

„Schon das Wort Anforderung selbst kann falsch verstanden werden, sind die einzelnen Anforderungen doch meist eher flexibel handhabbar als erforderlich.“

*Eric Honour*

# 6

## Anforderung oder Anforderung – der feine Unterschied

---

### Fragen, die dieses Kapitel beantwortet:

- Was versteht man unter einer Anforderung?
- Welche Arten von Anforderungen lassen sich unterscheiden?
- Wozu dient eine Unterscheidung nach Art?
- Welche Detailstufen von Anforderungen gibt es?
- Warum sollte ich Anforderungen priorisieren?

# 6.1 Unterschiedliche Anforderungen

### Einleitung

Anforderung ist nicht gleich Anforderung. Dies wird klar, wenn man sich die lange Liste der Adjektive betrachtet, die dem Wort Anforderung in diesem und anderen Büchern zum Requirements-Engineering vorangestellt werden. Da gibt es dokumentierte und wirkliche Anforderungen, zentrale und nebensächliche, kundenspezifische und allgemeine, projekt- und firmenbezogene, und sogar romantische und klassische.<sup>1</sup>

Dieses Kapitel erklärt Ihnen den Begriff *Anforderung* selbst und erläutert drei wichtige Unterscheidungen:

- Art,
- Detailstufe und
- Priorität.

Die Charakterisierung der Anforderungen nach diesen Merkmalen hat sich in vielen unserer Projekte als hilfreich erwiesen. Die Merkmale können im Projekt als Checkliste dafür dienen, ob alle möglichen Anforderungen abgedeckt sind. Die Fragen hierfür lauten: „Welche Detailstufen sind notwendig, und habe ich jeweils alle Anforderungen der Detailstufe?“, „Welche Arten brauche ich, und habe ich jeweils alle Anforderungen der Art?“ und so weiter.

Jedem der Merkmale ist ein eigener Abschnitt in diesem Kapitel gewidmet.

## 6.1.1 Der Begriff der Anforderung

### IEEE

IEEE, das Institut der Elektrik- und Elektronik-Ingenieure, definiert Anforderung wie folgt (siehe [IEEE610]):

1. Beschaffenheit oder Fähigkeit, die von einem Benutzer zur Lösung eines Problems oder zur Erreichung eines Ziels benötigt wird.
2. Beschaffenheit oder Fähigkeit, die ein System oder System-Teile erfüllen oder besitzen muss, um einen Vertrag, eine Norm, eine Spezifikation oder andere, formell vorgegebene Dokumente zu erfüllen.
3. Eine dokumentierte Darstellung einer Bedingung oder Fähigkeit gemäß 1 oder 2.

Interessant ist darin insbesondere der dritte Aspekt, der zwischen Anforderung und dokumentierter Anforderung unterscheidet. Er wird uns in 6.4 beschäftigen.

### Definition der SOPHISTen

Wir finden folgende Definition sehr nützlich. Sie hat sich in der Praxis als gut verständlich, umfassend und ausreichend konkret erwiesen:

*Eine Anforderung ist eine Aussage über eine Eigenschaft oder Leistung eines Produktes, eines Prozesses oder der am Prozess beteiligten Personen.*

### Produkt versus System


Sagt jemand „Anforderung“, so meint er häufig nur einen Teil der Bedeutung, die mit unserer Definition abgedeckt ist. Er versteht den Begriff dann als Forderung nach einer Leistung, die „das System“ als ein Endergebnis einer Entwicklung erbringen

<sup>1</sup> Diese Begriffswahl geht auf ein Buch von Robert Pirsig zurück: „Zen und die Kunst, ein Motorrad zu warten“.

muss. Der Begriff „Produkt“ in der Definition meint jedoch mehr als nur „System“, umfasst Software und Hardware und zum Beispiel auch Abnahmekriterien, Handbücher, Protokolle, Planungsdokumente und so weiter. Dieser Ansatz folgt dem des V-Modells [VMM97].

Stakeholder wollen aber gerne sicher gehen, dass gewisse Regeln bei der Entwicklung eingehalten werden (Leistung oder Eigenschaft eines Prozesses), dass etwa ein definiertes Vorgehen verwendet wird. Oder dass die Systemingenieure eine bestimmte Qualifikation aufweisen (Eigenschaft einer am Prozess beteiligten Person). Auch diese Aspekte sind Anforderungen, die es im Rahmen der Entwicklung zu erfüllen gilt. Sie werden jedoch nicht immer in Anforderungsspezifikationen beschrieben. Kapitel 1 und Abschnitt 6.2 gehen auf diese Aspekte detaillierter ein.

Prozess, Person

 1 Qualität

## 6.1.2 Notwendige Unterscheidungen

Vermutlich gibt es mehr unterschiedliche Gruppierungen von Anforderungen, als es Autoren zu diesem Thema gibt. Ralph Young alleine nennt 21 verschiedene, auch ausdrücklich solche, die man nicht verwenden sollte (siehe [Young2004]). Auf eine Aussage über „die Anforderungen“ folgt also unweigerlich die Frage „Welche Anforderungen?“. Den maßgeblichen Literaturquellen und den verschiedenen Standards ist gemein, dass sie sich über verschiedene *Arten* und verschiedene *Ebenen* von Anforderungen äußern. Seit zunehmend inkrementell entwickelt wird, kommt noch eine weitere zentrale Gruppierung hinzu, die nach der Verbindlichkeit oder *Priorität*. Abbildung 6.1 zeigt die wichtigsten Gruppierungen und einige weitere im Überblick.


Zentrale Gruppierungen

Wichtige Merkmale	Siehe Abschnitt	Weitere Merkmale
Art	6.2	Stabilität
Ebene	6.3	Status
Priorität	6.4	Risiko
		Machbarkeit
		...

Abbildung 6.1: Gruppierungen für Anforderungen

Eine ganze Reihe weiterer Gruppierungen ist denkbar und sinnvoll (siehe vor allem Kapitel 13 „Ordnung im Chaos“). Sie sollten die Gruppen und Gruppierungen an den Bedürfnissen Ihrer Organisation oder Ihres Projektes orientieren. Die hier vorgestellten Gruppierungen können Ihnen als Ausgangspunkt dienen. Es ist durchaus wahrscheinlich, dass sich Ihre Gruppierungen mit der Zeit ändern. Achten Sie immer darauf, dass jede Gruppierung anhand von Merkmalen erfolgt, die im Projektteam zusammen ausgearbeitet und genau definiert wurden. Sonst interpretiert jeder Mitarbeiter die Gruppen anders, und die Anforderungen werden nicht gleichartig zugeordnet.

Ausgangspunkt

 13 RM

Die nachfolgenden drei Abschnitte dieses Kapitels gehen jeweils auf eine der Gruppierungen genauer ein und begründen, weshalb sie so wichtig sind.

# 6.2 Gruppierung nach Art

## 6.2.1 Arten von Anforderungen

Die Gruppen innerhalb der Gruppierung nach Art können fast beliebig gewählt werden. Wir geben die folgenden vor und denken, damit alle Anforderungen einsortieren zu können:

- funktionale Anforderungen,
- technische Anforderungen,
- Anforderungen an die zu verarbeitenden Informationen,
- Anforderungen an die Benutzerschnittstelle,
- Qualitätsanforderungen,
- Anforderungen an sonstige Lieferbestandteile,
- Anforderungen an die Durchführung der Entwicklung und
- rechtlich-vertragliche Anforderungen.

Um zu erkennen, was damit jeweils genau gemeint ist, können Sie im Abschnitt 6.5 auf ausführliche Beispiele zurückgreifen. Zunächst reicht ein intuitives Verständnis aus.

Nicht-funktionale Anforderungen

Der oft verwendete Begriff *nicht-funktionale* Anforderungen spiegelt sich in obiger Aufstellung auch wider: Jede Anforderung, die keine funktionale Anforderung ist, ist nicht-funktional. Kapitel 10 „Die nicht-funktionalen Anforderungen“ behandelt diese Gruppe ausführlich.

## 6.2.2 Merkmale der einzelnen Gruppen

Funktionale Anforderungen

*Funktionale Anforderungen* beschreiben Aktionen, die von einem System selbstständig ausgeführt werden sollen, Interaktionen des Systems mit menschlichen Nutzern oder Systemen (Eingaben, Ausgaben) und Anforderungen zu allgemeinen, funktionalen Vereinbarungen und Einschränkungen. Manchmal werden sie Verhaltensanforderungen (*behavioral requirements*, [Davis93]) genannt, weswegen sie oft in verschiedenen Arten von Verhaltensdiagrammen dargestellt werden. Kapitel 7 „Dokumentation von Anforderungen“ sagt mehr zu den Dokumentationstechniken.

→ 7 Doku

Technische Anforderungen

Bei der Gruppe der *technischen Anforderungen* geht es um technische Forderungen, im Gegensatz zu den eher fachlich motivierten funktionalen Anforderungen. Dazu zählen Hardwareanforderungen, Architekturanforderungen, Anforderungen an die Programmiersprachen, ... Diese Anforderungen werden oft notwendig, weil ein System in ein bestehendes technisches Umfeld eingebettet werden muss oder Verträge die Nutzung einer bestimmten Infrastruktur vorschreiben.

Anforderungen an die zu verarbeitenden Informationen

*Anforderungen an die zu verarbeitenden Informationen* werden typischerweise in Glossaren, Datenverzeichnissen (*repositories*) und Domänenbeschreibungen dargelegt sowie in Entitäts- oder Klassenmodellen und anderen Arten von Strukturdiagrammen. Diese Anforderungen behandeln die Begriffswelt des Problem- oder Anwendungsbereichs und stellen gewissermaßen den Wortschatz dar, aus dem sich die fachlich motivierten Anforderungen bedienen, also vor allem die der höheren Ebenen

(siehe Abschnitt 6.3). Der Ontologiebildung ist im Requirements-Engineering ein eigener Forschungszweig gewidmet, siehe dazu die grundlegenden Arbeiten von Julio Leite [Leite93]. Als Anforderungen, die besonders auf Schnittstellen zwischen Systemen oder Systemteilen eingehen, werden die zu verarbeitenden Informationen oft auch sehr technisch beschrieben; verbunden mit Datentypen etwa, und auch Wertebereichen und Regeln für die Handhabung.

Unter *Anforderungen an die Benutzerschnittstelle* ist einerseits zu verstehen, was üblicherweise unter Mensch-Maschine-Schnittstelle eingeordnet wird. Also Form und Funktion von Ein- und Ausgabe-Geräten, die dem menschlichen Benutzer die Interaktion mit dem System ermöglichen. Andererseits können auch angrenzende Systeme über eine Benutzerschnittstelle mit dem zu entwickelnden System verbunden sein, wenn der Zweck diese zu Benutzern macht. Das ist zum Beispiel bei reinen Infrastruktur- oder Middleware-Systemen der Fall, bei denen das System ausschließlich anderen Systemen eine Dienstleistung anbietet.

Mit *Qualitätsanforderungen* sind Anforderungen gemeint, die Angaben über die Güte des Produktes, des Prozesses oder der am Prozess beteiligten Personen machen. Manchmal werden sie auch Dienstgüte-Anforderungen oder Dienstqualitäts-Anforderungen (*quality of service requirements*) genannt, wobei diese sich in der Regel nur auf Systeme oder eventuell noch allgemeiner auf Produkte beziehen, nicht jedoch auf Prozesse oder Personen. Die DIN EN ISO 66272 [DIN94] unterteilt die Dienstgüte in die fünf Merkmale Zuverlässigkeit, Benutzbarkeit, Effizienz, Änderbarkeit und Übertragbarkeit (siehe Abbildung 6.2). Sie beschreibt alle möglichen Aspekte der Dienstqualität von Systemen praxistauglich und ordnet sie überschneidungsfrei in einem hierarchischen Schema an. Diese Art von Anforderungen ist ausführlich im Kapitel 10 „Die nicht-funktionalen Anforderungen in der Systementwicklung“ dargestellt.

Anforderungen  
an die Benutzer-  
schnittstelle

Qualitäts-  
anforderungen

10 nFA

## Dienstqualität

- > Funktionalität
  - Richtigkeit
  - Angemessenheit
  - Interoperabilität
  - Ordnungsmäßigkeit
  - Sicherheit
- > Zuverlässigkeit
  - Systemreife
  - Systemwiederherstellbarkeit
  - Systemfehlertoleranz
- > Benutzbarkeit
  - Systemverständlichkeit
  - Systemlernbarkeit
  - Systembedienbarkeit
- > Effizienz
  - Systemzeitverhalten
  - Systemverbrauchsverhalten
- > Änderbarkeit
  - Systemanalysierbarkeit
  - Systemmodifizierbarkeit
  - Systemstabilität
  - Systemprüfbarkeit
- > Übertragbarkeit
  - Systemanpassbarkeit
  - Systeminstallierbarkeit
  - Systemkonformität
  - Systemtauschbarkeit

Legende:

- > Qualitätsmerkmale
- Submerkmale

Abbildung 6.2: Unterteilung der Anforderungen an die Dienstqualität nach DIN 66272

## 6 Anforderung oder Anforderung

Anforderungen an sonstige Lieferbestandteile

Selten besteht das System nur aus einem Stück kompilierten Programmcodes. Eine Reihe von Dingen, die auf den ersten Blick gar nicht nach Systembestandteil aussehen, kann der Stakeholder zusätzlich vom Entwickler fordern. Man nennt sie *Anforderungen an sonstige Lieferbestandteile*. Dabei können zum Beispiel nicht nur die Existenz eines Installationshandbuches gefordert sein, sondern auch Anforderungen an das Handbuch selbst gestellt werden.

Anforderungen an die Durchführung der Entwicklung und Einführung

Manchmal möchten Stakeholder darauf Einfluss nehmen, in welcher Art und Weise das System entwickelt oder eingeführt werden soll. Zu nennen sind hier unter anderem Anforderungen an die Vorgehensweise (Software-Erstellung, Software-Prüfung), anzuwendende Standards, Hilfsmittel (Tools), die Durchführung von Besprechungen, von Abnahmetests (fachliche Abnahme, betriebliche Abnahme) und die Festlegung von Terminen. Diese Art nennt man *Anforderungen an die Durchführung der Entwicklung und Einführung*. Eine andere Bezeichnung ist Projektanforderung (*project requirement*, [IEEE830]). Diese Gruppe von Anforderungen wird meist nicht in einer Anforderungsspezifikation beschrieben, sondern eher im Projekthandbuch oder in Vereinbarungen über die Zusammenarbeit von Auftraggeber und Auftragnehmer.

Rechtlich-vertragliche Anforderungen

Unter *rechtlich-vertraglichen Anforderungen* sind Angaben zu Zahlungsmeilensteinen sowie zu Vertragsstrafen, dem Umgang mit Änderungen der Anforderungen, Eskalationspfaden und so weiter zu verstehen. Wie der Name schon sagt, stehen diese Anforderungen meist in den Verträgen zwischen Auftragnehmer und Auftraggeber.

### 6.2.3 Gründe für die Gruppierung nach Art

Es spricht nichts dagegen, eine Anforderung gleichzeitig mehreren Gruppen zuzuordnen. Vor allem recht abstrakte Anforderungen (siehe Abschnitt 6.3) sind nicht zweifelsfrei gruppierbar. In den abgeleiteten feineren Anforderungen kann die Zuordnung dann präzisiert werden. Prinzipiell sollten Sie Ihre Anforderungen jedoch möglichst eindeutig gruppieren. Denn eine Gruppierung nach Art

- erhöht die Lesbarkeit von Anforderungsspezifikationen für jede Zielgruppe,
- ermöglicht geeignete Entwurfsentscheidungen,
- fördert die Wiederverwendung und
- erleichtert das Analysieren von Anforderungen.

Lesbarkeit

Neben der Unterscheidung nach Detailebenen (vergleiche Abschnitt 6.3) fördert die Klassifikation nach Art die Lesbarkeit von Anforderungsspezifikationen am effektivsten. Die Ursache ist einleuchtend: Wird das Requirements-Management auf angemessene Weise umgesetzt, ist es leicht möglich, die Anforderungen nach diesen Arten zu gruppieren. So kann jede Lesergruppe die Arten von Anforderungen, die für sie von Interesse sind, unmittelbar herausfiltern und den Rest ausblenden.

Entwurfsentscheidungen




Besonders die technischen Anforderungen und die Qualitätsanforderungen beeinflussen die Lösung maßgeblich, wie man sich anhand der Aspekte Änderbarkeit und Effizienz leicht vor Augen führen kann. Deshalb ist es unabdingbar, dass der Auftragnehmer früh im Projekt die Antworten auf Fragen zur Dienstqualität bekommt, um einen geeigneten Entwurf entwickeln zu können. Kaum etwas bringt eine Architektur leichter zum Einsturz als derartige Anforderungen, die zu spät bekannt werden.

Sieht man sich die acht genannten Arten an, so stellt sich heraus, dass sich vor allem die funktionalen Anforderungen von Projekt zu Projekt inhaltlich unterscheiden und auch die Anforderungen an die zu verarbeitenden Informationen. Die meisten Anforderungen der anderen Arten bleiben hingegen weitgehend gleich, zumindest ihrer Form nach. Bei Standardprodukten, die nur noch an die Wünsche des Auftraggebers angepasst werden, oder bei Produktfamilien sind nicht einmal die funktionalen Anteile und die verarbeiteten Informationen recht variabel. Das legt es nahe, auch Anforderungen wiederzuverwenden, nicht nur Lösungen für Anforderungen. Die Kapitel 9 „Anforderungsschablone“, 10 „Die nicht-funktionalen Anforderungen in der Systementwicklung“ und 15 „Und jetzt? – Strategien zur Einführung“ gehen darauf näher ein.


Wenn Sie Anforderungen nach ihrer Art klassifizieren, werden sie auch einfacher zu ermitteln, zu formulieren und teilweise sogar zu validieren sein. Die Bearbeiter können sich jeweils auf einen Aspekt konzentrieren. Die Anforderungsermittlung gestaltet sich effektiver, weil der Analytiker einer bestimmten Zielgruppe konkrete Fragen aus ihrer Perspektive stellen kann. Die Formulierung der Anforderungen fällt leichter, weil die Zusammenhänge auf die jeweilige Art beschränkt werden können. Sätze, die mehrere Arten von Anforderungen enthalten, können vermieden werden. Dadurch werden Anforderungen wiederum klarer. In aller Regel werden die Anforderungsspezifikationen auch kleiner, da nicht so viel Kontext beschrieben werden muss. Schließlich sind Anforderungen auch leichter zu validieren, weil der Fokus hier ebenfalls auf eine Art begrenzt bleiben kann.

Einzigster Nachteil der Klassifizierung nach Art: Zusammenhänge zwischen verschiedenen Arten von Anforderungen sind schwerer zu erkennen, wenn sie beispielsweise an ganz verschiedenen Stellen in der Anforderungsspezifikation beschrieben sind. Geeignete Maßnahmen zum Requirements-Management wirken dem jedoch entgegen (siehe Kapitel 13 „Ordnung im Chaos“).

Wieder-  
verwendung

-  9 Schablone
-  10 nfA
-  15 Einführung

Leichteres  
Analysieren

 13 RM

## 6.3 Gruppierung nach Detailebene

In der Praxis trifft man auf unterschiedliche Ansätze, Anforderungen auf mehreren Ebenen zu schreiben. Weil die Begriffe aus dem Bereich der Softwareentwicklung so unterschiedlich aufgefasst werden, ist mittlerweile ein schon sagenhaftes Dickicht entstanden, das kaum mehr zu durchdringen ist.

Uns geht es an dieser Stelle darum zu erklären, auf welchen Kern sich all die verschiedenen Modelle zurückführen lassen. Wir möchten ausdrücklich kein weiteres Modell hinzufügen! Welches Sie tatsächlich verwenden, bleibt natürlich Ihnen überlassen. Unterscheiden Sie nur die Ebenen innerhalb der verwendeten Terminologie.

Im folgenden Erklärungsmodell wird jede einzelne Anforderung einer von fünf Detailstufen zugeordnet, die von 0 bis 4 durchnummeriert sind. Dabei gehören die größten Anforderungen (Ziele, grobe Systembeschreibungen, ...) der Ebene 0 an, die feinsten (Feinspezifikation, Modulanforderung, ...) der Ebene 4. Konkrete Namen werden in der folgenden Aufstellung bewusst vermieden, weil dann Begriffe nicht so leicht intuitiv anders als beabsichtigt aufgefasst werden. Dass unser Erklärungsmodell nun gerade fünf Ebenen hat, ist einigermaßen willkürlich; in der Praxis gibt es auch Projekte mit sieben und mehr Ebenen, solche Herausforderungen sind jedoch vergleichsweise selten.

Erklärungsmodell

### 6.3.1 Gründe für die Gruppierung nach Detailebenen

Warum ist es sinnvoll, Anforderungen auf verschiedenen Detailebenen zu beschreiben? Es wird dadurch möglich, zielgruppenspezifisch zu schreiben. Die Dokumente werden leichter verständlich, da man geeignet abstrakt formulieren kann, und es kann zwischen verschiedenen Projektbeteiligten leichter ein gemeinsames Verständnis erzielt werden. Gleichzeitig verringert man den Aufwand, der für das Validieren der Anforderungen notwendig ist.

Vorteile und Gefahren

Geben die Stakeholder eine grobe Systembeschreibung, so ist den Entwicklern zunächst ungefähr klar, worum es gehen soll. Die schriftlich fixierten Ziele eines Projektes als Anforderungen mit hohem Abstraktionsgrad ermöglichen den Managern messbare Rückschlüsse, ob das Projekt erfolgreich ist oder war. Beschriebene Anwendungsfälle – schon etwas detaillierter – umreißen für Projektleiter sehr viel klarer, was zum System gehören soll und was nicht. Architekten können mit fachlich motivierten Nutzerforderungen – noch genauer und weniger abstrakt – Grobentwürfe erstellen. Datenbankdesigner finden die für ihre Arbeit notwendigen Details in Feinspezifikationen.

Durchmischen Sie die verschiedenen Ebenen besser nicht, denn sonst laufen Sie Gefahr, etwas wegzulassen. Zum anderen müssten sich Ihre Leser der im Extremfall einzigen undifferenzierten Anforderungsspezifikation durch Unmengen von Informationen kämpfen, die für sie vielleicht momentan nicht relevant sind.

Durch feinere Detailstufen wird eine Anforderungsspezifikation als Ganzes zwar schwerer lesbar, einzelne Anforderungen aber verständlicher. Feine oder grobe Anforderungen sollten nicht isoliert gesehen werden, sondern jeweils im Zusammenhang mit der Lesergruppe und deren momentanem Interesse. Grobe Anforderungen stecken für den Leser den Rahmen ab, feine spezifizieren Details.

Fein versus präzise

Im Allgemeinen kann man sagen, dass Anforderungen einer höheren Ebene, das heißt einer Ebene mit kleiner Zahl, interpretierbarer sind als Anforderungen einer niedrigeren Ebene (mit einer hohen Zahl). Je niedriger die Ebene, desto weniger Spielraum gibt es, eine Anforderung unterschiedlich auszulegen (Abschnitt 6.5 gibt Beispiele hierfür an). Deshalb ist es gängige Praxis, dass der Auftragnehmer die Anforderungen des Auftraggebers feiner spezifiziert, um zu einem gemeinsamen Verständnis zu kommen, und sie dann vom Auftraggeber abnehmen lässt. Man beachte abschließend, dass mit „feiner“ nicht „präziser formuliert“ gemeint ist. Anforderungen können auf jeder Ebene mehr oder weniger präzise formuliert werden.

### 6.3.2 Die Detailebenen 0 bis 4

Logisch betrachtet lassen sich Anforderungen verschiedenen Ebenen zuordnen. Die Anforderungen einer Ebene ergeben sich durch die Verfeinerung von einer oder mehreren Anforderungen auf einer höheren Ebene. Häufigster Fall ist dabei eine abstrakte Anforderung, aus der sich durch die Wahl einer Lösung mehrere konkretere ableiten lassen ([Davis93]). Daraus ergibt sich eine Hierarchie von Anforderungen, die Abbildung 6.3 darstellt.



Der linken Spalte ist die generische Struktur zu entnehmen, die die Ziffern 0 bis 4 verwendet. In der rechten Spalte werden Begriffe aus der Literatur genannt, die häufig in einem gleichen oder zumindest ähnlichen Zusammenhang gesehen werden. Das durch die verschiedenen Ebenen suggerierte sequenzielle Vorgehen vom Groben zum Feinen ist nicht so praxisnah, wie viele Lehrbücher oder Vorgehensmodelle uns glauben machen wollen. Im Projekt werden selten wirklich 100%ig sauber schrittweise Verfeinerungen an Anforderungen vorgenommen. Das Verfahren orientiert sich eher am Grundsatz middle-out als am Top-down-Prinzip. Üblicherweise bestehen von Anfang an Anforderungen auf allen fünf Levels. Beim Verfeinern einer Anforderung ergeben sich meist auch Anforderungen anderer Arten (siehe 6.2).

Es gehört zu den wohl schwierigsten Aufgaben der Anforderungsanalyse, Anforderungen der richtigen Ebene zuzuordnen und Lücken aufzufüllen, die dadurch in anderen Ebenen entstehen. Nur wenn Sie die Anforderungen sehr genau verstehen, können Sie ihre hierarchischen Zusammenhänge erkennen und gegebenenfalls weitere Anforderungen ergänzen. Gut, dass dieser Schritt nur notwendig ist, wenn die Anforderungen lange leben oder von vielen unterschiedlichen Zielgruppen gelesen werden. Geeignete Maßnahmen des Requirements-Managements erleichtern dem Analytiker die Arbeit (siehe Kapitel 13 „Ordnung im Chaos“).

Unsere Terminologie	Andere Terminologien
Spezifikationslevel 0	Grobe Systembeschreibung, Systemziele, Systemüberblick, Vision, introduction, mission statement, business goals, business requirements
Spezifikationslevel 1	Grobe Systembeschreibung, Anwendungsfall (Use Case), Geschäftsprozess, user story, Geschäftsvorfall, (Anwendungs-) Szenario, Funktionsbeschreibung, Funktionsgliederung, fachliche Anforderung, organizational requirement, Abnahmekriterien, Featureliste, Kontextabgrenzung
Spezifikationslevel 2	Anwenderforderung, Nutzeranforderung, Operational Concept Description, Interface Requirements Specification, Lastenheft, Sollkonzept, Grobspezifikation, operational requirement, betriebliche Anforderung, Fachkonzept, Abnahmekriterien, Featureliste
Spezifikationslevel 3	Technische Anforderung, Schnittstellenübersicht, Schnittstellenbeschreibung, System Segment Specification, Interface Requirements Specification, Systemanforderung, Pflichtenheft, Lastenheft, Feinspezifikation, Abnahmekriterien, Featureliste
Spezifikationslevel 4	Technische Anforderung, Schnittstellenübersicht, Schnittstellenbeschreibung, System Requirement Specification, Interface Design Description, Software- + Hardwareanforderungen, Pflichtenheft, Feinspezifikation, Modulanforderung, Testfälle, Abnahmekriterien

**Abbildung 6.3:** Vergleich unterschiedlicher Terminologien

Interessant ist nun die Frage, zu welcher Ebene eine Anforderung gehört, oder, andersherum, was die Ebenen voneinander unterscheidet. Im Folgenden finden Sie deshalb einen kurzen Überblick und einige Hinweise, was für die einzelnen Ebenen von Anforderungen spezifisch ist. Gemeinsam ist allen Ebenen, dass sie ganz und gar analytischer Natur sind. Beispiele finden Sie in Abschnitt 6.5.

## 6 Anforderung oder Anforderung


Spezifikationslevel 0	Eine Anforderung auf <i>Spezifikationslevel 0</i> umreißt in wenigen Sätzen (üblicherweise ein bis drei Seiten) den Gesamtumfang des zu entwickelnden Systems. Sie dient dazu, einen ersten Überblick zu vermitteln. Sie stellt eher einen Fließtext zur Erläuterung des Systemumfangs als eine Spezifikation im Sinne der Informatik dar. Häufig enthält sie auch Absichtserklärungen, Ziele des Systems, eine Liste mit Hauptmerkmalen, etcetera.
Spezifikationslevel 1	Anforderungen des <i>Spezifikationslevels 1</i> beschreiben die Teilvorgänge des Systems genauer als in der groben Systembeschreibung. Gibt es kein Spezifikationslevel 0, so beginnt man hier üblicherweise mit überschaubaren 50–200 Anforderungen. Die Unterteilung in Teilvorgänge kann mit Use Cases geschehen oder sich an einer funktionalen Sichtweise orientieren. Manchmal werden hier schon die Anwenderforderungen angesiedelt (siehe Abbildung 6.8). Weil es aber doch einen großen qualitativen Unterschied zwischen Use Case und Anwenderforderung im Sinne einer Anforderung „Das System muss ...“ gibt, sollten die Forderungen der Nutzer auf Level 2 stehen.
Spezifikationslevel 2	Anforderungen des <i>Spezifikationslevels 2</i> sind alle fachlich motivierten Anforderungen. Häufig sind es diejenigen, die vom Anwender vorgegeben werden. Hierbei werden die Anforderungen meist ohne Differenzierung nach möglichen Teilsystemen formuliert. Auf dieser Basis entsteht meistens die Architektur.
Spezifikationslevel 3	Oft detaillieren mehrere Level-3-Anforderungen eine Level-2-Anforderung. Von einer Anforderung des Spezifikationslevels 2 gelangt man zu Anforderungen des <i>Spezifikationslevels 3</i> , indem man sich die Frage stellt „Wer macht was?“. Häufig kann man dann zwischen Aktionen, die der Anwender ausführen können soll, um eine Systemfunktion zu veranlassen, und der eigentlichen Systemfunktion unterscheiden. Ebenso häufig werden Anforderungen in dieser Detailebene verschiedenen Teilsystemen zugeordnet, um beispielsweise jedes Teilsystem mit einem gesonderten Vertrag zur Implementierung abzudecken.
Spezifikationslevel 4	Die Anforderungen des <i>Spezifikationslevels 4</i> geben schließlich die Auflösung des Gesamtsystems in Soft- und Hardware oder weitere Module, Konfigurationselemente und so weiter wieder. Hierbei werden die bisherigen Anforderungen durch weitere Anforderungen (meist technische Anforderungen) detailliert oder ergänzt.  Der Übergang zu ihnen kann anhand folgender Überlegung nachvollzogen werden: Auf der Ebene der Anforderungen des Spezifikationslevels 3 ist es theoretisch noch möglich, das System durch einen menschlichen Bearbeiter zu realisieren, den die Anwender „benutzen“. Man könnte so ein System als Papiersystem bezeichnen, weil der bewusste Bearbeiter vermutlich Unmengen von Papier braucht, um das System zu „sein“. Durch die Anforderungen des Spezifikationslevels 4 ist der Lösungsraum in einer Art und Weise eingeschränkt, dass ein Papiersystem im Sinne der Anforderungen keine gültige Lösung wäre.

## 6.4 Gruppierung nach Priorität

Wussten Sie, dass rund die Hälfte der von Systemen geforderten Eigenschaften später kein einziges Mal verwendet wird? Zu diesem Ergebnis kommt die angesehene Standish Group in ihrem neuesten CHAOS-Report [CHAOS2003]. Das lässt den Schluss zu, dass viele Features zwar aufwändig, aber umsonst analysiert und entwickelt werden. Dieser Aufwand ließe sich vermutlich besser investieren.

### 6.4.1 Allgemeines

Der Schlüssel zu zielgerichteten Entwicklungen heißt Priorisierung. Mit Priorität bezeichnet man allgemein jede Art von Stellenwert innerhalb einer Rangfolge. Bei Anforderungen kann so eine Rangfolge zum Beispiel nach Wichtigkeit oder Dringlichkeit oder nach beidem gebildet werden. Die Priorität zählt zu den typischen Attributen einer Anforderung, die in Werkzeugen fürs Requirements-Management abgebildet werden. Mehr dazu in Kapitel 13 „Ordnung im Chaos“.

 13 RM

Priorisierte Anforderungen und auch das Priorisieren als Vorgang selbst unterstützen mehrere Vorgänge im Requirements-Engineering:

Zweck

- Zeit und Kosten treffender schätzen
- Geeignete Kompromisse finden (Zeit- und Kostenplanung, Qualitätsanforderungen)
- Implizite Annahmen klären
- Konflikte konstruktiv lösen (verschiedene Sichtweisen und Interessen unter den Stakeholdern)
- Stakeholder mit einer gemeinsamen Vision versehen
- Die wirklichen<sup>2</sup> Anforderungen ermitteln und aushandeln
- Kräfte effizient einsetzen (Einsatzplanung, Qualitätssicherung)
- Inkremente planen, die den jeweils maximalen Nutzen aufweisen
- Verbesserungen zielgerichtet entwickeln

In den meisten Projekten reicht es schon, Anforderungen nach einem groben Schema mit drei oder vier Stufen zu bewerten, etwa *notwendig*, *gewünscht* und *optional*, wie das Karl Wiegers vorschlägt ([Wiegers99]). Abstufungen wie *hoch*, *mittel*, *niedrig* sind ungeeignet, weil diese Bezeichner von den Stakeholdern oft unterschiedlich interpretiert werden. Definieren Sie deshalb die Bedeutung des Schemas genau und führen Sie eher ein weiteres Schema ein, als alle Aspekte durch ein einziges auszudrücken. Auf unserer Webseite finden Sie ein zusammengesetztes Bewertungsschema, das ebenfalls auf Karl Wiegers zurückgeht.

Einfache  
Lösungen

Die Anforderungen können Sie in Workshops priorisieren, entweder in einer Diskussion oder indem Sie Punkte vergeben: Jeder Teilnehmer hat eine vorab bestimmte Anzahl von Punkten, die er frei auf die aufgelisteten Anforderungen verteilen kann. Nachdem alle ihre Punkte verteilt haben, ordnet man die Anforderungen nach der

<sup>2</sup> Im Gegensatz zu denen, die zunächst einmal formuliert wurden ([Young2001]).

Anzahl der Punkte, teilt die Wertemenge in drei oder vier etwa gleich große Untermengen und erhält so die verschiedenen Prioritätsgruppen. Dieses Verfahren können Sie zum Beispiel leicht an einer Präsentationswand und mit bunten Klebepunkten umsetzen. Achten Sie darauf, dass jeweils nur maximal fünfzehn Anforderungen gleichzeitig zu bewerten sind, andernfalls verlieren die Abstimmenden den Überblick. Daran zeigt sich auch, dass dieses Verfahren besonders für Anforderungen höherer Ebenen (siehe Abschnitt 6.3) geeignet ist.

### Kano-Modell

Ein ausgefeilteres Modell hat der Japaner Noriaki Kano Ende der 70er Jahre entwickelt. Es passt vor allem für Auftragnehmer in IT-Projekten, da es die Gruppe von Anforderungen berücksichtigt, derer sich der Auftragnehmer nicht oder noch nicht bewusst ist. Kano hat vorgeschlagen, Anforderungen in drei Gruppen einzuteilen:

- Basisfaktoren,
- Leistungsfaktoren und
- Begeisterungsfaktoren.

### 3 Idee-System

Kapitel 3 „Von der Idee zum System“ geht ausführlicher auf dieses Modell ein.

### Weiterführende Ansätze

Verschiedene Autoren haben sich eine Reihe von mehr oder weniger detaillierten Verfahren ausgedacht, die zur Erreichung der oben genannten Ziele dienen. Wir empfehlen Ihnen unter anderem die Arbeiten von Barry Boehm (WinWin-Modell [Boehm95], COCOMO [Boehm2000]) und von Karlsson und Ryan [Karlsson97].

Eine ganze Reihe von Kriterien sind denkbar, anhand derer sich Prioritäten vergeben lassen:

- Ist es besonders kritisch, wenn die Leistung oder Eigenschaft nicht gegeben ist?
- Sind vielleicht sogar Menschenleben in Gefahr oder droht ein hoher finanzieller oder Image-Schaden?
- Wird die Leistung oder die Eigenschaft häufig abverlangt werden?
- Spielt die Leistung oder Eigenschaft bei einem der Kerngeschäftsprozesse eine Rolle?
- Muss die Forderung aus rechtlichen Gründen zwingend erfüllt werden, weil zum Beispiel personenbezogene Daten betroffen sind?

### Prioritäten vererben

Prioritäten können vererbt werden, von Anforderungen eines groben Spezifikationslevels auf Anforderungen niedrigerer Ebene. Bereits Ziele als die früheste oder abstrakteste Form von Anforderungen können bewertet werden. Sie sollten zusätzlich Vererbungsregeln festlegen, nach denen sich dann Prioritäten abstrakter Anforderungen auf konkrete übertragen lassen. Hat eine abstrakte Anforderung eine hohe Priorität, so kann man festlegen, dass mindestens eine (oder alle) der von ihr abgeleiteten konkreteren Anforderungen diese Priorität aufweisen müssen. Dabei ist jedoch Vorsicht geboten: wenn Kosten und Zeitplan geschätzt werden, indem man Anforderungen bestimmter Prioritätsgruppen zählt, so kann die Versuchung groß sein, aus einer Anforderung fünf andere abzuleiten, wo es drei auch getan hätten.

## 6.4.2 Kritikalität

Manchmal wird anstelle von *Priorität* der Begriff *Kritikalität* verwendet. Unter *Kritikalität* ist in diesem Zusammenhang eine Rangordnung zu verstehen, nach der bewertet werden kann, welche Auswirkungen es hat, wenn eine Anforderung nicht erfüllt wird. Dabei fließen oft der erwartete Schaden und die Eintrittswahrscheinlichkeit des Schadens mit in die Bewertung ein. Der Auftraggeber sollte sich bewusst machen, welche Anforderungen besonders kritisch sind. Dies dem Auftragnehmer zu überlassen, ist gefährlich, im schlimmsten Fall fatal.

Sinnvoll sind drei- oder vierstufige Rangordnungen, entweder mit sprechenden Namen versehen („sehr hoch“, „hoch“, ...) oder einfach als Ziffern von 1 bis 4 oder Buchstaben von A bis D dargestellt. Hier als Beispiel der Vorschlag von Alistair Cockburn (spricht man „Co-burn“, [Cockburn2002]):

Drei oder vier  
Stufen

Verlust von Leben
Verlust von (über-)lebenswichtigen Vermögenswerten
Verlust von Vermögenswerten, der jedoch die Handlungsfreiheit nicht einschränkt
Verlust von Behaglichkeit

Abbildung 6.4: Kritikalität nach Cockburn

## 6.4.3 Rechtliche Verbindlichkeit

Die rechtliche *Verbindlichkeit* beschreibt den Grad der Verpflichtung, den der Stakeholder den Angaben in der Anforderungsspezifikation beimisst. Besonders im Zusammenhang mit Verträgen wird dieser Begriff oft verwendet. Kennzeichnen Sie die Anforderungen mit einem Grad an Verbindlichkeit, wenn Sie unterschiedlich strikte Vorgaben ausdrücken möchten. Abbildung 6.5 zeigt mögliche Verbindlichkeitsgrade und die zugehörigen Schlüsselwörter, die man in den Anforderungen selbst verwenden kann. Der Zusammenhang zu den in Abschnitt 6.4.1 genannten Stufen *notwendig*, *gewünscht*, *optional* ... ist offensichtlich.

Verbindlichkeit	Deutsches Schlüsselwort	Englisches Schlüsselwort
Pflicht	muss (soll)	shall
Wunsch	soll (sollte)	should
Vorschlag	kann	may
Absicht	wird	will <sup>3</sup>
Kommentar	-	-

Abbildung 6.5: Per Schlüsselwort lässt sich die Verbindlichkeit bestimmen

<sup>3</sup> Vorsicht: Unter Umständen kann das Englische *will* auch intuitiv als Kennzeichen für eine unabdingbare Forderung verstanden werden. Gegebenenfalls sollte die Bedeutung einfach im Projekt definiert werden.

- Eindeutige Kennzeichnung** Im Grunde ist es aber egal, welche Schlüsselworte verwendet werden, und sogar, ob überhaupt durchgängig Schlüsselworte benutzt werden. Entscheidend ist, dass man Anforderungen eindeutig kennzeichnet. Werden die Anforderungen in Datenbanken gespeichert, so kann zum Beispiel jeder Anforderung das Attribut „Verbindlichkeit“ zugeordnet werden, für das vordefinierte Werte zugelassen sind. Natürlich sind Anforderungsspezifikationen aber viel leichter zu lesen und ihre Verbindlichkeit ist einfacher zu erfassen, wenn diese direkt aus dem Satz hervorgeht.
- Kommentar** *Kommentare* schreibt der Stakeholder, um Hintergründe zu erläutern. Sie können auf unterschiedliche Art und Weise dokumentiert werden, zum Beispiel um den Zweck einer vorangehenden Anforderung zu beschreiben, um einen Überblick zu geben, um Inhalte grafisch darzustellen oder tabellarisch zusammenzufassen. Kommentare machen umfangreiche Anforderungsspezifikationen meist erst verständlich. In Kommentaren sollten die anderen Schlüsselworte dann vermieden werden.

## 6.5 Beispiele

Die folgende Abbildung zeigt verschiedene natürlichsprachliche formulierte Beispiele zu den Gruppen Art und Ebene. Beispiele für die Gruppe *Priorität* ergeben sich, indem Sie in jedem der aufgeführten Anforderungen die Schlüsselworte durch andere ersetzen. Siehe auch Kapitel 7 „Dokumentation von Anforderungen“, dort sind weitere Beispiele in anderer Darstellung aufgeführt.

 7 Doku

Gruppe	Beispiele
<b>Art</b>	
Funktionale Anforderungen	Das System muss Ausleihgegenstände in den Bestand aufnehmen können.
	Das System muss es dem Benutzer bei der Aufnahme eines Ausleihgegenstandes in den Bestand ermöglichen, den Autor, den Titel und die ISBN einzugeben.
	Das System muss sicherstellen, dass jeder Ausleihgegenstand nur genau einmal im Bestand vorkommt.
	Möchte ein Ausleiher einen Ausleihgegenstand ausleihen, obwohl er bereits welche ausgeliehen hat und deren Ausleihfrist bereits abgelaufen ist, soll eine Hupe den Standort des Ausleihers vor dem Schalter beschallen.
Technische Anforderungen	Das System muss mit einer CORBA-Architektur entwickelt werden.
	Alle Signalkabel des Systems müssen dem Standard der ETHERNET-Kategorie 5: 10BASE5, 10BASE2 oder 10BASET entsprechen.
	Das System muss ausschließlich mit der Programmiersprache Java entwickelt sein.
	Das System muss eine Hupe der Firma „Freighter-Gear“, Modell „Nebelfrei“ verwenden.

Gruppe	Beispiele
<b>Art (Forts.)</b>	
Anforderungen an die Benutzerschnittstelle	Das System muss in der Liste der ausgeliehenen Bücher die Schriftart Helvetica in einer Größe von 11pt verwenden.
	Das System muss den „Entleihen“-Button blau darstellen.
	Das System muss den angeschlossenen Systemen die Möglichkeit bieten, die gespeicherten Objekte im Format XY abzurufen.
	Das System muss, wenn die Hupe aktiviert wird, diese für fünf Sekunden ertönen lassen.
Qualitätsanforderungen	Das System muss sich im Falle des Auftretens eines Fehlers der Kategorie 1 innerhalb von 20 Stunden wieder auf vollem Leistungsniveau befinden.
	Das System muss jede Einzelanfrage an den Bestand der Zentralbibliothek innerhalb von 30 Sekunden ausführen.
	Wenn ein Fehler der Kategorie 2 oder 3 auftritt, muss das System den Benutzer auf mögliche Ursachen hinweisen.
	Das System muss die Hupe so verwenden, dass diese ihre volle Lautstärke von 112 Dezibel innerhalb von 30 Millisekunden erreicht (Attack-Time).
Anforderungen an sonstige Lieferbestandteile	Der Fleisch-Lesbarkeits-Index der Benutzerdokumentation muss einen Wert von 50 überschreiten.
	Das zu liefernde Klassenmodell muss so gegliedert sein, dass jeder Sachverhalt auf einer DIN A4-Seite dargestellt werden kann.
	Ein Software-Pflege- und -Änderungskonzept, das dem V-Modell-Standard entspricht, muss zum Zahlungsmeilenstein 3 vorgelegt werden.
	Der Auftragnehmer muss mit den zukünftigen Nutzern des Systems eine 5-tägige Schulung im technischen Bereich durchführen.
	Die Hupe muss mit drei farblich verschiedenen Tarnüberzügen geliefert werden.
Anforderungen an die Durchführung der Entwicklung	Der Auftragnehmer muss mit dem Auftraggeber monatliche Reviews der zu erstellenden Dokumente durchführen.
	Der Auftragnehmer muss das OOA-Modell mit dem Tool „quick-OOA“ entwerfen.
	Bei der fachlichen Abnahme des Systems muss der Auftragnehmer darauf achten, dass die Vertreter des Auftraggebers einen Gehörschutz tragen.
Rechtlich-vertragliche Anforderungen	Alle Änderungen, die der Auftragnehmer an den Anforderungen vornehmen möchte, müssen vom verantwortlichen Vertreter des Auftraggebers durch Unterschrift genehmigt werden, bevor sie für die Entwicklung relevant werden.
	Wird ein Meilenstein vom Auftragnehmer um mehr als zwei Wochen überschritten, ohne ein Ergebnis vorzulegen, oder entspricht mindestens ein Ergebnis zu diesem Zeitpunkt nicht den Anforderungen, so ist der Lenkungsreis einzuberufen.

## 6 Anforderung oder Anforderung

Gruppe	Beispiele
<b>Art (Forts.)</b>	
Rechtlich-vertragliche Anforderungen (Forts.)	Der Auftraggeber leistet für jeden Meilenstein, der abgenommen wurde, ein Drittel der vertraglich vereinbarten Summe für die Entwicklung des Systems.
<b>Ebene</b>	
Level 0	Es soll ein System geschaffen werden, das die verwaltungstechnischen Abläufe der Stadtbibliothek besser gestaltet, damit weniger Leihobjekte abhanden kommen.
	Bis Ende 2001 soll jeder zweite Bürger einen Bibliotheksausweis besitzen.
	Dem gegen Regeln der Bibliothek verstoßenden Ausleiher soll eine Lehre erteilt werden.
Level 1	... Das alte Verfahren, Ausleihen handschriftlich auf den Leihobjektkarten zu kennzeichnen, ist zu fehleranfällig. Es muss zu jeder Zeit bekannt sein, ob sich ein Leihobjekt im Bestand befindet oder ob es verliehen ist. Falls es verliehen ist, muss der Entleiher bekannt sein. Das neue System soll dies unterstützen, indem die Informationen zu den Leihobjekten elektronisch gespeichert werden ...
Level 2	Das System muss Entleiher eindeutig identifizieren können.
	Das System muss dem Bibliothekar beim Entleihvorgang eine Liste aller nicht entliehenen Exemplare des Leihobjektes anzeigen.
	Das System muss sicherstellen, dass entlehene Objekte erst wieder entliehen werden können, nachdem sie zurückgegeben wurden.
Level 3	Das System muss es dem Bibliothekar beim Entleihvorgang ermöglichen, die ISBN des Leihobjektes einzugeben.
	Das System muss nach Eingabe der ISBN den Bestand durchsuchen und alle Exemplare finden, die aktuell nicht entliehen sind und die eingegebene ISBN haben.
	Das System muss alle im Bestand gefundenen Exemplare des Leihobjektes nach der eindeutigen Exemplar-Nummer aufsteigend sortiert anzeigen.
Level 4	Die Datenbank-Komponente muss für das Durchsuchen des Bestandes anhand des Kriteriums „ISBN“ weniger als drei Sekunden benötigen.
	Die durchschnittliche Datenübertragungsrate zwischen Terminal und Server muss mindestens 28 kbit/s betragen.



## 6.6 Management-Zusammenfassung

*Anforderung.* Der Begriff „Anforderung“ ist recht weit zu fassen. Hier die genaue Definition:

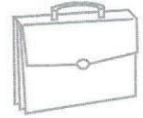
Eine Anforderung ist eine Aussage über eine Eigenschaft oder Leistung eines Produktes, eines Prozesses oder der am Prozess beteiligten Personen.

Anforderungen sollten differenziert werden nach den Faktoren Priorität, Art und Detailebene:

*Art.* Ordnen Sie Anforderungen einer von acht Arten zu: Man unterscheidet funktionale Anforderungen, technische Anforderungen, Anforderungen an die zu verarbeitenden Informationen, Anforderungen an die Benutzerschnittstelle, Qualitätsanforderungen, Anforderungen an sonstige Lieferbestandteile, Anforderungen an die Durchführung der Entwicklung und rechtlich-vertragliche Anforderungen.

*Detailebene.* Klassifizieren Sie jede Anforderung nach ihrer Granularität (Abstraktionsgrad, Detaillierung). Mindestens fünf Stufen kann man unterscheiden, in der Praxis sind drei bis vier sinnvoll. Sie sollten Anforderungen generell auf mehreren Ebenen schreiben.

*Priorität.* Anforderungen sind einzeln gemäß ihrer Priorität zu gewichten. Dies gilt insbesondere in sicherheitskritischen Anwendungsbereichen (Kritikalität) und bei Vertragsmodellen, in denen es wichtig ist, die Verbindlichkeit verschiedener Anforderungen festzulegen.



## 6.7 Kontrollfragen

- Haben Sie Anforderungen aller acht Arten bedacht?
- Unterscheiden Sie die verschiedenen Ebenen von Anforderungen ausreichend fein? Ist die Anzahl der Ebenen noch handhabbar?
- Sind die einzelnen Anforderungen priorisiert? Haben Sie es durch weniger wichtig eingestufte Anforderungen ermöglicht, die Releases flexibel zu planen?
- Haben Sie den Eindruck, dass Ihre Gruppierungen für Ihre Zwecke nützlich sind?

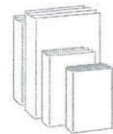
## 6.8 Weiterführende Literatur

[Boehm2000]

Boehm, B.: Software Cost Estimation with Cocomo II, Prentice Hall 2000, ISBN 0-13-026692-2

[Boehm95]

B. Boehm, B.: Software Requirements Negotiation and Renegotiation Aids: A Theory-W Based Spiral Approach, Proceedings of the 17th International Conference on Software Engineering (ICSE-17) 1995



[CHAOS2003]

The Standish Group International, Inc: CHAOS Chronicles 2003 Report, West Yarmouth 2003, [www.standishgroup.com](http://www.standishgroup.com)

[Cockburn2002]

Cockburn, A.: Agile Software Development, Boston, Addison-Wesley 2002. ISBN 0-201-69969-94

[Davis93]

Davis, A.M.: Software Requirements – Objects, Functions, & States, Prentice Hall 1993. ISBN 0-13-805763-x

[DIN94]

Norm DIN 66272: Bewerten von Softwareprodukten: Qualitätsmerkmale und Leitfaden zu ihrer Verwendung. Ausg. Oktober 1994. DIN Deutsches Institut für Normung e.V.

[IEEE610]

IEEE Std 610.12-1990, IEEE Standard Glossary of Software Engineering Terminology, IEEE Press 1990

[IEEE830]

IEEE Std 830-1998, IEEE Recommended Practice for Software Requirements Specifications, IEEE Press 1998

[Karlsson97]

Karlsson, J. und Ryan, K.: A Cost-Value Approach for Prioritizing Requirements, IEEE Computer, September/October 1997

[Leite93]

Leite, J.: A Strategy for Conceptual Model Acquisition, Proceedings of the First IEEE International Symposium on Requirements Engineering, SanDiego, Ca, IEEE Computer Society Press, 1993

[VMM97]

Allgemeiner Umdruck Nr. 251, Entwicklungsstandard für IT-Systeme des Bundes, Methodenzuordnung, Juni 1997

[Wieggers99]

Wieggers, K.: First Things First: Prioritizing Requirements, Software Development, vol. 7, no. 9, September 1999, oder auch <http://www.processimpact.com/pubs.shtml>

[Young2001]

Young, R.R.: Effective Requirements Practices, Addison-Wesley 2001. ISBN 0-20170-912-0

[Young2004]

Young, R.R.: The Requirements Engineering Handbook, Artech House 2004. ISBN 1-58053-266-7