

---

# Übersicht und Bewertung von Traceability-Methoden

---

Im Rahmen von AP100 und fortgeführt im AP210 wurden die Traceability-Methoden unterschiedlicher Disziplinen aus dem Bereich der Forschung und der Industrie zu einem Methodenkatalog zusammengestellt und anschließend auf ihre Defizite und Stärken untersucht.

## Ziele

Das primäre Ziel der Zusammenstellung ist es, einen Überblick von bekannten Methoden der Nachverfolgbarkeit (Traceability) zu geben, wie sie in Forschung und Industrie zu finden sind.

Die Analyse fokussiert weiterhin auf den Defiziten und Vorteilen von einzelnen Methoden. Durch die Bildung von Clustern sollen allgemeine Aussagen bezüglich der auftretenden Defizite und Vorteile von Methoden in Forschung bestimmt werden.

## Vorgehensweise

Der Katalog beinhaltet insgesamt 27 Methoden, die nach Umsetzungsgrad (Methode, Prototyp, Tool) und Zuordnung zu den Phasen des Traceability-Prozesses<sup>1</sup> (Planung, Erfassung, Verwendung, Pflege) klassifiziert sind.

Ergänzt wurden die allgemeinen Informationen durch den Titel der Quelle, seinen Autor bzw. Hersteller und der Bezeichnung oder dem bereits existierenden Namen. Mittels Kurzbeschreibung wurde jeder Eintrag im Anschluss beschrieben.

Sämtliche Voruntersuchungen hatten das Ziel, Defizite und Vorteile zu identifizieren. Projektmitarbeiter haben die Methoden anhand der recherchierten Veröffentlichungen und weiterführender Materialien analysiert. Die Ergebnisse wurden zunächst in natürlich sprachlicher Form (z.B. *Der Aufwand ist reduziert mithilfe der automatischen Erstellung von Traceability Informationen*) notiert. Anschließend wurden die Einträge auf Basis ihrer Ähnlichkeit in Gruppen (Cluster) eingeteilt. Nachdem die Gruppen gebildet wurden, definierten die Projektmitarbeiter Titel für jedes Cluster (z.B. Aufwandsreduzierung).

Um die Übersichtlichkeit zu gewährleisten dürfen den Methoden maximal drei Defizite und drei Vorteile zugeordnet werden.

---

<sup>1</sup> Sünnetcioglu, Atakan; Brandenburg, Elisabeth; Auricht, Maik; Stark, Rainer (2014): Durchgängiger Traceability-Prozess im Systems Engineering. In: Tag des Systems Engineering, S. 133.

## Ergebnisse

Die Defizite von Methoden ergeben die folgenden Cluster:

Cluster (Defizite)	Beschreibung
Softwarelastig	Nur der Source-Code von einer Software wird mit anderen Artefakten verknüpft. Die Methodik funktioniert nur wenn der Source-Code analysiert wird.
Kognitiver Aufwand	Die Verknüpfung findet manuell statt und/oder wird der Anwender beim Entscheidungsprozess nicht unterstützt.
Darstellung	Die Visualisierung von Tracelinks ist nicht vorhanden / hat keinen Mehrwert, oder die GUI vom Software-System zeigt schlechte Bedienbarkeit.
Beschränkte Artefakt-Arten	Das System erlaubt dem Anwender nur bestimmte Arten von Artefakten zu verknüpfen (z.B. nur Anforderungen mit den Testcases).
Integrierbarkeit	Die Artefakte müssen ein bestimmtes Format haben wenn sie als Input zum Einsatz kommen sollen (z.B. als XML-Datei). Die Anwendung ist beschränkt aufgrund von systemtechnischen Notwendigkeiten (z.B. Plug-in für IBM DOORS).
Large-Scale System	Die Traceability-Methodik ist für die Anwendung bei einem großen System nicht geeignet.
Pflegeaufwändig	Das ganze System oder ein Bestandteil davon verursacht einen hohen Aufwand aufgrund von notwendiger Pflege (z.B. aufgrund von einem Glossar, das häufig gepflegt werden muss).

Insgesamt 44 Defizite wurden identifiziert. Die folgende Abbildung zeigt die Auftrittshäufigkeit von Defiziten:

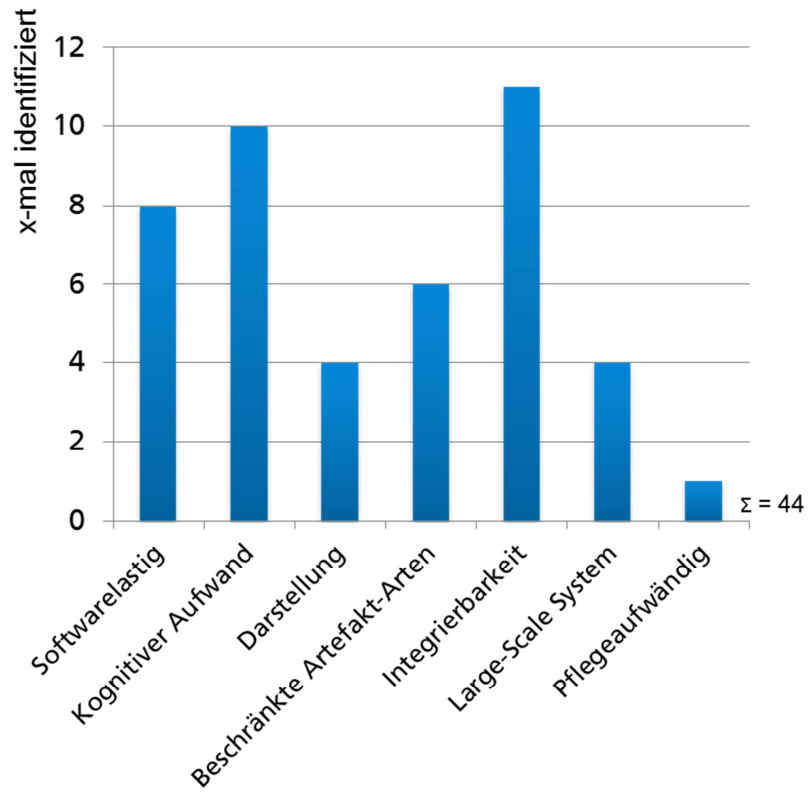


Abbildung 1: Defizite von Methoden

Die Vorteile von Methoden ergeben die folgenden Cluster:

<b>Cluster (Vorteile)</b>	<b>Beschreibung</b>
Aufwandsreduzierung	Die Methode verringert den notwendigen Aufwand für die Erfassung von Tracelinks
Anpassungsfähigkeit an Traceability (TA)-Prozess	Die Methode ermöglicht die Bestimmung von Beziehungstypen, Artefakten, Erfassungsalgorithmen etc.
Übertragbarkeit auf andere Projekte	Die Traceability-Informationen und die definierten Algorithmen können auch bei anderen Projekten verwendet werden
Integration mit anderen Systemen	Die Methode hat Schnittstellen zu anderen Systemen (z.B. mit IBM DOORS)
Kollaboratives Arbeiten unterstützt	Die Methode erlaubt die Zusammenarbeit von mehreren Anwendern und/oder sieht vor, dass mehrere Anwender mit Tracelinks arbeiten
Datentransfer möglich	Die Traceability-Informationen und/oder Berichte können mit üblichen Formaten exportiert/importiert werden
Klar definierte Anwendungsszenarien	Die Methode unterstützt mindestens ein bestimmtes Anwendungsszenario (z.B. Vollständigkeitsanalyse)

Insgesamt 51 Vorteile wurden identifiziert. Die Zuordnung der Vorteile zu den Clustern ist folgendermaßen:

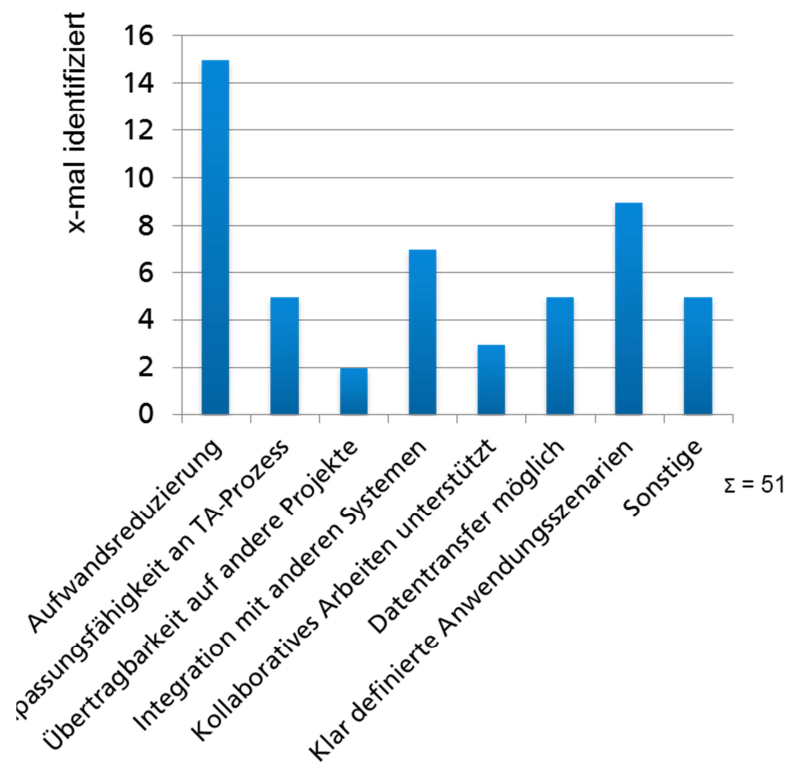


Abbildung 2: Vorteile von Methoden

## Fazit

### Defizite

Bei den Defiziten ist zu erkennen, dass die Integrierbarkeit, der kognitive Aufwand und die Softwarelastigkeit die Hauptprobleme sind.

Die Integrationsprobleme entstehen meist aufgrund der unflexiblen Voraussetzungen für die Anwendung der Methoden. Zum Beispiel [2] setzt voraus, dass die Anforderungen passend zum Natural Language Processing formuliert werden. [6] und [13] setzen voraus, dass die Artefakte ein bestimmtes Dateiformat (z.B. XML) haben, so dass die Verknüpfung möglich ist. Ein weiterer Aspekt, die Integrierbarkeit, befasst sich mit den technischen Abhängigkeiten von Methoden. Die Methoden [18] und [21] sind nur als Plug-in für ein bestimmtes System (z.B. DOORS und Eclipse) vorgesehen und damit repräsentative Beispiele für dieses Problem.

Erfassung von Tracelinks ist eine aufwändige Aufgabe [Winkler&Pilgrim]. Kognitiver Aufwand ist besonders hoch, wenn die Methode die Anwender während der Erfassung von Tracelinks nicht unterstützt. Mehrere Methoden, unter anderem [1], [3], [14] und [16] sind Beispiele, die nur eine manuelle Erfassung von Tracelinks erlaubt.

Das Forschungsgebiet der Traceability stammt aus der Softwareentwicklung [Gotel&Finkelstein]. Aus diesem Grund kommt es häufig vor, dass die entwickelten bzw. erforschten Methoden sich auf die Softwareentwicklung fokussieren. [4] und [15] verwenden die Traces von durchgeführter Software (z.B. während der Durchführung eines Testszenarios) um Tracelinks automatisch zu erstellen. Ein weiteres relevantes Problem ist die Einschränkung des Einsatzgebiets der Methoden. [8], [20] und [26] fokussieren lediglich auf die Verknüpfung vom Softwarecode mit Anforderungen.

### Vorteile

Die Analyse der Vorteile von Methoden zeigt deutlich, dass die Aufwandsreduzierung ein wichtiger Schwerpunkt in der Forschung ist. Klar definierte Anwendungsszenarien und die Integration mit anderen Systemen sind die weiteren Vorteile der analysierten Methoden.

Die Methoden streben nach der Automatisierung der Erfassung von Traceability Informationen. Am häufigsten werden Information Retrieval (u.a. [8], [9] und [18]) und Automatisierungsregeln ([2] und [23]) verwendet, um Tracelinks automatisch zu identifizieren. Eine weitere Möglichkeit für die Aufwandsreduzierung ist die Überwachung von Softwarecode während der Ausführung von Testszenarien [15]. Damit können beispielweise Tracelinks zwischen Softwarecode und Testszenarien automatisch erfasst werden.

In der Literatur wird häufig erwähnt, dass der Mehrwert von Tracelinks nicht immer bekannt ist [Winkler&Pilgrim]. In dem Methodenkatalog gibt es Methoden, die für eine bestimmte Anwendung entwickelt werden. Durch die Anwendung von Tracelinks können Anforderungen systematisch priorisiert werden [3], Browsing bzw. Navigation zwischen Artefakten ermöglicht werden [5], oder die Artefakte und Zusammenhänge visualisiert werden [16][20][21]. Ebenso ist es möglich, mithilfe von Tracelinks eine Vollständigkeitsanalyse durchzuführen [13][26].

Die Traceability Methoden müssen Informationen über die Artefakte bekommen. Die Integration von Traceability Methoden mit anderen Systemen (i.d.R. Systeme für die Produktentstehung) ist für den Informationsaustausch notwendig. Beispielsweise kann [25] mit einem PDM System (Teamcenter) über eine Schnittstelle verbunden werden, so dass Tracelinks zwischen den Artefakten im PDM System erstellt werden können. [18] und [20] sind bereits in einer Entwicklungsumgebung (Eclipse) integriert und ermöglichen damit eine direkte Erfassung von Tracelinks während der Programmierung. Ein weiteres Beispiel zeigt die Methode [13], die Schnittstellen zu mehreren etablierten Werkzeugen beinhaltet, sodass Office und Textbearbeitung, Anforderungsmanagement, Projektmanagement und gar Simulationenwerkzeuge in den Traceability Prozess einbezogen werden können.

---

## Quellen

[Winkler&Pilgrim] Winkler S, Pilgrim J (2010) A survey of traceability in requirements engineering and model-driven development. *Softw Syst Model* 9(4): 529-565. doi: 10.1007/s10270-009-0145-0

[Gotel&Finkelstein] Gotel O, Finkelstein CW An analysis of the requirements traceability problem. In: *IEEE International Conference on Requirements Engineering*, pp 94–101

## Anlagen

- opTRAC Methodenkatalog und Methodenbewertung hinsichtlich Defizite und Stärken

ID	Autor	Titel der Quelle	Methoden-/ Toolbezeichnung	Defizite (Beschreibung)			Probleme / Nachteile (Cluster)			Vorteile (Beschreibung)			Vorteile (Cluster)			Quellenangabe
				Defizit 1	Defizit 2	Defizit 3	Defizit 1	Defizit 2	Defizit 3	Vorteil 1	Vorteil 2	Vorteil 3	Vorteil 1	Vorteil 2	Vorteil 3	
1	Egyed, Grünbacher, Heindl, Biffi	Value-Based Requirements Traceability: Lessons Learned	Value-Based Requirements Traceability	Kognitiver Aufwand ist da weil die Tracelinks manuell erstellt werden. Das Ziel dieser Vorgehensweise ist es, diesen Aufwand zu reduzieren.			Kognitiver Aufwand			Reduzierung des Aufwandes durch sinnvolle Auswahl der Tracelink Granularität (Trade off between cost and benefit)	Full Life Cycle "end-to-end" traceability is also considered	Die Methode ist Software-unabhängig und universell einsetzbar ohne eine neue IT-Landschaft	Aufwandsreduzierung	Sonstige		Egyed, Alexander; Grünbacher, Paul; Heindl, Matthias; Biffi, Stefan: Value-Based Requirements Traceability: Lessons Learned. In: Requirements Engineering Conference, 2007. RE '07. 15th IEEE International.
2	Spanoudakis, Zisman, Pérez-Minana, Krause	Rule-based generation of requirements traceability relations	Rule-based generation of requirements traceability relations	Natural Language Processing ist nötig für dieses System. Die Anforderungen müssen dementsprechend formuliert werden.			Integrierbarkeit			Regeln sind übertragbar. Anwendung bei anderen Projekten wäre möglich	Reduzierung des Aufwandes durch automatische Identifizierung der Tracelinks	Verschiedene Beziehungs-Typen werden berücksichtigt. Tracelinks werden mit der passenden Art der Beziehung erstellt.	Übertragbarkeit auf andere Projekte	Aufwandsreduzierung	Anpassungsfähigkeit an TA-Prozess	Spanoudakis, George; Zisman, Andrea; Pérez-Minana, Elena; Krause, Paul (2004): Rule-based generation of requirements traceability relations. In: Journal of Systems and Software 72 (2), S. 105-127. DOI: 10.1016/S0164-1212(03)00242-5.
3	Eben, Danilidis, Lindemann	Interrelating and Prioritising Requirements on multiple Hierarchy Levels	-	Entscheidungsfindung über die Verknüpfungen erfolgt "manuell".	Anwendung von Matrizen bei großen Systemen schwer umsetzbar.		Kognitiver Aufwand	Large-Scale System		Systematisches Vorgehen bei der Priorisierung der Anforderungen			Klar definierte Anwendungsszenarien			Eben, K. G. M., Danilidis, C., Lindemann, U.: Interrelating and Prioritising Requirements on Multiple Hierarchy Levels. Unter Mitarbeit von Marjanovic, D., Storga M., Pavkovic N., Bogetic N. In: DS 60: Proceedings of DESIGN 2010, the 11th International Design Conference, S. 1055-1064.
4	Egyed, Grünbacher	Automating Requirements Traceability-Beyond the Record and Replay Paradigm	-	Footprints von Software werden verwendet um die Verknüpfungen zu erstellen bzw. identifizieren	"...For example, by highlighting artifacts and situations that requirespecial attention..."		Softwarelastig	Darstellung	Large-Scale System	Der Aufwand ist reduziert mithilfe der automatischen Erstellung von Traceability Informationen	Die Methode hilft bei der Erfassung von Tracelinks, die ohne Unterstützung schwer zu identifizieren wären.		Aufwandsreduzierung	Sonstige		Egyed, A.; Grunbacher, P. (2002): Automating requirements traceability: Beyond the record replay paradigm. In: Automated Software Engineering, 2002. Proceedings. ASE 2002. 17th IEEE International Conference on, S. 163-171.
5	Kaindl, Kramer, Diallo	Semiotomatic generation of glossary links: a practical solution	RETH (Requirements Engineering Through Hypertext)	Die Methode berücksichtigt nur Wörter. Zusammenhänge trotz der Abwesenheit von wörtlichen Ähnlichkeiten werden nicht identifiziert. Bauteile haben häufig eine standardisierte Bezeichnung und ein Glossar könnte deswegen für die Bauteile nützlich sein. Allerdings werden die Anforderungen meistens "natürlich" formuliert und diese Methode wäre für Anforderungen evtl. nicht umsetzbar.			Beschränkte Artefakt-Arten			Teilautomatisierung reduziert den Aufwand und der Anwender hat trotzdem "Kontrolle" über das System	Links werden benutzt, um Browsing zwischen Informationen zu ermöglichen		Aufwandsreduzierung	Klar definierte Anwendungsszenarien		Kaindl, Hermann; Kramer, Stefan; Diallo, Papa Samba Niang (1999): Semiautomated Generation of Glossary Links: A Practical Solution. In: Proceedings of the Tenth ACM Conference on Hypertext and Hypermedia - Returning to Our Diverse Roots: Returning to Our Diverse Roots, New York, NY, USA: ACM (HYPERTEXT '99), S. 3-12. Online verfügbar unter <a href="http://doi.acm.org/10.1145/294469.294473">http://doi.acm.org/10.1145/294469.294473</a> .
6	Spanoudakis, Zisman, Pérez-Minana, Krause	Rule-based generation of requirements traceability relations	Anm. d. Redaktion: "CLAWS Algorithmus"	XML Representationen sind notwendig um die Use Cases, Anforderungen und Object Models anhand Rules verknüpfen zu können.			Integrierbarkeit									Spanoudakis, George; Zisman, Andrea; Pérez-Minana, Elena; Krause, Paul (2004): Rule-based generation of requirements traceability relations. In: Journal of Systems and Software 72 (2), S. 105-127. DOI: 10.1016/S0164-1212(03)00242-5.
7	Cerbah, Euzenat	Traceability between models and texts through terminology	-	insb. Die Erstellung und Pflege von einem Glossar kostet viel Zeit.	nicht bei allen Artefaktkombinationen gleich gut einsetzbar - zwischen Funktionen (lösungsneutral) und Produktstruktur (lösungspezifisch) bspw. eher schlecht einsetzbar		Pflegeaufwändig	Beschränkte Artefakt-Arten		Aufwandreduzierung bei der Erfassung von Tracelink durch textuelle Ähnlichkeiten			Aufwandsreduzierung			Cerbah, Farid; Euzenat, Jérôme (2001): Traceability between models and texts through terminology. In: Natural Language for Data Bases (NLDB'00) 38 (1), S. 31-43. DOI: 10.1016/S0169-023X(01)00015-5.
8	Antoniol, Canfora, Casazza, De Lucia	Information Retrieval Models for Recovering Traceability Links between Code and Documentation	-	Die Studie konzentriert sich auf Code-Dokument Verknüpfung. Die Methode "Vector Space Information Retrieval" kann allerdings etwas flexibler eingesetzt werden.	Es ist nicht ganz klar wie diese Methodik zusätzlich zu einem bestehenden (z.B. Dokumentenmanagementsystem) integriert werden könnte.		Softwarelastig	Integrierbarkeit		IR Methode wird verwendet um den Aufwand bei der Erfassung von Traceability Informationen zu reduzieren	Gute Übertragbarkeit auf andere Projekte. Als Input werden nur Source Code und Textdokumente benötigt		Aufwandsreduzierung	Übertragbarkeit auf andere Projekte		Antoniol, G.; Canfora, G.; Casazza, G.; Lucia, A. de (2000): Information retrieval models for recovering traceability links between code and documentation. In: Software Maintenance, 2000. Proceedings. International Conference on, S. 40-49.
9	Hayes, Dekhtyar, Karthikeyan Sundaram	Advancing Candidate Link Generation for Requirements Tracing	RETRO (Requirements Tracing On-traget)	Formatänderung war nötig so dass RETRO aus Anforderungen die Vektoren erstellen konnte. (siehe S. 12)			Integrierbarkeit			Mehrere IR Methoden sind angepasst und in RETRO eingebaut.	IR Methoden reduzieren den Aufwand und unterstützen den Anwender	Gute, nachgewiesene Leistung hinsichtlich recall, und akzeptable Leistung hinsichtlich precision	Aufwandsreduzierung			Hayes, J. H., Dekhtyar, A., Sundaram, S. K. (2006): Advancing candidate link generation for requirements tracing: the study of methods. In: Software Engineering, IEEE Transactions on 32 (1), S. 4-19. DOI: 10.1109/TSE.2006.3.



ID	Autor	Titel der Quelle	Methoden-/ Toolbezeichnung	Defizite (Beschreibung)			Probleme / Nachteile (Cluster)			Vorteile (Beschreibung)			Vorteile (Cluster)			Quellenangabe
				Defizit 1	Defizit 2	Defizit 3	Defizit 1	Defizit 2	Defizit 3	Vorteil 1	Vorteil 2	Vorteil 3	Vorteil 1	Vorteil 2	Vorteil 3	
10	De Lucia, Fasano, Oliveto, Tortora	ADAMS Re-Trace: Traceability Link Recovery via Latent Semantic Indexing	Re-Trace	Eine Auswahl für die Kategorie der Master-Artefakten ist erforderlich. Im Paper sind die folgenden Artefakt-Arten genannt (im Screenshot): - Code class- Sequence diagram - Test case- Use case Ich bin mir nicht sicher ob Text-Dokumente hier als Artefakt verwendet werden können.	die Case-Studies fokussieren sich auf ein Software Entwicklungsprojekt.		Beschränkte Artefakt-Arten	Softwarelastig		Pflege der Tracelinks ist mit dieser Methode auch möglich + Visualisierung möglich	Zusammenarbeit von mehreren Menschen ist vorgesehen und unterstützt	Reduzierung des Aufwandes durch die Anwendung der LSI Methode.	Sonstige	Kollaboratives Arbeiten unterstützt	Aufwandsreduzierung	De Lucia, A., Fasano, F., Oliveto, R., Tortora, G. (2005): ADAMS Re-Trace: A Traceability Recovery Tool. In: Software Maintenance and Reengineering, 2005. CSMR 2005. Ninth European Conference on, S. 32-41.
11	Natt och Dag, Regnell, Vincenzo, Brinkkemper	A Linguistic-Engineering Approach to Large-Scale Requirements Management	ReqSimile (tool)	nur Anforderungen werden berücksichtigt.	Übersicht von Tracelinks nicht vorhanden.		Beschränkte Artefakt-Arten	Darstellung		Durch textuelle Ähnlichkeit ist es möglich die Natürlich-Sprachliche Anforderungen zu verlinken bzw. die Duplikate zu identifizieren				Aufwandsreduzierung		<a href="http://reqsimile.sourceforge.net/">http://reqsimile.sourceforge.net/</a>
12	Lin, Lin, Cleland-Huang, Settimi, Amaya, Bedford	Poirot: A Distributed Tool Supporting Enterprise-Wide Automated Traceability	POIROT TraceMaker; probabilistic inference model							Schnittstellen zu Case Tool Servern und Konvertierungsmechanismen sind vorhanden.	Tracelink Vorschläge werden generiert durch Probabilistic Network Model und dadurch wird der Aufwand geringer			Integration mit anderen Systemen	Aufwandsreduzierung	Jun Lin; Chan Chou Lin; Cleland-Huang, J.; Settimi, R.; Amaya, J.; Bedford, G. et al. (2008): Poirot: A Distributed Tool Supporting Enterprise-Wide Automated Traceability. In: Requirements Engineering, 14th IEEE International Conference, S. 363-364.
13	Dassault Systems (Reqtify)		Reqtify	Die Tracelinks zwischen den Unterlagen muss zunächst manuell erstellt werden.	"Common Requirement Format" ist erforderlich um Informationen importieren zu können.		Kognitiver Aufwand	Integrierbarkeit		Es ist möglich aus Word & PDF Dateien Einträge zu importieren.	Zusätzlich Funktionen für die Verwendung von Tracelinks vorhanden (Coverage analysis, reports generation, impact analysis etc.)	Schnittstellen vorhanden für andere etablierte Werkzeuge	Datentransfer möglich	Klar definierte Anwendungsszenarien	Integration mit anderen Systemen	<a href="http://www.3ds.com/products-services/catia/capabilities/requirements-engineering/reqtify/">http://www.3ds.com/products-services/catia/capabilities/requirements-engineering/reqtify/</a>
14	Roy		TraceCloud	GUI ist unübersichtlich (siehe Screenshots & Demo Video)	Nur Anforderungen werden miteinander verknüpft.	Links werden manuell erstellt (keine Automatisierung)...	Darstellung	Beschränkte Artefakt-Arten	Kognitiver Aufwand	Berichterstattung mit üblichen Formaten ist unterstützt	Es ist möglich, verschiedene (Projekt)Rollen zu definieren	Defect, Requirements und Test Management Systeme sind integriert. Tracelinks werden direkt innerhalb des gleichen Systems verwendet.	Datentransfer möglich	Kollaboratives Arbeiten unterstützt	Integration mit anderen Systemen	<a href="http://www.tracecloud.com">http://www.tracecloud.com</a>
15	Egyed, Binder, Grünbacher	STRADA: A Tool for Scenario-based Feature-to-Code Trace Detection and Analysis	STRADA (Scenario-based TRACE Detection and Analysis)	Tracelinks to source code (from requirements) through testing.	STRADA can be applied to any software system that can be observed during execution. +The tool is currently restricted to Java-based systems.		Softwarelastig	Integrierbarkeit		Automatische Erfassung von Tracelinks reduziert den Aufwand				Aufwandsreduzierung		Egyed, A.; Binder, G.; Grunbacher, P. (2007): STRADA: A Tool for Scenario-Based Feature-to-Code Trace Detection and Analysis. In: Software Engineering - Companion, 2007. ICSE 2007 Companion. 29th International Conference on, S. 41-42.
16	Aleksy, Hildenbrand, Obergfell, Schwind	A Pragmatic Approach to Traceability in Model-Driven Development	OMEGA TRACES	Fokussiert sich auf die Verbindung zwischen Code und Model.	Models und Anforderungen sollen manuell verknüpft werden.		Softwarelastig	Kognitiver Aufwand		Export von Traceability Informationen mittels einer XML Datei ist unterstützt	Traceability Visualization ist vorhanden um die Systeme zu analysieren		Datentransfer möglich	Klar definierte Anwendungsszenarien		Aleksy, Markus; Hildenbrand, Tobias; Obergfell, Claudia; Schwind, Michael (2009): A Pragmatic Approach to Traceability in Model-Driven Development. In: In proceeding of: PRIMMUM - Process Innovation for Enterprise Software.
17	De Lucia, Landi, Oliveto, Tortora	Enhancing IBM Requisite Pro with IR-based traceability recovery features	ReqTracer Pro - Plugin; Vector Space Model (VSM)	nur 2 Möglichkeiten für die Darstellung: List und Matrix. Sollte die Anzahl von Anforderungen sehr hoch sein, wären diese Darstellungen nicht geeignet.			Softwarelastig	Darstellung		Integriert in einem Artefakt-Management System (CASE Tool)	"Similarity threshold" einstellbar. Damit können die Ergebnisliste weniger, aber präzisere Vorschläge enthalten	Einsatz von IR Methoden generiert Vorschläge und reduziert damit der Aufwand bei der Erfassung von Tracelinks	Integration mit anderen Systemen	Anpassungsfähigkeit an TA-Prozess	Aufwandsreduzierung	Lucia, Andrea de; Landi, Raffaele; Oliveto, Rocco; Tortora, Genevieve (2008): Enhancing IBM Requisite Pro with IR-based traceability recovery features. In: submitted to Eclipse ft.
18	Bavota, Colangelo, De Lucia, Fusco, Oliveto, Panichella	TraceME: Traceability Management in Eclipse	"Lucene IR engine"; Vector Space Model (VSM)	In dem Tracelink Recovery Modus sieht der Anwender nur die Dateinamen von Artefakten.	Entwickelt als Eclipse Plug-in.		Kognitiver Aufwand	Integrierbarkeit		Methode ist integriert in einem Development Environment. Dadurch wird die "online" Erfassung unterstützt	Definition von neuen Artefakten und Artefakt-Kategorien ist möglich	Einsatz von IR Methoden generiert Vorschläge und reduziert damit der Aufwand bei der Erfassung von Tracelinks	Integration mit anderen Systemen	Anpassungsfähigkeit an TA-Prozess	Aufwandsreduzierung	Bavota, G.; Colangelo, L.; Lucia, A. de; Fusco, S.; Oliveto, R.; Panichella, A. (2012): TraceME: Traceability Management in Eclipse. In: Software Maintenance (ICSM), 2012 28th IEEE International Conference on, S. 642-645.
19	Witte, Angius	The OpenTrace Workbench for Automatic Traceability Link Recovery and Evaluation - Guide for Users and Developers	OpenTrace; irgendwie Ontologie-basiert; VSM/LSA und "vocabulary based" Tracing	Bedienung ist nicht intuitiv.	Es ist fraglich ob dieses Tool passende Schnittstellen zu den anderen Tools hat / haben könnte.		Kognitiver Aufwand	Integrierbarkeit		Tracelink Erstellung ist frei planbar (Link Einstellungen, IR Techniques, u.ä.) können situationsabhängig angepasst werden			Anpassungsfähigkeit an TA-Prozess			Angius, Elan; Witte, Rene: OpenTrace: An Open Source Workbench for Automatic Software Traceability Link Recovery. In: 2012 19th Working Conference on Reverse Engineering (WCRE). Kingston, ON, Canada, S. 507-508.
20	Marcus, Xie, Poshyvanyk	When and How to Visualize Traceability Links?	TraceViz	IR ist für die Verlinkung von Source-Code und External Documentation (auch Anforderungen) benutzt.			Softwarelastig			Visualisierung der Tracelinks ist möglich um Erfassung, Browsing und Pflege zu unterstützen	Methode ist integriert in einem Development Environment. Dadurch wird die "online" Erfassung unterstützt		Klar definierte Anwendungsszenarien	Integration mit anderen Systemen		Marcus, Andrian; Xie, Xinrong; Poshyvanyk, Denys (2005): When and How to Visualize Traceability Links? In: Proceedings of the 3rd International Workshop on Traceability in Emerging Forms of Software Engineering, S. 56-61. DOI: 10.1145/1107656.1107669.

ID	Autor	Titel der Quelle	Methoden-/ Toolbezeichnung	Defizite (Beschreibung)			Probleme / Nachteile (Cluster)			Vorteile (Beschreibung)			Vorteile (Cluster)			Quellenangabe
				Defizit 1	Defizit 2	Defizit 3	Defizit 1	Defizit 2	Defizit 3	Vorteil 1	Vorteil 2	Vorteil 3	Vorteil 1	Vorteil 2	Vorteil 3	
21	Integrate	TraceLine for DOORS	TraceLine	Plug-in für Doors. Mit anderen Systemen oder als Stand-Alone nicht anwendbar.	Nur Anforderungen werden miteinander verknüpft.	Verknüpfung erfolgt manuell.				Es ist möglich Berichte mit Traceability-Informationen zu erstellen (HTML, Visio, Mindjet Formate sind unterstützt)	Gezieltes Browsing zwischen Anforderungen und relevanten Informationen ist unterstützt (natürlich nur wenn die Links vorhanden sind!)				<a href="http://www.integrate.biz/traceLine/">http://www.integrate.biz/traceLine/</a>	
22	Maurer	LOOME - Software Tool for Multi-domain Complexity Management.	LOOME, Ableitung von Dependency Graphs	Import ist nur aus Excel möglich	Die Erstellung von Tracelinks ist manuell und es gibt keine Unterstützung dafür.		Integrierbarkeit	Beschränkte Artefakt-Arten	Kognitiver Aufwand	Visualisierung komplexer Zusammenhänge ermöglicht und vereinfacht die Beherrschung der Komplexität (in Produktentwicklung).	Daten sind zwischen Loomeo und MS Excel austauschbar.	Es gibt vier Analysemethoden (Berechnung indirekten Abhängigkeiten, Clusteranalysen, Triangularisierung, Grafen)	Klar definierte Anwendungsszenarien	Datentransfer möglich	<a href="http://loomeo.com/de">http://loomeo.com/de</a>	
23	Stark, Figge	EcoTracing - A systems engineering method for efficient tracelink modelling	EcoTracing				Integrierbarkeit			Erhebliche Aufwandsreduzierung durch Regel-basierte Tracelink Erfassung	Beliebige Artefakte können miteinander verknüpft werden		Aufwandsreduzierung	Anpassungsfähigkeit an TA-Prozess	Stark, Reiner, Figge, Asmus (2011): Eco tracing - a systems engineering method for efficient tracelink modelling. In: Cuiley, S. J. et al. (Hg.): Proceedings of the 18th International Conference on Engineering Design, Product and Systems Design.	
24	Cleland-Huang, Chang, Christensen	Event-Based Traceability for Managing Evolutionary Change	-	Zusätzliche Elemente soll in die IT-Landschaft integriert werden, wie z.B. Event Server.	die Erstellung von Tracelinks ist nicht unterstützt. Die Methode geht davon aus, dass die Tracelinks bereits existieren.	Die EBT ist für die großen Systeme mit sehr vielen Links nicht geeignet. (siehe Conclusion)	Integrierbarkeit	Kognitiver Aufwand	Large-Scale System	Anwender werden über die Änderungen in Artefakten informiert	Kollaboration ist vorgesehen und unterstützt		Kollaboratives Arbeiten unterstützt		Cleland-Huang, J., Chang, C. K., Christensen, M. (2003): Event-based traceability for managing evolutionary change. In: Software Engineering, IEEE Transactions on 29 (9), S. 796-810. DOI: 10.1109/TSE.2003.1232285.	
25	iSystems		METUS							Verschiedene Anwendungsszenarien (z.B. Varianz optimieren, Design to cost, Projektmanagement etc.) sind im System integriert verfügbar	PLM Integration ist möglich (aktuell nur für Siemens Teamcenter)	Div. Weitere Vorteile: - Drag & Drop für Verknüpfung - Berichte & Analysen - Erweiterungsmöglichkeit (insb. Automatisierung) mit Add-ins und Formeln	Klar definierte Anwendungsszenarien	Integration mit anderen Systemen	Sonstige	<a href="http://id-consult.com/metus/metus-im-ueberblick">http://id-consult.com/metus/metus-im-ueberblick</a>
26	Ahmed M. Salem	A Model for Enhancing Requirements Traceability and Analysis	Event-based Traceability with Traceability Engine (TEC), Traceability Viewer (TVC) and Quality Assurance (QA) Components	Es geht hauptsächlich um die Verknüpfung von Anforderungen mit dem Software Code.			Softwareelastig			Eine Coverage-Analyse ist vorgesehen, die zur Identifikation der nicht-erfüllten Anforderungen dient.	Änderungen in Anforderungen werden automatisch erkannt und der relevante Source-Code wird dementsprechend markiert.		Klar definierte Anwendungsszenarien		Salem, Ahmed M (2010): Model for Enhancing Requirements Traceability and Analysis. In: IUACSA 1 (5). DOI: 10.14569/IUACSA.2010.010503.	
27	Alexander Egyed	A Scenario-Driven Approach to Traceability Tool heißt "TraceAnalyser"	-	(Abstract): this paper demonstrates how observable run-time information of software systems can be used to detect traceability information between software systems and their models. Tracelinks zwischen Szenarien und Model Elements können aber auch identifiziert werden.	das Footprint Graph wäre zu groß bei einem komplexeren System.		Softwareelastig	Large-Scale System		Iteratives Vorgehen ist möglich und empfohlen. So kann die Analyse auch relativ frühzeitig starten. Die erste Iteration zeigt, welche Input-Daten sinnvoll wären. Die Ergebnisse werden nach jeder Iteration präziser.	Automatisierter Ansatz ermöglicht eine effiziente Erfassung von Tracelinks mit weniger Aufwand		Sonstige	Aufwandsreduzierung	Egyed, A. (2001): A scenario-driven approach to traceability. In: Software Engineering, 2001. ICSE 2001. Proceedings of the 23rd International Conference on, S. 123-132.	