



□ Gernot Glawe

(gernot.glawe@acando.de)
 ist im Business Consulting der Management- und IT-Beratung Acando tätig. Glawe ist Senior Consultant mit dem Schwerpunkt Vorgehensmodelle. Er befasst sich außerdem mit mit Requirements Engineering und der Dokumentation von Softwareprojekten.



□ Jakob Müller

(jakob.mueller@acando.de)
 absolvierte sein Studium der Wirtschaftsinformatik an der Dualen Hochschule Baden-Württemberg Stuttgart. Seit 2010 arbeitet er im Bereich Business Consulting der Management- und IT-Beratung Acando GmbH. Seine Schwerpunkte liegen im Projektmanagement und Requirements Management. Derzeit unterstützt er einen Großkunden im Finanzsektor.

Verfolgungsjagd mit einfachen Mitteln: Wie Traceability auf Office-Basis gelingt

Während bei kleineren und agilen Vorhaben oft ohne Traceability gearbeitet wird, ist dies bei komplexen Entwicklungsprojekten nicht mehr möglich. Bei Software, die internen oder externen Prüfungen standhalten muss oder gesetzlichen Regelungen unterliegt, ist ein klarer Nachweis dafür erforderlich, dass die Anforderungen wie beschrieben umgesetzt sind. In solchen Fällen ist Aufwand für die Nachverfolgbarkeit erforderlich. Best Practices aus dem Projektalltag legen eine integrierte Lösung nahe: Anforderungen werden innerhalb der Dokumentation logisch miteinander verknüpft und Verbindungen zu Arbeitsdokumenten werden automatisch geschaffen.

Wozu Traceability?

Anforderungen an ein Produkt werden in einem Text beschrieben. Will man später auf Textstellen verweisen, sind Angaben wie „Seite 3, Absatz 2“ unhandlich. Deswegen sind Texte, mit denen viel gearbeitet wird, auch anders strukturiert. Ein prominentes Beispiel dafür ist die Bibel. Ein Verweis auf Matthäus, Kapitel 20, Vers 16 ist eindeutig, egal auf welcher Seite der Text steht. So wird auch Anforderungstext in Elemente aufgeteilt, d. h. in Requirements, die auch isoliert verständlich sind.

Dabei stellt eine lückenlos umgesetzte Traceability (Nachverfolgbarkeit) sicher, dass das Endprodukt tatsächlich den ursprünglichen Anforderungen entspricht. Zwischen den verschiedenen Entwicklungsphasen sind Medienbrüche zu überwinden und auch beim Wechsel von Verantwortlichkeiten darf der Zusammenhang zwischen einer Anforderung und ihrer späteren Umsetzung nicht verloren gehen.

Was ist daran so schwer? Der Prozess der Anforderungsermittlung lässt sich veranschaulichen am Beispiel von vier Personen, die nebeneinandersitzen: Die erste Person beschreibt eine Idee für ein Produkt in vier

eindeutigen Elementen und gibt sie als Anforderungen Nr. 1 bis 4 an den Nachbarn. Der verfeinert sie, indem er daraus z. B. 20 Informationselemente macht. So geht es weiter bis zur vierten Person – und schnell wird es sehr unübersichtlich.

Der Übergang zwischen den Phasen ist also einer der anspruchsvollsten Teile einer Systembeschreibung. Meist handelt es sich nicht nur um einen Medienbruch, sondern auch um einen Personen-, Methoden-, Philosophie-, Sprach- und Paradigmenwechsel. Es besteht also ein erhöhtes Risiko, dass Informationen verloren gehen oder falsch bzw. missverständlich übermittelt werden. Deshalb sollte den Bearbeitern bei jedem Phasenübergang klar sein, auf welche vorangegangenen Anforderungen sich die Artefakte (Übermittlungsergebnisse) jeweils beziehen. Dies leistet eine Traceability-Lösung.

Am vereinfachten Beispiel des Rational Unified Process (RUP) lassen sich die Beziehungen der Informationselemente von Phase zu Phase nachvollziehen. Für eine eindeutige Identifikation werden die Requirements üblicherweise fortlaufend mit den Anfangsbuchstaben der Phasen nummeriert:

- Phase 1) **Inception:** Definition der fachlichen Ziele aus einer Vision/Idee.
 Artefakt: Präsentationsmedien
 Element: Beschreibung eines Teils der Idee, z. B. „Das Produkt soll drucken können“ (I1)
- Phase 2) **Elaboration:** Ableitung von Anforderungen für die User Requirements (UR) und Erstellung des Fachkonzepts.
 Artefakt: Formatierter Text
 Element: Absatz im Dokument, der z. B. eine Anforderung beschreibt, z. B. „Drucken von PDF-Dokumenten“ (E1)
- Phase 3) **System-Design:** Technische Beschreibung der Anforderungen (System Requirements)
 Artefakt: UML-Modell
 Element: Usecase im Usecasediagramm, z. B. „Schnelldruck von PDF-Dokumenten“ (S1)
- Phase 4) **Construction:** Umsetzung der Requirements
 Artefakt: Programmiersprache
 Element: eine Funktion im Programmcode, z. B. „print_pdf()“ (C1)

In jeder Phase entstehen unterschiedlich feine Requirements, die in der nächsten Phase durch Lösungselemente erfüllt werden. Diese Lösungselemente stellen selbst wiederum Requirements für die Folgephase dar. So wird im Beispiel Element I1 durch E1 erfüllt und E1 ist ein Requirement für S1.

Überträgt man diese Struktur von aufeinander bezogenen Anforderungen auf ein Projekt mit mehreren Hundert Elementen, wird leicht deutlich, dass ein Beziehungsmanagement der Requirements notwendig ist, also Traceability.

Drei Wege führen zu Traceability

Einfach, aber begrenzt: die externe Traceability-Matrix

Diese Lösung ist relativ einfach und eignet sich für kleine Teams, die sich erstmals mit Requirements Engineering beschäftigen. Es werden die Requirements in der Phase 2, der Elaboration (Fachkonzeption), als Fließtext erfasst und bereits vorstrukturiert (z.B. nach Prozessen, Funktionen und Daten).

Dies ist in vielen Unternehmen bereits gängige Praxis. Die Elemente werden mit eindeutigen Schlüsseln versehen (E1, E2...). Elemente späterer Phasen referenzieren jeweils auf die ID des oder der entsprechenden Requirements. Um deren Eindeutigkeit und eine Gesamtsicht zu gewährleisten, werden alle ID's an zentraler Stelle vergeben und dort mit Titel und Bezug zu den jeweiligen Elementen in den verschiedenen Phasen versehen.

Der Aufwand für die erstmalige Implementierung ist gering, lediglich einige Vorlagen müssen erstellt werden. Ausgaben für weitere Software fallen nicht an. Die Kosten für die Pflege der Matrix sind dagegen hoch.

Weitere Nachteile sind: geringe Flexibilität und hohe Fehleranfälligkeit, gerade bei umfangreichen Vorhaben. So ist eine Referenzierung zwischen verschiedenen Anforderungen zwar prinzipiell möglich, kann aber praktisch nur auf stabilen Versionen erfolgen – Datei- und Verzeichnisnamen müssen dabei immer gleich lauten.

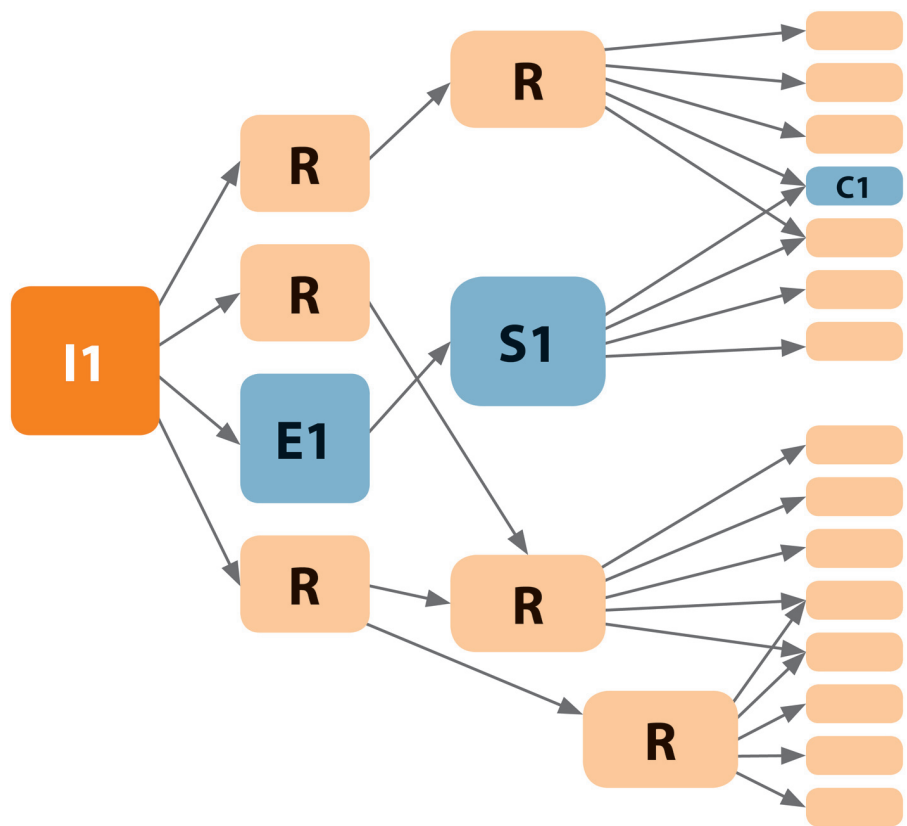


Abb. 1: Traceability von Requirements über Projektphasen

Dies widerspricht allerdings der Einfachheit der Lösung, denn die Erfahrung aus der Praxis zeigt, dass Dokumente oder ganze Verzeichnisbäume während des Projektverlaufs umbenannt werden. Diesem Risiko kann man mithilfe von Makros begegnen, die die Validität der Hyperlinks bei jedem Versionsstand automatisch prüfen.

Wie sich schon bei der einfachen Beispieltabelle zeigt, wird die Matrix schnell unübersichtlich. Eine weiterführende Lösung ist dann eine Datenbank für Requirements und ihre Beziehungen.

Komfortabel, aber aufwendig: die Datenbank

Bei diesem Vorgehen werden nicht nur die Anforderungs-IDs, sondern die kompletten Anforderungen, also formatierter Text der Phase 2 zentral in einer Datenbank erfasst.

Die vormaligen Pflichtenhefte bzw. Fachkonzepte aus dieser Phase enthalten nur noch einen Rahmen zusammengehöriger Anforderungen, der bei Bedarf durch entsprechende Exportfunktionalitäten mit Inhalten aus der Datenbank gefüllt wird. Das Fachkonzept wird also aus einer Requirements-Datenbank erzeugt.

Die Pflege der Requirements erfolgt dabei ausschließlich in der Datenbank, ggfs. unterstützt durch Importfunktionen. Der lesende Zugriff kann nun entweder in einer Suchfunktion der Datenbank oder in einem durch einen Export erzeugten Dokument stattfinden. Letzteres findet häufig eine höhere Akzeptanz, da es in einleitenden Kapiteln Hintergrundinformationen geben kann. Darin dürfen allerdings selbst keine Requirements verborgen sein. Eine gültige Hintergrundinformation wäre etwa ein Gesetzestext, der Grundlage für eine gewünschte Funktionalität des Produktes oder Programmes ist.

Mit der Datenbanklösung werden Redundanz- und Konsistenzprobleme stark reduziert und Verknüpfungen zwischen Anforderungen sowie zwischen den einzelnen Phasen sind immer aktuell. Allerdings sind die Kosten und der Know-how-Bedarf für

Inception	Elaboration	System Design	Construction
I1	E1,E2,E3,E4		
	E1	S1	
		S1	C1,C2,C3,C4,C

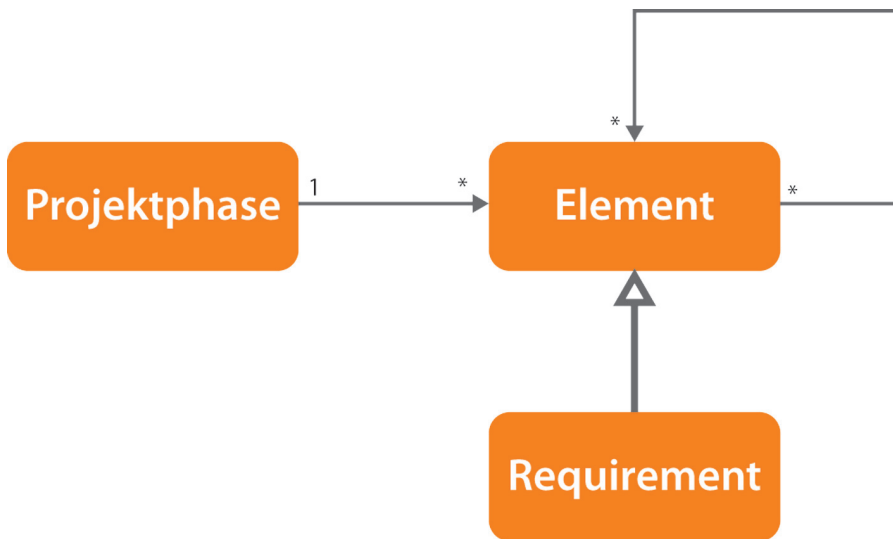


Abb. 2: Vereinfachtes Modell der Datenbanklösung

Einrichtung und Wartung höher als bei der externen Traceability-Matrix. Dafür sind aber geringere laufende Kosten für die Pflege der Requirements zu erwarten.

Intelligent, aber hausgemacht: die integrierte Nachverfolgbarkeit

Eine intelligente dritte Möglichkeit ist die integrierte Nachverfolgbarkeit auf Basis von Office-Produkten und Script-Lösungen. Dieses Konzept entlastet Anwender und Systemdesigner und trägt gleichzeitig dem kreativen Prozess der Anforderungsbeschreibung Rechnung.

Die Attraktion der integrierten Traceability-Lösung ist, dass man direkt aus dem Lesen oder Bearbeiten eines Elements zu anderen Elementen der Vor- oder Nachphase springen kann. Ein erster Ansatz dazu ist die eingebettete Verlinkung, bei der die Information einfach per Klick auf einen Hyperlink direkt im Text aufgerufen wird.

Die Praxis zeigt jedoch, dass sich Anwender durch die vielen Links im Text irritiert und im Lesefluss beeinträchtigt fühlen. Der Ärger darüber kann sich negativ auf die Akzeptanz des Inhalts auswirken.

Ein weiterer Nachteil der eingebetteten Verlinkungen besteht darin, dass sie Abhängigkeiten zwischen einzelnen Textstellen innerhalb der Requirements, aber nicht zwischen den Requirements selbst herstellen.

Die integrierte Verlinkung löst die Probleme. Dabei wird der Text von Anfang an in Elementen, den Requirements, erzeugt. Um diese in Beziehung zueinander

zu setzen, wird nicht nur sequenziell Text erzeugt, sondern von Anfang an eine Struktur für jede Beschreibung von Requirements angewandt. Bewährt hat sich dabei der folgende dreistufige Aufbau:

- 1) ID und Bezeichnung
Beispiel: E1 – Drucken von PDF-Dokumenten
- 2) Inhalt
- 3) Weitere Metainformationen, wie Links zu vor- und nachgelagerten Requirements und sonstige Links

Beispiel:

Traceability-Informationen: Erfüllt I1 – Drucken; wird durch S1 – Usecase PDF-Drucken – erfüllt.

Weitere Informationen: Link zu rechtlichen Hintergründen.

Eine ähnliche Struktur und Hintergründe dazu beschreibt auch Sissi Kloss [Klo06].

Erzeugung eines Requirements:

Um ein Requirement für die integrierbare Nachverfolgbarkeit zu erstellen, haben sich folgende Schritte als sinnvoll erwiesen:

1. Neues Requirement aus einem Template erzeugen
Dabei ist es für die Handhabung und die Akzeptanz wichtig, dass der Inhaltsbereich über die gesamte Seitenbreite geht und genügend Platz bietet. Tabellenartiger Aufbau von Requirements verringert den Platz und damit die Lesbarkeit.

2. Inhalt schreiben
3. IDs der verbundenen Requirements eingeben. Sie fungieren als Platzhalter und werden im Publishing Process durch die entsprechenden Links ersetzt.
4. Publishing: Die Veröffentlichung des Requirements für das Projektteam sollte aus Akzeptanz- und Aufwandsgründen möglichst einfach ablaufen, z. B.:

- das Dokument per E-Mail versenden
- ein Word-Makro schreiben
- das Word-Dokument auf eine Webseite hochladen

Publishing

Veröffentlicht werden nur stabile Versionen. Je nach Kontext ist zu entscheiden, ob damit auch ein Abnahmeprozess verbunden ist. Der Publishing-Process ersetzt die ID durch die Links auf die verbundenen Requirements. Sind diese nicht mehr vorhanden oder wurden sie seit der letzten Version geändert, erhält der Autor eine Nachricht und behebt den Verlinkungsfehler.

Bei konsequenter Anwendung werden häufige Fehlerquellen minimiert. Entfällt z. B. eine Anforderung oder kommt eine neue hinzu, wird dies in den nachfolgenden Phasen automatisch berücksichtigt. Erweiterungen, wie Prüfung auf Änderung und automatische Mitteilung an die Bearbeiter der verbundenen Requirements aus anderen Phasen, sind denkbar.

Das Dokument wird anschließend auf der für das Projekt gewählten Plattform publiziert. Details dazu können unter dem Pattern „Big Picture“ in [Rüp03] nachgelesen werden, wo sich auch andere interessante Ansätze zu Projektdokumentationen finden.

Auf die zentrale Publishing-Plattform verweisen auch die Requirements Links. Wenn der Leser im publizierten Dokument auf den Link klickt, erreicht er immer eine stabile, weil publizierte Version. Der Autor kann nun seine aktuelle Version weiterbearbeiten, ohne den Arbeitsprozess der Autoren der vor- und nachgelagerten Requirements zu stören.

Welche Lösung für wen?

Das Vorgehen mit der externen Traceability-Matrix kann aufgrund der recht einfachen Handhabung durchaus ein erster Einstieg in die systematische Nachverfolgung von Requirements sein. Um Konsis-

tenz zu gewährleisten, ist allerdings ein hohes Maß an Disziplin seitens der Anwender erforderlich. Eine Unterstützung durch Automatismen, Tools und Makros wird schnell aufwendig.

Datenbankbasierte Lösungen sind vielfältig im Einsatz. Sie haben sich in der Praxis bereits bewährt, aber die Einführung der Skills, Prozesse und neuen Software lohnt sich nur bei komplexen Projekten oder als firmenweite Lösung.

Für die Methode der integrierten Nachverfolgbarkeit ist keine umfassende technische Schulung der Mitarbeiter nötig. Die Anwender müssen lediglich die vereinbarten Arbeitsschritte einhalten und auf die

korrekte Verwendung der IDs achten. Alle anderen Informationen werden im Rahmen der Lösung automatisch aufbereitet.

Dieses Konzept eignet sich auch zur Vorbereitung auf die Einführung einer integrierten Requirements Management-Software. Dabei wird der verminderte Aufwand auf Seiten der Requirements-Bearbeiter mit einem Programmieraufwand für die Publishing-Skripte erkaufte.

Viele Erkenntnisse kommen bekanntlich erst in der Praxis. Wenn man also zunächst einen einfachen Requirements-Prozess anwendet, erhält man wertvolle Hinweise darauf, welche Lösung man tatsächlich braucht. ■

Quellen

[HUL05] Requirements Engineering Second Edition; Kindle Edition

[Klo06] Sissi Kloss; Single Source Publishing. Topicorientierte Strukturierung und DITA

[Rüp03] Rüpung, Andreas; Agile Documentation: A Pattern Guide to Producing Lightweight Documents for Software Projects (Wiley Software Patterns Series)

[VUE] Open Source Vektorzeichenprogramm, schnell durch Tastaturbedienung:

<http://vue.tufts.edu/>



Acando ist eine Unternehmensberatung, die in Partnerschaft mit ihren Kunden nachhaltige Verbesserungspotenziale identifiziert und diese mithilfe von Informationstechnologien implementiert. Acando bietet ein ausgewogenes Verhältnis von hohem Kundennutzen, kurzen Projektzeiten und geringen Kosten. Die Gruppe beschäftigt rund 1.000 Mitarbeiter in fünf europäischen Ländern, der jährliche Umsatz beträgt etwa 170 Millionen Euro.

Acando AB ist an der NASDAQ OMX Nordic gelistet. Unsere Unternehmenskultur basiert auf drei Kernwerten: Teamgeist, Resultate und Leidenschaft.

Die Kunden der Management- und IT-Beratung profitieren von einem stark ausgeprägten Geschäftsbereich Business Consulting mit großer Expertise in den Bereichen Prozessoptimierung, Projektmanagement, Requirements Management und Business Performance Management. Acando Consultants gestalten Projekte mit detaillierter Prozesskenntnis sowie methodischem und technologischem Know-how und werden kontinuierlich für neue Standards und Technologien zertifiziert.

www.acando.de