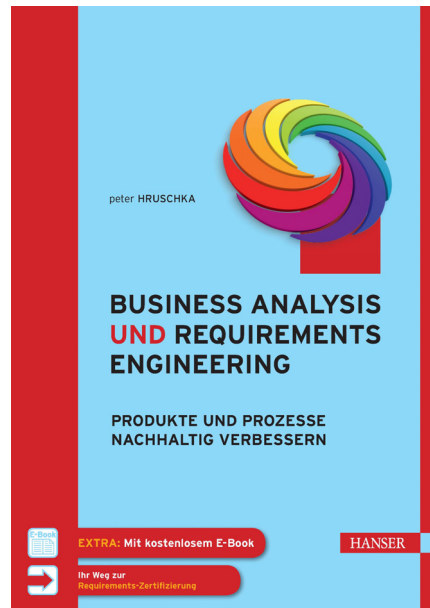


HANSER



Leseprobe

zu

„Business Analysis und Requirements Engineering“

von Peter Hruschka

ISBN (Buch): 978-3-446-43807-1

ISBN (E-Book): 978-3-446-43862-0

Weitere Informationen und Bestellungen unter
<http://www.hanser-fachbuch.de/978-3-446-43807-1>

sowie im Buchhandel

© Carl Hanser Verlag München

■ 1.7 Definition: Business Analysis und Requirements Engineering

Bevor wir uns mit den Inhalten von Business Analysis und Requirements Engineering näher beschäftigen, sollten wir uns auf Definitionen einigen. Was also verbirgt sich hinter den beiden Begriffen?

Das International Requirements Engineering Board [IREB], dessen Programm diesem Buch zugrunde liegt, hat die Gesamtheit des Themas als Requirements Engineering bezeichnet. Die beiden Hauptteile, die dazu gehören, sind einerseits die Requirements-Analyse – von vielen auch als Systemanalyse bezeichnet – und andererseits Requirements-Management, d. h., wenn ich einmal Anforderungen habe, diese auch zu managen, zu verwalten, zu pflegen. Zu den Unterpunkten der Requirements-Analyse gehört einerseits, Requirements aus Kunden herauszulocken, und andererseits, diese auch zu Papier zu bringen – also Requirements ermitteln und spezifizieren.

Sie sehen an dem einfachen Mengendiagramm in Bild 1.3 die drei Teile, aus denen dieses Gebiet besteht:

- Anforderungen ermitteln (oder herauslocken),
- Anforderungen spezifizieren (auf irgendeine Art niederschreiben oder zeichnen) und
- Anforderungen managen (verwalten, pflegen, versionieren, priorisieren).

Die ersten beiden davon werden oft auch als „Requirements-Analyse“ oder „Systemanalyse“ zusammengefasst. Welche Wörter Sie für die Summe aller dieser Tätigkeiten verwenden, überlasse ich Ihrer Entscheidung. Das IREB hat sich für „Requirements Engineering“ als Oberbegriff entschieden; ob Sie das in Ihrem Umfeld als Requirements Engineering bezeichnen oder Requirements-Management als Oberbegriff verwenden oder vielleicht Problemanalyse oder Business-Analyse, überlasse ich Ihnen.

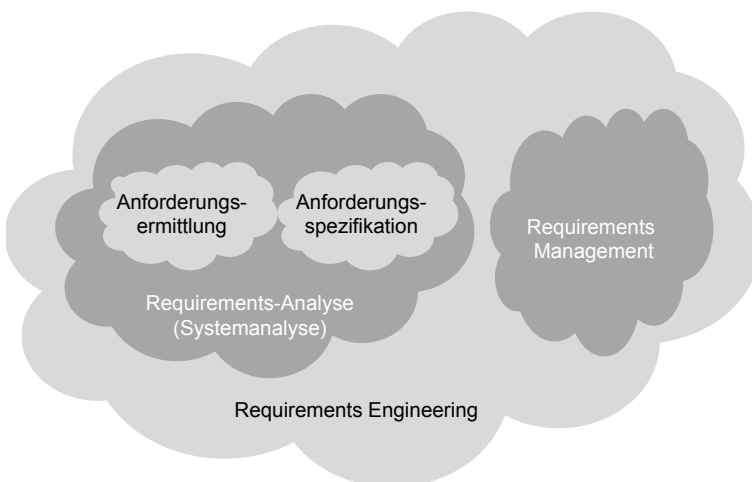


BILD 1.3 Die Hauptthemen im Requirements Engineering

Definition: Requirements Engineering

Ich möchte Ihnen aber neben diesem Mengendiagramm doch noch eine ausführliche Definition geben. Requirements Engineering im heutigen Sinne ist ein kooperativer, iterativer und inkrementeller

Prozess mit folgenden drei Zielen:

1. Alle relevanten Anforderungen sollen bekannt und im erforderlichen Detaillierungsgrad verstanden sein.
2. Die involvierten Personen und Organisationen (Stakeholder) sollen ausreichende Übereinstimmung über die bekannten Anforderungen erzielen.
3. Die Anforderungen sollen konform zu den Dokumentationsvorschriften der Organisation spezifiziert sein.

Alleine in der Einleitung dieser Definition stecken drei Aussagen drin. Ein kooperativer Prozess heißt: Wir arbeiten zusammen und nicht gegeneinander. Wir wollen nicht Anforderungen über den Zaun werfen zu anderen Leuten, sondern wir wollen sie gemeinsam entwickeln zwischen Auftraggeber und Auftragnehmer.

Die Wörter „iterativ“ und „inkrementell“ richten sich gegen das alte Wasserfallmodell. Ich muss nicht alle Anforderungen am Anfang perfekt kennen. Ich kann in Iterationen – schrittweise – Teile davon, weitere Teile davon, noch mehr Teile davon zeitgerecht entwickeln.

Danach lernen wir durch die Definition die drei wesentlichen Ziele dieses kooperativen Prozesses kennen.

Das erste Ziel ist: Ich möchte alle **relevanten** Anforderungen so gut und im **erforderlichen Detaillierungsgrad** kennen, dass ich eine Lösung darauf basieren lassen kann. Sehen Sie die Weichmacher in dieser Definition? Alle relevanten Anforderungen in hinreichendem Detaillierungsgrad? Was ist relevant? Was bedeutet hinreichender Detaillierungsgrad? Wir werden uns ausführlich mit beiden Fragen auseinandersetzen.

Das zweite Ziel besagt, dass die beteiligten und betroffenen Personen, die Stakeholder, die Organisationen, die mitspielen müssen, hinreichende Übereinstimmungen über diese Anforderungen erzielen sollten. Nun ja, in der Diktatur reicht es, wenn der Diktator ja sagt. Wenn Sie etwas demokratischer aufgestellt sind, brauchen Sie vielleicht die Zustimmung von ein paar mehr Leuten. Aber selbst in einer Demokratie sind 50,01 % eine Mehrheit. Es wird also immer wieder Personen geben, die dagegen sind, die andere Wünsche haben. Wir wollen aber, dass der Prozess dafür sorgt, dass wir eine hinreichende Übereinstimmung über die bekannten Anforderungen erzielt haben.

Und das dritte Ziel für diesen Prozess Requirements Engineering ist noch einfacher. Wir wollen, dass die Anforderungen nach den Regeln des Hauses, nach Ihren Spielregeln, nach Ihren Dokumentationsvorgaben aufgeschrieben werden. Wenn Sie in Ihrem Haus gar keine Vorschriften haben, wie man ein Pflichtenheft oder ein Lastenheft erstellt, sind Sie auf jeden Fall konform dazu ☺. Aber die meisten Firmen haben natürlich Vorgaben, was alles festzuhalten ist und wie es festgehalten werden sollte.

Requirements Engineering ist also ein kooperativer Prozess, der in Zyklen durchgeführt wird. In jedem Zyklus muss ich genügend herausfinden für das, was ich demnächst umsetzen möchte. Ich brauche genügend Übereinstimmung und ich muss es hausgerecht festhalten, nach unseren Dokumentationsvorschriften.

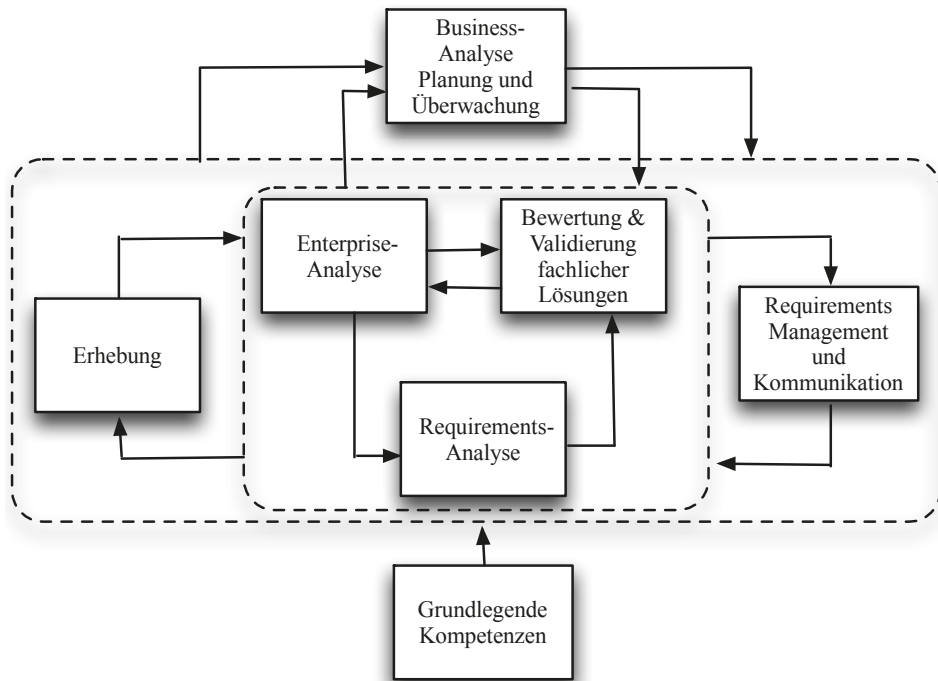


BILD 1.4 Wissensgebiete der Business-Analyse

Wenden wir uns nun dem etwas weiter gefassten Begriff „Business Analysis“ zu. Das International Institute of Business Analysis [IIBA] definiert in seinem Business Analysis Body of Knowledge [BABOK] Folgendes:

Business Analysis ist eine Menge von Aufgaben und Techniken, die genutzt werden, als Liaison zwischen Stakeholdern zu arbeiten, um die Struktur, die Direktiven (Policies) und die Arbeitsweise einer Organisation zu verstehen und (fachliche) Lösungen vorzuschlagen, die es der Organisation ermöglichen, Ihre Ziele zu erreichen.

Definition: Business Analysis

Das entsprechende Überblickbild über Business-Analyse identifiziert acht Kernbereiche, die dort als Wissensgebiete (Knowledge Areas) bezeichnet werden (vgl. Bild 1.4).

Sicherlich sehen Sie die starken Überlappungen zu der Definition von Requirements Engineering. Auch hier stehen die Erhebung (von Anforderungen) und das Managen von Anforderungen rechts und links im Fokus. Analysieren wurde hier dreigeteilt: Enterprise-Analyse und Requirements-Analyse, natürlich nicht ohne beides zu bewerten und zu validieren. Eingerahmt werden diese Kerntätigkeiten durch die Planung und Überwachung des Gesamtverfahrens sowie die Forderungen nach grundlegenden Kompetenzen der Mitwirkenden, die wir später in diesem Kapitel genauer betrachten werden.

Was ist den beiden Definitionen gemeinsam? Sowohl bei Business Analysis wie auch beim Requirements Engineering geht es um das Verstehen eines Systems (d. h. eines Geschäfts, einer Organisation, eines Produkts oder eines Systems), mit dem Ziel, Schwachstellen zu identifizieren, Ideen für Verbesserungen zu entwickeln, damit zu besseren oder neuen Pro-

dukten bzw. zu optimierten Ablauf- und Aufbauorganisationen zu kommen. Bild 1.5 fasst die wesentlichen Aspekte von Business Analysis und Requirements Engineering zusammen:

- Wir gehen davon aus, dass es jemanden gibt, der mit dem Ist-Zustand von Produkten oder Organisationen nicht zufrieden ist. Jemand, der ehrgeizige Ziele hat, Wünsche und Ideen und Visionen, oder aber durch äußere Zwänge wie Gesetze und andere Auflagen gezwungen wird, am Status-quo etwas zu verändern.
- Durch Analyse des heutigen Zustands von Organisationen und Produkten identifiziert man Schwachstellen und entwickelt Lösungsideen und Verbesserungsvorschläge, die man in Form von möglichst präzisen Anforderungen an jemanden übergeben kann, der in der Lage ist, die Lösung in die Praxis umzusetzen. Entweder mittels IT-Systemen oder aber auch durch organisatorische Maßnahmen.
- Diese Tätigkeiten werden gemeinsam von allen Stakeholdern durchgeführt, die ein Interesse an der Sache haben, in einem kooperativen Prozess, wobei wir ausgewogen über kurz-, mittel- und langfristige Lösungen nachdenken. Je kurzfristiger wir etwas ändern wollen, desto genauer müssen wir Anforderungen spezifizieren. Je langfristiger Themen sind, desto vager können wir im Ergebnis bleiben. Durch iteratives und inkrementelles Vorgehen sichern wir jederzeit die Konzentration und Bündelung unserer Kräfte auf das zurzeit Notwendige.

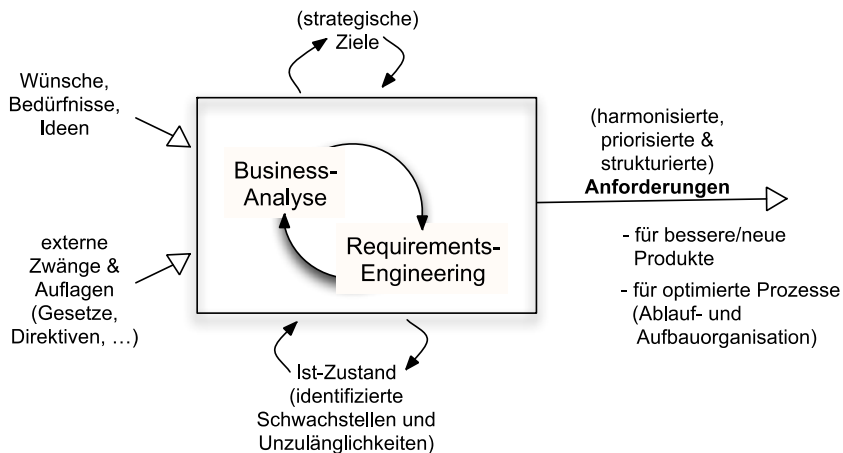


BILD 1.5 Business-Analyse und Requirements Engineering

Intermezzo

Bevor wir uns im Detail verlieren – oder nach so vielen Seiten Buch schon verloren haben –, gehen wir nochmals zurück, ganz an den Anfang. Business Analysis und Requirements Engineering streben nach einem guten Verständnis heutiger Geschäftsprozesse oder Produktfunktionalität, damit wir gezielt Vorgaben für Verbesserungen, Erweiterungen, Modifikationen etc. machen können. Die Ergebnisse dieser Analysetätigkeiten und die Erkenntnisse wollen wir irgendwie „zu Papier bringen“ oder festhalten, damit sie uns als Diskussionsgrundlage dienen und auch als Vorgaben für diejenigen, die uns helfen, die erkannten Probleme zu lösen und unsere Produkte und Prozesse nachhaltig zu verbessern.

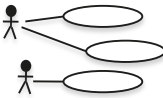

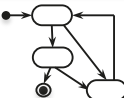
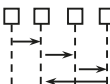
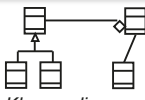
Um die Ergebnisse des Analyseprozesses festzuhalten, stehen uns heute eine Vielzahl unterschiedlicher Ausdrucksmittel zur Verfügung. Ich möchte Ihnen, nach den vielen Details in den vorangegangenen Kapiteln, in einem Überblick nochmals die Alternativen aufzeigen, die Ihnen zum Festhalten zur Verfügung stehen.

In der folgenden großen Matrix sehen Sie in der **vertikalen** Achse die unterschiedlichen Arten der Anforderungen: Wir studieren einerseits unsere Geschäftsprozesse oder Produktprozesse, natürlich zusammen mit den Dingen, die dabei verbraucht, bearbeitet oder erzeugt werden. Und wir interessieren uns für Qualitätseigenschaften und Randbedingungen.

Betrachten wir das mit etwas anderen Worten. Wir haben im Wesentlichen funktionale Anforderungen, das, was ein System oder ein Produkt tun soll, die Funktionen und die Daten und das Verhalten des Systems und wir haben daneben noch die beiden anderen Kategorien Qualitätsanforderungen und Randbedingungen. Die funktionalen Anforderungen können wir in die geschäftliche Abläufe oder Produktfunktionalität und in die Daten gliedern. Wir betreiben ja schließlich Informationsverarbeitung: Information sind unsere betrieblichen Daten oder Produktdaten; und die Verarbeitung sind die dazugehörigen Abläufe.

In der **horizontalen** Achse sehen Sie die drei prinzipiellen Möglichkeiten, die Ergebnisse zu erfassen. Alle diese Anforderungen, funktionale Anforderungen und nichtfunktionale Anforderungen, also Qualitätsanforderungen und Randbedingungen, kann ich auf drei unterschiedliche Arten darstellen.

1. Die Form der Spezifikation kann, wie in vielen Projekten heute noch häufig verwendet, einfach Text sein; umgangssprachliche, deutsche, englische oder andere natürlichsprachige Sätze.
2. Wir haben seit 30 Jahren aber auch eine Vielzahl von grafischen Modellen, die wir nutzen können, um die unterschiedlichen Arten der Anforderungen zu Papier zu bringen.
3. Und last but not least stehen uns auch Prototypen zur Verfügung, um Anforderungen transparenter zu machen.

			Form der Spezifikation		
Arten von Anforderungen			Text	(grafische) Modelle	
Funktionale Anforderungen	Abläufe	grob	Das System soll diesen Prozess unterstützen.	 Use-Case-Diagramm	Prototyp
		fein, linear	Das System soll Schritt1 ausführen. Danach soll das System Schritt 2 ausführen.	 Aktivitätsdiagramm	
		asynchron	Wenn das Ereignis eintritt, soll das System in den Zustand ... wechseln	 Zustandsdiagramm (StateChart)	
		beispielhaft	Das System soll folgende konkrete Schrittfolge ausführen: Schritt 1; Schritt 2; Schritt	 Sequenzdiagramm	
	Daten		Glossar mit alphabetisch geordneten Begriffsdefinitionen	 UML-Klassendiagramm oder Entity-Relationship-Diagramm	
Qualitätsanforderungen			Die Funktion soll folgende Qualitätseigenschaften aufweisen:	Jeweils verknüpft mit funktionalen Anforderungen (d.h. evtl. angehängt an bestimmte Modellelemente)	
Randbedingungen			Folgende Randbedingungen sind einzuhalten: ...		

Unterschiedliche Spezifikation der Analyseergebnisse

Lassen Sie uns im Einzelnen sehen, was das für die unterschiedlichen Arten von Anforderungen bedeutet.

Beginnen wir mit den Abläufen in Textform. Auf einer sehr groben Ebene kann ich ganze Prozesse in einem deutschen Satz fordern. „Das System soll Prozess A unterstützen“; oder „das System soll den Geschäftsprozess B unterstützen“ sind umgangssprachliche Sätze, die ganze, eventuell sehr komplexe Abläufe fordern.

Die grafische Alternative dazu sind Use-Case-Diagramme. Wir zeichnen all die gewünschten Prozesse als Ellipsen auf und die dazugehörigen Auslöser, die Akteure, als Strichmännchen. Sie entscheiden, ob Sie lieber deutsche Sätze schreiben oder ob Sie Use-Case-Diagramme malen.

Etwas genauer können wir die Abläufe auch mit deutschen Sätzen beschreiben. Wenn es sich um lineare Abläufe handelt, würden wir ganz einfach sagen. „Das System soll zuerst

Schritt 1 ausführen, dann Schritt 2, danach, unter der Bedingung x , den Schritt 3.“ Damit haben wir umgangssprachlich Aktivitäten beschrieben.

Die UML-Alternative sind Aktivitätsdiagramme. Wir zeichnen die Reihenfolge von solchen Schritten mit entsprechenden Abfragen oder mit ein bisschen Parallelität in Form eines Aktivitätsdiagramms. (Alternativ können Sie natürlich auch Flussdiagramme, BPMN, Petrinetze oder jegliche andere Art von Ablaufmodellen nutzen.)

Wenn die Abläufe nicht ganz so linear sind, sondern eher asynchron durch Ereignisse beeinflusst werden, dann würden wir in Deutsch formulieren: „Wenn das Ereignis eintritt, dann soll das System die oder jene Aktion ausführen und in den Zustand xxx gelangen.“ Die entsprechende grafische Darstellung sind Zustandsautomaten mit Zuständen und Übergängen. Sie sind beschriftet mit Ereignissen und Aktivitäten und Aktionen.

Ein Teil, den wir aus diesem Buch ausgeklammert haben, sind beispielhafte Abläufe. In den 1980er-Jahren war es unüblich, Beispiele zu verwenden. Wir strebten immer vollständige, konsistente und widerspruchsfreie Spezifikationen an. Mit dem Eintreffen der UML sind Beispiele wieder salonfähig geworden. Wir können ebenso beispielhafte Abläufe spezifizieren, auch wenn wir wissen, dass es nur ein Beispiel ist und keine vollständige Spezifikation, getreu dem Motto: Ein gutes Beispiel ist besser als eine schlechte Abstraktion.

Wir würden uns also einen typischen szenarienhaften Ablauf ansehen und ihn in Deutsch z. B. folgendermaßen beschreiben: „Das System soll ganz konkret diesen Schritt machen. Danach ganz konkret diesen Schritt. Danach diesen Schritt.“

Ein Szenario zeigt wenig Alternativen und Verzweigungen auf, sondern ist ein klipp und klarer Durchlauf durch das System. Auch dafür gibt es ein entsprechendes UML-Diagramm, nämlich Sequenzdiagramme, die bei Analytikern nur mittelmäßig beliebt sind; sie erfordern sehr viel Aufwand. Die vorher erwähnten Ausdrucksmittel (Use Cases und Aktivitätsdiagramme) sind wesentlich populärer, aber Szenarien sind ein sehr probates Mittel, um einen beispielhaften Ablauf auszudrücken. Grafisch darstellen würden Sie das als Sequenzdiagramm (oder auch als Kommunikationsdiagramm, als Message Sequence Charts und noch einige andere Diagrammartentypen).

Sie haben natürlich noch die dritte Alternative. Wenn Sie weder schreiben noch Bildchen zeichnen wollen, können Sie Abläufe einfach prototypen. Sie können einen Prototyp schreiben für einen typischen Geschäftsprozessablauf und diesen dem Kunden als Spezifikation oder als Ergänzung zu einer textuellen oder einer grafischen Spezifikation vorführen.

Wechseln wir in den Bereich der Daten. Auch hier kommen Sie mit Texten alleine zurecht. Ein anständig geschriebenes Glossar mit Begriffsdefinitionen für alle verwendeten Wörter in den Abläufen, alphabetisch geordnet, ist ein wunderbares Nachschlagewerk, um die Funktionen und Abläufe besser zu verstehen und jeden Begriff eindeutig definiert zu haben.

Wenn Sie allerdings sehr viele Begriffe haben, dann werden Sie unter Umständen diese vielen Begriffe in Form eines Entity-Relationship-Diagramms ordnen und viele Attribute zu einer Entity zusammenpacken. In der UML würden Sie zur Darstellung der Entities ein Klassendiagramm verwenden. Das sind nur zwei unterschiedliche Notationen für den gleichen Zweck, oft auch logisches Datenmodell genannt.

Auch für die Daten haben Sie die Möglichkeit, Prototypen zu entwickeln. In dem Fall wären es eher Prototypen für Oberflächen. Sie zeigen die Masken mit den entsprechenden Attributen, die bestimmte Werte annehmen, oder sie zeigen entsprechende Ausgabeformulare, entspre-

chende Reports, die das System generieren soll und zeigen damit beispielhaft oder ziemlich vollständig, welche Arten von Elementen Sie in den Masken oder in den Reports erwarten.

Nutzen Sie die jeweilige Stärke, nehmen Sie Grafiken vielleicht eher für den Überblick und Sprache eher für die Details und ergänzen Sie dies über entsprechende Prototypen.

Wenn Ihre Leser nicht gewillt sind, viel zu lesen oder Notationen von UML-Diagrammen zu lernen, sind Prototypen vielleicht das bessere Mittel, um über Anforderungen zu sprechen.

Jetzt bleibt uns noch die schwierige Gruppe der nichtfunktionalen Anforderungen, der Qualitätsanforderungen und der Randbedingungen. Diese werden meist in Textform geschrieben. Bei einer Qualitätsanforderung würden Sie zum Beispiel Sätze schreiben: „Das System soll folgende Qualitätseigenschaften aufweisen“ und Sie würden etwas über Verfügbarkeit oder Benutzerfreundlichkeit oder über Sicherheit oder rechtliche Anforderungen als deutschen Satz hinschreiben und entsprechend fordern.

Bei Randbedingungen würden Sie sagen: „Folgende Randbedingungen sind einzuhalten.“ Sie geben technologische Randbedingungen vor, ein bestimmtes Betriebssystem, das genutzt werden soll, eine Library, die eingesetzt werden muss, oder Sie geben zeitliche Randbedingungen vor, wie Maximalzeit für das Projekt, oder Sie geben Vorgaben über Einkaufen im Gegensatz zur Selbstfertigung (Make-or-Buy-Entscheidungen).

Für Qualitätsanforderungen und Randbedingungen verwenden wir keine grafischen Ausdrucksmittel, aber wir binden diese Sätze an die entsprechenden grafischen Ausdrucksmittel von Modellen. Sie würden zum Beispiel eine Performance-Anforderung an einen Use Case binden, um zu sagen, dass dieser Prozess nicht länger als drei Sekunden dauern soll. Oder Sie geben eine Häufigkeitsanforderung für ein Datenelement an und sagen: Von diesem Datenelement brauchen wir 12 000 Stück. Oder Sie binden eine Sicherheitsanforderung an eine ganz bestimmte Aktivität oder eine rechtliche Anforderung an eine andere Aktivität.

Wir schreiben also nichtfunktionale Anforderungen hauptsächlich in umgangssprachlichen Sätzen, verknüpfen sie aber mit entsprechenden Teilen des Modells. Aber auch in diesem Bereich der nichtfunktionalen Anforderungen haben wir noch die Möglichkeit des Prototyping und sollten sie auch sehr stark nutzen. Im Gegensatz zu den Oberflächenprototypen, die uns mehr Daten und Funktionen zeigen, sprechen wir hier von technischen Prototypen, von vertikalen Prototypen, von Durchstichen, um zum Beispiel die Performance bestimmter Technologien auszuprobieren. Wir testen, ob dieses eingekaufte Framework performant oder absturzsicher genug ist mit technischen Prototypen.

Sie haben also die Qual der Wahl beim Spezifizieren. Wollen wir schreiben, wollen wir malen oder wollen wir Prototypen? Und das gilt für alle Arten von Anforderungen, für die beiden Arten von funktionalen Anforderungen (für Abläufe und Daten) wie auch für die beiden nichtfunktionalen Anforderungen, für Qualitätsanforderungen und für Randbedingungen.