing components and services.
- *Promise:* Drawing upon a number of traceability networks or non-traced artifacts in providing support for end-user tasks demands standards in the base representation of traces and trace elements. A Google-strength search capability may be incorporated into future traceability solutions to find new traces.

9.6 Recommended Research (Portable)

The major research theme to achieve portable traceability is ***to agree upon universal policies, standards, and a unified representation or language for expressing traceability concepts.*** Supporting research topics are listed below.

Research ID	**Description**	**Req ID**
Portable RT 1	Develop a unified representation or language for expressing traceability information models and for representing traces.	Portable Req 2

Portable RT 2	Define and agree upon the semantic meaning of the various types of trace artifacts and trace links used in different domains.	Portable Req 2
Portable RT 3	Define policies, standards, infrastructure, processes and tools for tracing distributed artifacts in distributed settings, enabling cross-boundary traceability of all forms.	Portable Req 1, 6, 7
Portable RT 4	Examine the likely forms of cross-boundary traceability required in the future.	Portable Req 6, 7
Portable RT 5	Provide a way to examine pre-established traceability and to assess its integration or reuse potential with or within other contexts of use.	Portable Req 4
Portable RT 6	Develop mechanisms to help extract, integrate and reuse traceability work products.	Portable Req 3, 4, 5, 6, 7, 8
Portable RT 7	Learn about traceability representations, policies and standards in other distributed industries (such as the food industry), and the regulatory standards that mandate it, to apply lessons to software and systems contexts.	Portable Req 1, 2
Portable RT 8	Re-conceptualize traceability as a service so that it can be procured and interchanged at will.	Portable Req 8

9.7 Positive Adoption Practices for Industry (Portable)

Portable IP 1 — Practitioners actively engage in defining and using policies and standards that enable cross-boundary traceability of multiple forms.

Portable IP 2 — Practitioners use a unified representation or language to describe both the intended traceability and the actual traceability on their projects.

Portable IP 3 — Practitioners reuse and integrate the traceability from other projects, and from components of other projects and services, with ease.

Portable IP 4 — Professional bodies agree upon ways to encourage and enforce the use of industry agreed upon standards, policies, representations and terminology for traceability.

[continued…]

10 Traceability Challenge 7: Traceability that is Valued

Traceability is a strategic priority and valued by all; every stakeholder has a role to play and actively discharges his or her responsibilities.

10.1 Link to Vision (Valued)

In the vision scenario, all the stakeholders simply expect the traceability to be there in the engineer's project just like computation, electricity and oxygen. Traceability is a commodity that is built into organizations and projects since they have realized that they cannot be agile and competitive without it. Its value is undisputed and has long been institutionalized within the engineer's organization and the wider industry, supported by top management and workers alike. Every action that the stakeholders take on the project preserves and adds to this valuable traceability asset. The engineer could not do her job without traceability and the flying solar car business of her organization would not be viable in the longer-term without the value-added support provided by traceability.

10.2 Problems Addressed (Valued)

Traceability is often valued to the extent that organizations may invest in a tool; there is still somewhat of a misconception that tools will do the traceability job once configured. While current tools provide varying levels of support for traceability, they require organizations to define (at a minimum) a traceability process to be used effectively, and many organizations do not invest in this aspect of the tool procurement process sufficiently. Inadequate training in the ensuing traceability process compounds the issue. The required skills for doing a good job at traceability are unclear and so people may be allocated the job without sufficient preparation and training. Such people do not always see the personal reward from doing this job meticulously and there can be little motivation to do the task well if the benefits are perceived to be too few or too distant. This can lead to a lack of total stakeholder buy-in to establishing traceability. Traceability certifications do not exist, so are consequently not expected of people or of organizations, so what you get by way of traceability in practice can be a complete surprise. The granularity at which to trace also remains a value question concerning effort and payback, and getting this wrong can devalue any traceability that is established. To management, the competitive advantage of traceability may therefore end up not being evident. A typical concern that is a barrier to technology transfer is whether any investment in a traceability initiative, including training, is actually worth it; value is questioned, along with its value to whom.

10.3 Dream Process (Valued)

- ***Traceability Strategy.*** The inherent and added value of traceability will be discussed on day one of a project and everyone, henceforth, will work together to do a good job on it.
- ***Traceability Creation and Maintenance.*** Practitioners will love the intellectual challenge of creating and maintaining trace links. They will take real pride in their job because they know that what they do is valued and respected by their peers who will use the resulting traces in the future.

- ***Traceability Use.*** Everyone will want traceability and expect it, as it is one of the most valued support dimensions of a project, making testing more exacting and helping functioning code to meet its stakeholder requirements.

10.4 Goals (Valued)

Valued G 1	Everyone, from upper management to workers, understands and buys into the value of traceability on a project.
Valued G 2	A return on investment profile for traceability is available to consult and a traceability value proposition is used in strategic project planning.
Valued G 3	Resources are provisioned to match the traceability need for a project, meaning that people are trained in traceability logistics and tools are grounded in traceability processes.
Valued G 4	Traceability use is exploited to add value to many project planning and management tasks.

10.5 Requirements (Valued)

10.5.1 Traceability Strategy (Valued)

<u>*Valued Req 1*</u> To develop a value proposition for traceability on a project to help determine and sustain a suitable traceability strategy. [Valued G 2]

- *Status:* Some practitioners may be hard-pushed to articulate the actual value of traceability to them, or even to their projects and organizations, while some project managers still have not even heard of traceability. Traceability is obviously more valued in certain domains than in others at present, such as for safety-critical software systems, which impacts the degree of traceability planning and management undertaken in the various domains. While this should and will continue to be the case, as the value of traceability will differ widely between organizations dependent upon the business environment and domain of the company, there is scope to examine traceability value propositions in varying contexts. A recent follow-up survey of an industry pilot study showed that engineers found trace retrieval methods useful, so awareness of traceability value is emerging. However, there is currently no language as such for describing and discussing traceability value. When costs are cut on a project, traceability can be one of the first things to go, and this is somewhat indicative of how its actual value is construed, measured and managed at present.
- *Promise:* Value-based traceability is a growing area of interest in the traceability community and is leading the way in researching the concept of and measures for traceability value. Such research needs to find its way into strategic planning and management tools for traceability. Progress here will also depend upon advances with traceability challenge two.

<u>*Valued Req 2*</u> To provide people with the necessary knowledge and skills that they need to undertake their traceability tasks successfully. Further, to provide the requisite money, time and technology resources for these people to fulfill these tasks. [Valued G 3]

- *Status:* People are often assigned to a traceability task in practice with a vague job description and little prior training. While there are also many experienced traceability practitioners, there is a lack of an established industry-wide apprenticing or a mentoring model to acquire or impart the necessary traceability skills to new personnel. Few educational or training programs exist to impart proficiency in how to plan for and manage traceability, how to create and maintain traces, or even how to educate as to its inherent and wider value.

- *Promise:* There are conferences and workshops that emphasize traceability topics. The reporting on industry case studies can demonstrate value, put new practitioners in contact with seasoned ones, and hence communicate both the value of and skills underlying traceability. The systematic gathering of good practices and benchmark examples will help to foster knowledge sharing further.

<u>*Valued Req 3*</u> To define traceability roles and responsibilities on a project, both within and across organizations. [Valued G 3]

- *Status:* It is often unclear as to who is in charge of traceability in an organization (e.g., is it the requirements engineers, software architects, developers, maintenance team, etc.?) In practice, the responsibilities for traceability tend to lie with one or a few people on a project; traceability is rarely a fully distributed responsibility.
- *Promise:* Visibility is important to accountability and more could be done to make the roles and responsibilities for traceability visible, such as via the creation of traceability development contracts. To accommodate complex settings, traceability tasks really need to be made part of the job description of all software and systems development roles such that some form of traceability is made an integral part of everyday work activities.

<u>*Valued Req 4*</u> To make traceability assurance a fundamental part of project management and quality assurance practice, performing regular trace audits to monitor and measure value creation. [Valued G 3, G 4]

- *Status:* Traceability value is implied where it is required by standard process improvement initiatives, such as the Capability Maturity Model Integration (CMMI), or by various regulatory bodies. In such cases, traceability is fundamental to the process, valued as such and assessed. Where organizations have been appraised at certain maturity levels, some assurance of their traceability practices may be assumed.
- *Promise:* If traceability value propositions are defined and integrated into the software and systems development process more routinely, then they can be tracked and measured in the future.

10.5.2 Traceability Creation and Maintenance (Valued)

<u>*Valued Req 5*</u> To define value propositions for traces as they are created and to update these value propositions for traces as they are maintained. [Valued G 2]

- *Status:* Researchers have not built a convincing case regarding the value of traceability creation and maintenance, especially to those engaged in the traceability process, let alone the value of the respective strategies for so doing, such as creating traces early or on-demand, and maintaining traces continuously or on demand. There is also little understanding of those decisions that impact value when creating and maintaining traces, such as the specific technique to use and the granularity.
- *Promise:* Measures, baselines and benchmark experiments for examining the value of traces over time are needed here.

<u>*Valued Req 6*</u> To reward practitioners for doing a good job at traceability creation and maintenance. [Valued G 1]

- *Status:* What constitutes and defines a 'good' job at traceability creation and maintenance is not really a matter of consensus. Few educational or training programs exist to impart proficiency in how to create and maintain traceability as part of regular development training activities, nor convey the standards to which to aspire. Without such baselines it is difficult to set expectations and for practitioners to be held accountable for their work.
- *Promise:* Well-defined job descriptions for traceability creators and maintainers, ones that account for the specifics of organizations, domains and development approaches, and ones that set guidelines for practice, will help to advance the parameters for traceability quality measures. Industry awards for excellence in traceability creation and maintenance may be an option to build up the reputation of practice.

10.5.3 Traceability Use (Valued)

Valued Req 7 — To add value to wider project tasks through the use of traceability, to inform business decisions and to measure the resulting value. [Valued G 2, G 4]

- *Status:* Value-based traceability research is examining how traceability can support the global value estimation of a software product, release management, feature prioritization, etc. Nevertheless, few educational and training programs currently exist to impart proficiency in how to use the results of traceability for development or business strategic advantage. Practitioners and customers in regulated domains are primarily the ones demanding traceability at present, though it is not always clear whether this is due to mandate or due to the perception of value.
- *Promise:* Research in how traceability can be put to wider end-use in software and systems engineering, followed by education and training, will promote the value perception of traceability. The promise lies in software tools that use traceability more than today to provide sophisticated support to business stakeholders, as well as to the engineers. More case studies reporting on the risks and impact of not having readily accessible traceability on a project, in addition to positive value case studies, are needed.

10.6 Recommended Research (Valued)

The major research theme to achieve valued traceability is ***to raise awareness of the value of traceability, to gain buy-in to education and training, and to get commitment to implementation.*** Supporting research topics are listed below.

Research ID	**Description**	**Req ID**
Valued RT 1	Develop techniques, methods and tools to support and measure various traceability value propositions on a project.	Valued Req 1, 4, 5, 7
Valued RT 2	Define traceability roles and responsibilities within a traceability development contract, and provide support for instantiating and discharging these in different project and organizational settings.	Valued Req 2, 3, 6
Valued RT 3	Identify the core knowledge areas and associated skills for doing (and using) traceability, and create effective pedagogical materials (e.g., model examples) to integrate competency for traceability into software and systems engineering teaching and training.	Valued Req 2, 5, 7

Valued RT 4	Increase awareness of traceability value by developing software tools that use traces in more interesting and value-added ways than today for wider software and systems engineering and business tasks.	Valued Req 1, 4, 7
Valued RT 5	Gather experimental evidence within the Traceability Body of Knowledge (TBOK) on the role of traceability with respect to software and systems development success rates and longevity.	Valued Req 1, 7

10.7 Positive Adoption Practices for Industry (Valued)

Valued IP 1 — Managers are aware of the value of traceability on their project and in their organization, so they ensure that their employees are trained in the discipline and that they are compensated for doing a good job.

Valued IP 2 — Practitioners actively seek training, and potentially certification, in traceability excellence.

Valued IP 3 — Practitioners both want and demand traceability of their software and systems engineering work products and processes; customers of software, systems and services expect 'traceability inside'.

Valued IP 4 — Universities and colleges integrate traceability into their software and systems engineering curricula, at all degree levels, and students choose these curricula for their future job prospects.

Valued IP 5 — The Traceability Body of Knowledge (TBOK) is consulted to determine and use value propositions to guide traceability strategizing and practice.

[continued…]

11 Traceability Challenge 8: Traceability that is Ubiquitous

THE GRAND CHALLENGE OF TRACEABILITY *– Traceability is always there, without ever having to think about getting it there, as it is built into the engineering process; traceability has effectively "disappeared without a trace."*

11.1 Link to Vision (Ubiquitous)

There is no mention of traceability anywhere in the vision scenario as it is truly behind the scenes. The engineer does not establish traceability explicitly; traceability is established automatically via her actions and via the actions of others. Traceability of the requirements trade-offs and negotiations are automatically captured from the tooling environments that the engineer uses, along with the rationale. Traceability data is presented to the engineer in a ready-to-use and usable manner as a by-product of her engineering process and of using her tools, and is never explicitly sought. Traceability neither disrupts the engineer from her primary tasks nor does she spend a micro second thinking about it. Software components, systems and services are customized by the other engineers while not having to worry about the detailed specifics of the underlying technologies and traceability information.

11.2 Problems Addressed (Ubiquitous)

Traceability is perceived as, or actually is, a burden for practitioners as it is mostly manual and repetitive in nature. Establishing or using traceability often interrupts tasks that are considered more important when it comes to software and systems development. It also often requires engineers to use special-purpose tools and so disrupts their primary working practices. Establishing traceability manually is further open to human error and inconsistency, and its quality is only as good as the efforts of its weakest human link. Traceability should not be the goal of software and systems development, and it certainly should not force a break in an engineer's workflow, but it often ends up being construed that way. If traceability gets in the way, people simply stop doing it with the care and with the rigor that it demands to be successful; and, if traceability is not there when it is needed and expected, people stop using it. It can be a vicious cycle.

11.3 Dream Process (Ubiquitous)

- ***Traceability Strategy.*** An integrated development environment will be set up and configured to establish the traceability demanded of a project, in the confidence that the approach will be adapted as needed, allowing the people involved to focus on the more creative development work.
- ***Traceability Creation and Maintenance.*** Trace creation will be completely automated, to specified quality levels, with 100% recall and precision. Trace maintenance will either be completely automated or superseded by automated on-demand trace creation, dependent upon the cost proposition of either strategy.
- ***Traceability Use.*** Stakeholders will both use and come to depend upon traceability on an everyday basis, without even really knowing it.

11.4 Goals (Ubiquitous)

Ubiquitous G 1 Near zero (or acceptable) stakeholder effort is required to establish and make use of traceability, with no (or minimum) impact on their primary task.

Ubiquitous G 2 Traceability is de facto in software and systems development processes and their supporting integrated development environments.

Ubiquitous G 3 A virtuous cycle is sustained as traceability is established and used both painlessly and effectively.

11.5 Requirements (Ubiquitous)

While there are many dependencies between all the requirements of the seven previously discussed traceability challenges[17], traceability challenge eight is unique in that it really depends upon having made significant progress with satisfying the requirements of these previous seven challenges. In addition to the status and areas of promise for each requirement, the core dependencies with the previous challenges are also suggested in this section.

11.5.1 Traceability Strategy (Ubiquitous)

Ubiquitous Req 1 To automate routine traceability planning and management tasks. [Ubiquitous G 1]

- *Status:* Traceability has to be planned for on a project, and the implementation of this plan requires on-going human monitoring and control. This can comprise setting up a traceability solution (i.e., a trace-ability information model, process and tooling) and ensuring both its use and fitness for use over time. The underlying components of this task are under examination as part of a generic traceability process model to inform as to those areas amenable to automation and to offer more practice guidelines.
- *Promise:* Progress with traceability challenge one (purposed), in the form of profiles and patterns for traceability, will assist with defining and setting up traceability on a project. Progress with traceability challenges three (configurable) and six (portable) will lead to the parameterization, reuse and adaptation of traceability strategies, while progress with traceability challenge two (cost-effective) will reduce the cost of traceability start-up.
- *Dependencies:* Traceability that is purposed, cost-effective, configurable and portable.

Ubiquitous Req 2 To integrate traceability planning and management processes into the overarching software and systems development planning and management process. [Ubiquitous G 1, G 2]

- *Status:* Traceability is not always an integral part of general project planning and management, so it is often tackled in isolation as and when needed on projects, rather than built into the software and systems development lifecycle. Two exceptions are model-driven development and formal development processes where the transformations are essential to the underlying development philosophy and provide for traceability.

[17] Expressing these requirements dependencies and determining priorities remain topics for future work (see Section 12).

- *Promise:* Progress with traceability challenges one (purposed) and seven (valued) will assist with getting traceability integrated into wider development processes, as tighter support for primary development tasks is demanded and provided.
- *Dependencies:* Traceability that is purposed and valued.

<u>Ubiquitous Req 3</u> To determine where manual intervention is unavoidable in the traceability process, to keep the required human involvement to a minimum, and to provide for better process guidance and tool support when unavoidable. [Ubiquitous G 1, G 2, G 3]

- *Status:* While the ultimate goal may be for total automation of traceability, it is likely that there will always be some cases in which human intervention is required to assess the validity and value of traces, resulting in a more symbiotic system. Work on understanding the component activities of a generic traceability process model, and the potential human interaction points, is underway.
- *Promise:* Three key drivers for the complete automation of traceability are to reduce the cost of traceability, to increase the trust in the results and to allow for scale. Progress with traceability challenges two (cost-effective), four (trusted) and five (scalable) will help to shape the boundaries for what is viable in the way of traceability automation.
- *Dependencies:* Traceability that is cost-effective, trusted and scalable.

11.5.2 Traceability Creation and Maintenance (Ubiquitous)

<u>Ubiquitous Req 4</u> To create and maintain traces automatically, as a by-product of working in integrated development environments. Where manual intervention is unavoidable, to make traceability creation and maintenance a single 'click' process. [Ubiquitous G 1, G 2, G 3]

- *Status:* Traceability creation and maintenance is still mostly manual in practice, and it can become a full time job for some people in some projects. However, the automated recovery and capture of trace links is producing reasonable results in research settings and gaining some acceptance in industrial practice. There is also successful semi-automated maintenance of trace links in certain development contexts, such as in UML-based development, and research on the full automation of trace maintenance is gaining momentum. Leading requirements management and application lifecycle management tools provide for some flexibility in defining the traceability that can be enabled through their use and for some automated capturing of the traces (e.g., support for real-time trace capture as a by-product of working in the JAZZ environment). UML-based tools that support model-driven development are also leading the way in this area.
- *Promise:* Automated techniques, methods and tools for traceability creation and maintenance will continue to improve. More variety in the base techniques (e.g., information retrieval based, rule based, event based, etc.), along with options to vote on the results from competing techniques, will lead to improved quality levels in the traces they obtain. Moreover, the ability to automatically recover traces faster than identifying the delta of what has changed would potentially eliminate the need for traceability maintenance altogether (i.e., traces would simply be created on-demand and never maintained). What is lost from having no human involvement and no record of the trace evolution would need to be studied carefully, and the cost / benefit trade-off of trace creation versus trace maintenance also studied. However, the promise lies not just in performance improvements, but in closing the loop to ensure that the traces that are created and maintained are fit for purpose, account for the entire necessary artifact types and are trusted. This relies upon progress with traceability challenges one (purposed), four (trusted) and five (scalable).
- *Dependencies:* Traceability that is purposed, trusted and scalable.

11.5.3 Traceability Use (Ubiquitous)

Ubiquitous Req 5 — To support end-user tasks, without any distraction from the underlying traceability that is being retrieved and rendered visible to make this support possible. [Ubiquitous G 1, G 2, G 3]

- *Status:* Traceability is used in a number of wider software and systems engineering activities, such as testing, version control, configuration management and quality assurance. There are some traceability-enhanced tools for these areas that do not make the traceability evident and unwieldy. In general, end-users are presented with unintuitive traceability matrices and hierarchical reports at present, to interpret and make use of the traceability to support many other tasks. Their use can be cumbersome and get in the way of the task at hand, so end-users are often made very aware of the traceability that they are have to call upon.
- *Promise:* Improved support for end-user tasks relies upon progress with traceability challenge one (purposed) and on novel approaches to address issues of scale and complexity in traceability end-use, particularly through improved visualizations and task matching, so progress with traceability challenge five (scalability) too. Re-conceptualizing traceability as a service for wider software and systems development tasks, integral to all the supporting processes and tools, could also provide for advances here. This relies upon progress with traceability challenges three (configurable) and six (portable).
- *Dependencies:* Traceability that is purposed, configurable, scalable and portable.

11.6 Recommended Research (Ubiquitous)

The major research theme to achieve ubiquitous traceability is ***to provide automation such that traceability is encompassed within broader software and systems engineering processes, and is integral to all tool support.*** Supporting research topics are listed below.

Research ID	Description	Req ID
Ubiquitous RT 1	Investigate novel ways to define the traceability strategy, such as in an executable way, so that the traceability solution simply follows from the specification of the traceability need, as per model-driven or formal development.	Ubiquitous Req 1, 2
Ubiquitous RT 2	Total automation of (or 'one-click') traceability creation and trace maintenance, with quality and performance levels superior to manual efforts.	Ubiquitous Req 3, 4
Ubiquitous RT 3	Embed traceability into all the software and systems engineering techniques and methods for all of the tasks that it facilitates, and provide this traceability support seamlessly from within a total automated tooling solution that is underpinned by a sound traceability process.	Ubiquitous Req 3, 4, 5

11.7 Positive Adoption Practices for Industry (Ubiquitous)

Ubiquitous IP 1	Practitioners choose integrated development environments based upon the traceability-enabled software and systems engineering activities that they provide and enable. They have 'traceability inside'.
Ubiquitous IP 2	Practitioners configure the traceability parameters that they need on a project in an integrated development environment and then forget about it, as it is henceforth established and evolved as needed and behind the scenes.
Ubiquitous IP 3	Practitioners know that they are establishing and making use of traceability in their everyday tasks, but they do not have to do anything extra to achieve this. They further benefit from this traceability when developing and customizing their own applications based upon the composition of building blocks and services.
Ubiquitous IP 4	Practitioners do not talk about the 'traceability problem' because it has been solved.

[continued…]

12 Validation, Evolution and Intended Use

This technical report presents a snapshot of a community work in progress, now over five years into the process. The new and updated Grand Challenge of Traceability v1.0 has been cross-referenced to the draft Problem Statement and Grand Challenges (v0.1) document (Cleland-Huang et al. 2006) to maintain continuity. Figure 6 shows the traceability matrix between the two versions, as created by two of the contributing authors. The intended use of the reformulated material, along with the process for gathering feedback from the wider traceability community, is outlined in this section.

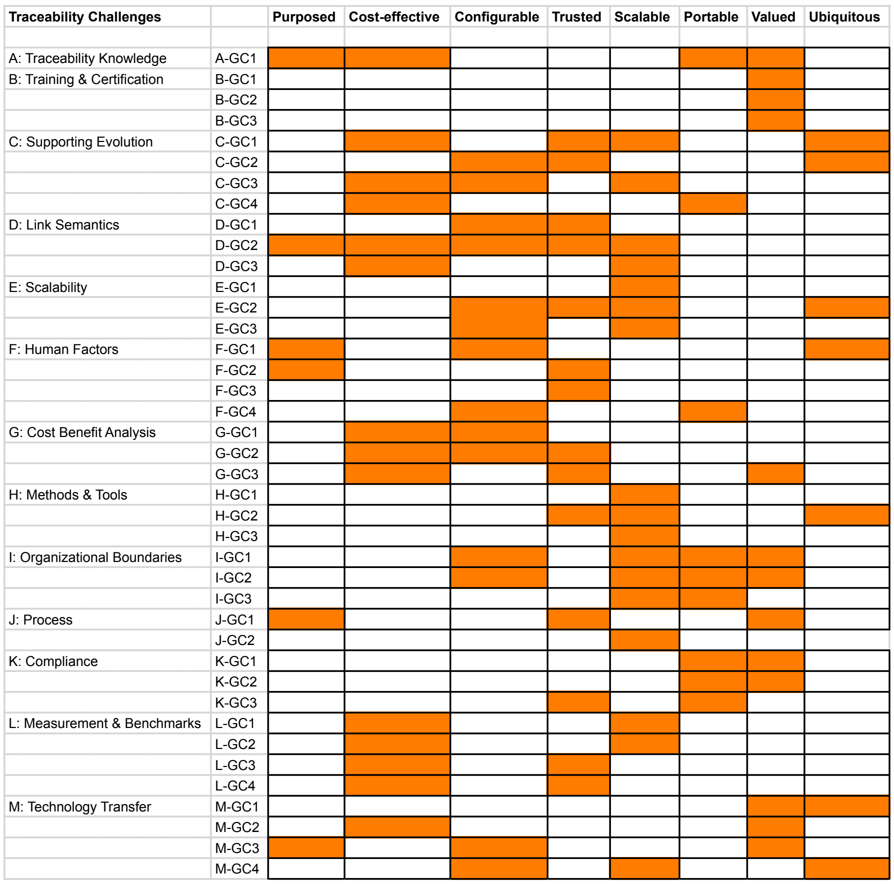

Fig. 6 Traceability matrix mapping the challenges of the draft Problem Statement and Grand Challenges (v0.1) document (Cleland-Huang et al. 2006) to those of The Grand Challenge of Traceability (v1.0)

12.1 Dissemination and Feedback Process

The core material from this technical report is made publicly available on the website of the Center of Excellence for Software Traceability (Hayes et al. 2007): *http://www.coest.org*. The CoEST website lists all eight traceability challenges and their major research themes. For each challenge, it summarizes the underlying goals, requirements, areas of promise, research topics and positive adoption practices for industry. The website has been set up as a community resource to disseminate traceability good practices, and to gain wider feedback to validate and evolve the work on the traceability challenges.

Feedback is currently being solicited on the individual research topics to gain community input on the likely impact of the research topic, the anticipated research difficulty and the effort required to accomplish the research. Given the internal traceability of the individual research topics to the requirements, goals and challenges within this document, the broader intention is to accumulate these data to ascertain the status of and progress with respect to the individual traceability challenges over time, and so, in turn, with the overarching grand challenge.

Feedback is also being sought from practitioners on the state of the industry practice. This is to assess whether the positive adoption practices are evident in any domains, organizations and projects, and to be in a position to track this status over time. References are also being sought to existing publications and ongoing research projects that address the various research topics. The intention here is to gain data to summarize the state of the art in a more exacting manner, to understand where traceability research efforts are and are not directed at present, and to assess the status of the overarching research theme for each traceability challenge over time.

Such data gathering is going to require a substantial and sustained effort by the traceability community to be both useful and successful. One proposal to ease this effort is to use the research topics and industry practices as a means to classify traceability-related submissions and publications at future conferences and workshops. This would help to track traceability research contributions and industrial reality going forward. Equally, each new research contribution in the field could be more explicit in documenting the traceability challenges that it tackles.

An environment for traceability experimentation and benchmarking is currently in development under the auspices of the Tracy project (Cleland-Huang et al. 2011). This environment, called TraceLab, intends to provide the traceability community with experiments and datasets to begin to baseline and benchmark traceability techniques, methods and tools. The proposal is to launch traceability contests within TraceLab that serve to contribute progress toward the various research topics. This will provide an additional way to collect data on traceability research efforts with respect to the challenges going forward.

12.2 Toward a Roadmap for Traceability Research

The material within this technical report forms the basis for a traceability research roadmap that is currently under preparation by the authors. The realization of the grand challenge of ubiquitous traceability is dependent upon progress with each of the seven other challenges. These traceability challenges are, themselves, crosscutting concerns, so progress on certain research topics will therefore contribute to a number of the other challenges in various ways. The intent of the research roadmap is to highlight these research dependencies and, in conjunction with early feedback from the CoEST website, to delineate priorities for traceability research over the near-term, mid-term and longer-term.

13 Conclusions

The Grand Challenge of Traceability (v1.0) is a major update to a draft document developed by members of the traceability community in 2006 (Cleland-Huang et al. 2006). It reformulates the forty prior grand challenges as seven major traceability challenges and one overarching grand challenge for traceability. Associated with these challenges are seven major themes for traceability research, along with one more dominating and long-term theme.

The Grand Challenge of Traceability is to make traceability ubiquitous:

The Grand Challenge of Traceability – Traceability that is Ubiquitous.
Traceability is always there, without ever having to think about getting it there, as it is built into the engineering process; traceability has effectively "disappeared without a trace."

Associated with achieving this grand challenge is the following major long-term research theme:

Long-term Research Theme – To provide automation such that traceability is encompassed within broader software and systems engineering processes, and is integral to all tool support.

To achieve such traceability ubiquity in software and systems engineering practice, seven underlying traceability challenges need to be tackled. Each of these challenges has a major research theme associated with it:

1 ***Purposed.*** Traceability is fit-for-purpose and supports stakeholder needs (i.e., traceability is requirements-driven).

   Major Research Theme – To define and instrument prototypical traceability profiles and patterns.

2 ***Cost-effective.*** The return from using traceability is adequate in relation to the outlay of establishing it.

   Major Research Theme – To develop cost-benefit models for analyzing stakeholder requirements for traceability and associated solution options at a fine-grained level of detail.

3 ***Configurable.*** Traceability is established as specified, moment-to-moment, and accommodates changing stakeholder needs.

   Major Research Theme – To use dynamic, heterogeneous and semantically rich traceability information models (or similar specifications of the intended traceability) to guide the definition and provision of traceability.

4 ***Trusted.*** All stakeholders have full confidence in the traceability, as it is created and maintained in the face of inconsistency, omissions and change; all stakeholders can and do depend upon the traceability provided.

   Major Research Theme – To perform systematic quality assessment and assurance of the traceability.

5 ***Scalable.*** Varying types of artifact can be traced, at variable levels of granularity and in quantity, as the traceability extends through-life and across organizational and business boundaries.

   Major Research Theme – To provide for levels of abstraction and granularity in traceability techniques, methods and tools, facilitated by improved trace visualizations, to handle very large datasets and the longevity of these data.

6 ***Portable.*** Traceability is exchanged, merged and reused across projects, organizations, domains, product lines and supporting tools.

   Major Research Theme – To agree upon universal policies, standards, and a unified representation or language for expressing traceability concepts.

7 ***Valued.*** Traceability is a strategic priority valued by all; every stakeholder has a role to play and actively discharges his or her responsibilities.

   Major Research Theme – To raise awareness of the value of traceability, to gain buy-in to education and training, and to get commitment to implementation.

The eight traceability challenges were determined by exploring the assumptions of a community vision for traceability in 2035. The major research themes associated with each challenge were determined by expressing the goals and requirements that would be needed of a generic traceability process to address the challenge, by examining the state of the art and the state of the practice, and by considering areas of promise and necessary topics for research. In conducting this systematic analysis, one challenge and its associated research theme appeared to depend upon progress with all of the others, and so it was labeled as the grand challenge of traceability. The intention of this new document is to provide a structured framework for directing, classifying and tracking past and future research efforts in the field of traceability.

Acknowledgements

The authors would like to thank all the participants of the two initial traceability workshops in which the initial grand challenges for traceability were explored (at NASA's IV&V facility in the Summer of 2006, and in Lexington, Kentucky in the Spring of 2007). They particularly thank Brian Berenbach for industry-related feedback and Patrick Mäder for glossary suggestions. The authors would also like to thank NASA and the NSF for funding these original workshops (NASA grant number NNX06AD02G and NSF grant number 0647443).

References

(Arkley and Riddle 2005)
Arkley, P. and Riddle, S. Overcoming the Traceability Benefit Problem. In Proceedings of the 13th IEEE International Conference on Requirements Engineering, Paris, France, 29 August - 2 September 2005, pp. 385 – 389.

(Bianchi et al. 2000)
Bianchi, A., Visaggio, G. and Fasolino, A.R. An Exploratory Case Study of the Maintenance Effectiveness of Traceability Models. In Proceedings of the 8th International Workshop on Program Comprehension, Limerick, Ireland, 10 - 11 June 2000, pp. 149 - 158.

(Cleland-Huang et al. 2006)
Cleland-Huang, J., Hayes J.H. and Dekhtyar, A. (Eds.) Center of Excellence for Traceability: Problem Statements and Grand Challenges (v0.1). Center of Excellence for Traceability Technical Report COET-GCT-06-01-0.9, 10 September 2006.

(Cleland-Huang et al. 2011)
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. and 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.

(CMMI Product Team 2010)
CMMI Product Team. CMMI for Development, Version 1.3. Technical Report CMU/SEI-2010-TR-033 (ESC-TR-2010-033), Carnegie Mellon University Software Engineering Institute, November 2010.

(Dahlstedt and Persson 2005)
Dahlstedt, A.G. and Persson, A. Requirements Interdependencies: State of the Art and Future Challenges. In Engineering and Managing Software Requirements, Aurum A. and Wohlin, C. (Eds.), Springer Berlin Heidelberg, 2005.

(Egyed et al. 2007)
Egyed, A., Grunbacher, P., Heindl, M. and Biffl, S. Value-based Requirements Traceability: Lessons Learned. In Proceedings of the 15th IEEE International Requirements Engineering Conference, New Delhi, India, 15 - 19 October 2007, pp. 115 - 118.

(FAA 1992)
Radio Technical Commission for Aeronautics, Inc. (RTCA). DO-178B: Software Considerations in Airborne Systems and Equipment Certification, Issued 12-1-92, Prepared by SC-167, Supersedes DO-178A, Errata Issued 3-26-99.

(FDA 2002)
U.S. Food and Drug Administration, General Principles of Software Validation; Final Guidance for Industry and FDA Staff, January 11, 2002, http://www.fda.gov/MedicalDevices/deviceregulationandguidance/guidancedocuments/ucm085281.htm.

(Gotel and Finkelstein 1994)
Gotel, O. and Finkelstein, A. An Analysis of the Requirements Traceability Problem. In Proceedings of the 1st IEEE International Conference on Requirements Engineering, Colorado Springs, Colorado, USA, 18 – 22 April 1994, pp. 94 – 101.

(Hayes et al. 2007)
Hayes J.H., Dekhtyar, A. and Cleland-Huang, J. Charter (Business Plan) for the Center of Excellence for Traceability. COET-CBP-07-02-1.0, 15 February 2007. (http://www.traceabilitycenter.org/).

(Lindvall and Sandahl 1996)
Lindvall, M. and K. Sandahl, K. Practical implications of traceability. Software — Practice and Experience, Volume 26, Number 10, October 1996, pp. 1161 - 1180.

(Mäder et al. 2009)
Mäder, P., Gotel, O. and 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.

(Pierce 1978)
Pierce, R. A Requirements Tracing Tool. ACM SIGSOFT Software Engineering Notes. Volume 3, Number 5, November 1978, pp. 53 - 60.

(Ramesh and Jarke 2001)
Ramesh B. and Jarke M. Towards Reference Models for Requirements Traceability. IEEE Transactions on Software Engineering, Volume 27, Number 1, January 2001, pp. 58 - 93.

(Spanoudakis and Zisman 2005)
Spanoudakis G. and Zisman A. Software Traceability: A Roadmap. In Handbook of Softawre Engineering and Knowledge Engineering, Volume 3: Recent Advances, Chang, S.K. (Ed.), World Scientific Publishing Co., ISBN:981-256-273-7, August 2005.

(von Knethen and Paech 2002)
von Knethen, A. and Paech, B. A survey on tracing approaches in practice and research, Fraunhofer IESE Research Report 095.01/E, Kaiserslautern, Germany, 2002, http://publica.fraunhofer.de/documents/N-9197.html (accessed January 2010).

(Winkler and von Pilgrim 2010)
Winkler, S. and von Pilgrim, J. A survey of traceability in requirements engineering and model-driven development. Software and Systems Modeling, Volume 9, Number 4, September 2010, pp. 529 - 565, Springer (Published on line 22 December 2009).

Glossary[18]

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*.

---

[18] An up to date version of this glossary is maintained on the website of the Center of Excellence for Software Traceability (CoEST): *http://www.coest.org*. Please direct any glossary additions or updates to this website. To promote consistency in the use of terms within the traceability community, preferred terms are denoted by * and U.S. English spellings are used throughout.

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 lifecycle – 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 lifecycle – 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 lifecycle-wide or end-to-end *traceability*, such as from requirements to code. *Vertical tracing* may employ both *forward tracing* and *backward tracing*.