

## **Business Capabilities**

Geschäftsfähigkeiten als effektives Werkzeug für die Gestaltung von Unternehmens- und IT-Architekturen

» Hier geht's  
direkt  
zum Buch

# **DIE LESEPROBE**

## 7 Modellierung von Business Capabilities

Die Voraussetzung für erfolgreiches Arbeiten mit Capabilities ist ein passendes, gut modelliertes Capability-Modell. Die in Kapitel 2 definierten Eigenschaften von Capability-Modellen stellen notwendige Voraussetzungen für gute Capability-Modelle dar, sind jedoch alleine nicht hinreichend für deren Qualität. Es gibt unzählige Möglichkeiten, Capabilities für einen Betrachtungsbereich zu definieren und zu schneiden, von denen nicht *die eine* die einzig richtige Variante ist. Als Hilfestellung zur Modellierung guter Capabilities beschreibt das vorliegende Kapitel einige systematische Ansätze für einen konsistenten Schnitt und gibt Empfehlungen.

### 7.1 Entwurfs- und Schnittkriterien

Für den Schnitt von Capability-Modellen kann man einige grundlegende Ansätze unterscheiden, die in den folgenden Abschnitten beschrieben sind. Capability-Modelle lassen sich komplett oder in Teilbereichen nach diesen Ansätzen modellieren. Die wichtigsten Schnittkriterien sind die drei folgenden:

- Objektorientierter Schnitt
- Funktionsorientierter Schnitt
- Kontextorientierter Schnitt

Für große Capability-Modelle sind »sortenreine« Schnitte kaum realisierbar bzw. nicht sinnvoll. Abgesehen davon sind die Entwurfskriterien auch nicht absolut trennscharf. In Capability-Modellen findet man daher oftmals Capabilities, deren Definition man mehrere Arten von Schnittkriterien zuordnen kann. Dies wird anhand der Beispiele in den nachfolgenden Abschnitten deutlich werden. Je nach Ebene des Capability-Modells sind andere Schnittkriterien sinnvoll, was Abschnitt 7.1.4 näher beleuchtet.

#### 7.1.1 Objektorientierter Schnitt

Der objektorientierte Schnitt nutzt eine auf *Objekte* fokussierte Perspektive zur Definition und Gruppierung von Capabilities. Dieser Schnitt fasst Fähigkeiten,

die sich auf denselben Objekttyp beziehen, auf der nächsthöheren Capability-Ebene zusammen. Unter dem allgemeinen Begriff des Objekttyps lassen sich sowohl Geschäfts- als auch Informationsobjekttypen erfassen. Ein Geschäftsobjekt ist eine reale Entität, die entweder dinglich oder ideell sein kann. Ein Fahrzeug oder eine Produktionsmaschine sind Beispiele für Geschäftsobjekte dinglicher Natur. Ein Kundenauftrag, eine Rechnungsposition oder eine Dienstleistung sind in erster Linie ideelle Geschäftsobjekte, für die es dingliche Repräsentationen geben kann, z.B. in Papierform.

Für alle Geschäftsobjekte gilt, dass sie digital durch ein Informationsobjekt – den sogenannten *digitalen Zwilling* – repräsentiert werden können. Ein Fahrzeug lässt sich beispielsweise durch ein Informationsobjekt repräsentieren, in dem man wichtige Eigenschaften wie die Fahrzeugidentifikationsnummer, das Fahrzeugmodell, seine Ausstattung, das Produktionsdatum usw. speichert. Es kann mehrere Informationsobjekttypen zur Repräsentation von Fahrzeugen geben. Einen Informationsobjekttyp könnte ein Fahrzeughändler nutzen, um z.B. Attribute wie Lieferzeit, Preis usw. zu speichern. Einen anderen Informationsobjekttyp könnte eine Kunden-App nutzen, die für ein Fahrzeug Attribute wie verbleibender Batterieladestand / Tankinhalt, KM-Stand und das letzte Navigationsziel definiert.

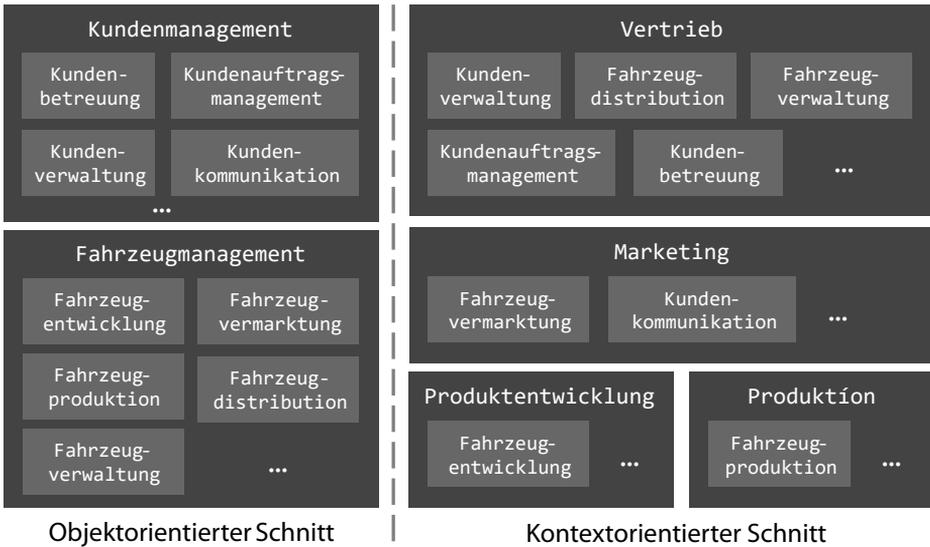
Da die Digitalisierung für alle relevanten Geschäftsobjekttypen ohnehin eine elektronische Repräsentation im Sinne eines Informationsobjekttyps erfordert, wird begrifflich oftmals gar nicht mehr explizit zwischen beiden unterschieden. Das Schnittkriterium hätte daher auch »informationsorientierter« Schnitt heißen können. Die Bezeichnung »objektorientierter« Schnitt scheint passender bzw. allgemeingültiger, da bei der Definition von Capabilities eine digitale Abbildung von Objekten im Sinne von Informationsobjekten nicht zwingend ist.

**Beispiel.** Als Beispiel sei der Informationsobjekttyp *Auftrag* betrachtet, für den sich bei einem objektorientierten Schnitt folgende Capabilities definieren und zu einer übergeordneten Capability *Auftragsmanagement* zusammenfassen lassen:

- Auftrag erfassen
- Auftrag delegieren
- Auftrag prüfen
- Auftrag löschen
- Auftragsstatus verwalten
- Auftrag stornieren

Alle untergeordneten Capabilities nutzen (benötigen, verarbeiten) Instanzen desselben Informationsobjekttyps *Auftrag*.

Abbildung 7–1 zeigt an weiteren Beispielen den Unterschied zwischen objektorientiertem und dem in Abschnitt 7.1.3 beschriebenen kontextorientierten Capability-Schnitt.



**Abb. 7-1:** Objekt- vs. kontextorientierter Capability-Schnitt am Beispiel

Beim objektorientierten Schnitt sind die Capabilities den beiden Informationsobjekttypen Kunde und Fahrzeug bzw. der jeweiligen Management-Capability für den Informationsobjekttyp zugeordnet. Innerhalb dieser beiden übergeordneten Capabilities gibt es keine weitere Gruppierung nach Kontext oder funktionaler Ähnlichkeit. Im Gegensatz dazu hat der kontextorientierte Schnitt vier Capabilities auf oberster Ebene, da für die betrachteten Unter-Capabilities vier fachliche Kontexte relevant sind. Capabilities für dieselben Informationsobjekttypen sind hier nicht gebündelt, sondern über die Kontext-Capabilities verteilt.

### Vorteile des objektorientierten Schnitts

Der objektorientierte Schnitt hat einige Vorteile, die im Folgenden kurz zusammengefasst sind:

- Gute zeitliche Stabilität des Informationsobjektmodells** – Die für Unternehmen relevanten Informationsobjekttypen weisen – zumindest auf den höheren Granularitäts- bzw. Abstraktionsstufen – eine hohe Stabilität auf, da sie eng an das Geschäftsmodell geknüpft sind. Beim objektorientierten Schnitt überträgt sich diese Stabilität auch auf die Capabilities.
- Erhöhung von Vertrauen und Akzeptanz** – Die Geschäfts- und Informationsobjekttypen sind in Unternehmen typischerweise als Begriffe bzw. relevante Elemente gut bekannt, und im Idealfall herrscht Konsens über die (semantische) Definition der Objekttypen. Durch einen objektorientierten Schnitt sorgt man für eine gute Verständlichkeit der Capabilities, da die Informationsobjekttypen

pen einen hohen Wiedererkennungswert haben. Damit erhöht sich das Vertrauen und auch die Akzeptanz in Bezug auf das Capability-Modell.

#### **Verbesserte Konsistenz von Informations- und Capability-Modellen – Ohne**

Nutzung eines objektorientierten Schnitts modelliert man schlimmstenfalls an den etablierten und bekannten Informationsobjekttypen vorbei. Dies kann zu inkonsistenten Begriffswelten und Unverständlichkeit des Capability-Modells führen. Auch wenn man nicht explizit einen objektorientierten Schnitt nutzt, so sollte man dennoch auf konsistente Begriffe achten.

#### **Nachteile des objektorientierten Schnitts**

Der objektorientierte Schnitt hat nicht nur Vorteile. Die beiden wesentlichen Nachteile sind im Folgenden kurz beschrieben:

**Gefahr von Redundanzen im Modell** – Funktional gleichartige Fähigkeiten, die für mehrere verschiedene Informationsobjekttyp in analoger Weise benötigt werden, führen im objektorientierten Schnitt zu mehreren Capabilities – jeweils eine pro Informationsobjekttyp. Bei großer Ähnlichkeit bzw. Verwandtschaft der Objekttypen befinden sich im Modell an verschiedenen Stellen die quasi gleichen Capabilities.

**Erschwerte Identifizierung funktionaler Synergien** – Dieser Nachteil hat dieselbe Ursache wie der erste: Durch die strikte Unterteilung der Capabilities nach Objekttyp lassen sich funktionale Synergien nur schwer identifizieren. Ein Beispiel sind die beiden Capabilities Rechnungsdruck und Mahnungsdruck, die aus den beiden Informationsobjekttypen Rechnung und Mahnung abgeleitet wurden. Auch wenn sich noch argumentieren lässt, dass sich diese im engeren Sinne nicht überschneiden, so ist es aus Synergiegründen vermutlich sinnvoller, eine allgemeinere Capability Dokumentendruck zu definieren.

#### **Hinweise zum Vorgehen**

Ein mögliches Vorgehen besteht darin, zunächst alle relevanten Informations- bzw. Geschäftsobjekttypen zu identifizieren und dann zu ermitteln, welche Fähigkeiten der aktuelle Betrachtungskontext in Bezug auf jeden einzelnen Objekttyp erfordert. Kandidaten für relevante Informationsobjekte kann man beispielsweise aus vorhandenen konzeptionellen oder auch technischen Informations- bzw. Datenmodellen herausziehen. Auch Prozessmodelle eignen sich als Informationsquelle, sofern diese auch die in den Abläufen verarbeiteten Geschäftsobjekttypen umfassen.

Aus dem betrachteten Beispiel des Auftragsmanagements lässt sich ein schematischer Ansatz ableiten: Für einen relevanten Informationsobjekttyp ABC definiert man eine Capability ABC-Management, unter die alle Fähigkeiten als unter-

geordnete Capabilities einsortiert werden, die das Management (Handhabung, Verwaltung usw.) des Informationsobjektyps ABC erfordert.

### 7.1.2 Funktionsorientierter Schnitt

Beim funktionsorientierten Schnitt steht die funktionale Art der Fähigkeit unabhängig von Informationsobjektypen und fachlichen Kontexten im Vordergrund. Die Zusammengehörigkeit von Unter-Capabilities innerhalb einer übergeordneten Capability orientiert sich an der funktionalen Ähnlichkeit (oder auch Gleichartigkeit) der bereitzustellenden Fähigkeit bzw. der zu bearbeitenden Aufgabe. Die Definition einer einzelnen Capability erfolgt beim funktionsorientierten Schnitt nach der Art der Aktivität, die mit der Fähigkeit verbunden ist.

**Beispiel.** Die folgenden Capabilities sind Beispiele für funktionsorientiert geschnittene Fähigkeiten, bei denen jeweils eine bestimmte Art von Aktivität, d.h. eine fachlich spezifische »Funktion«, in Form der Capability erfasst wird.

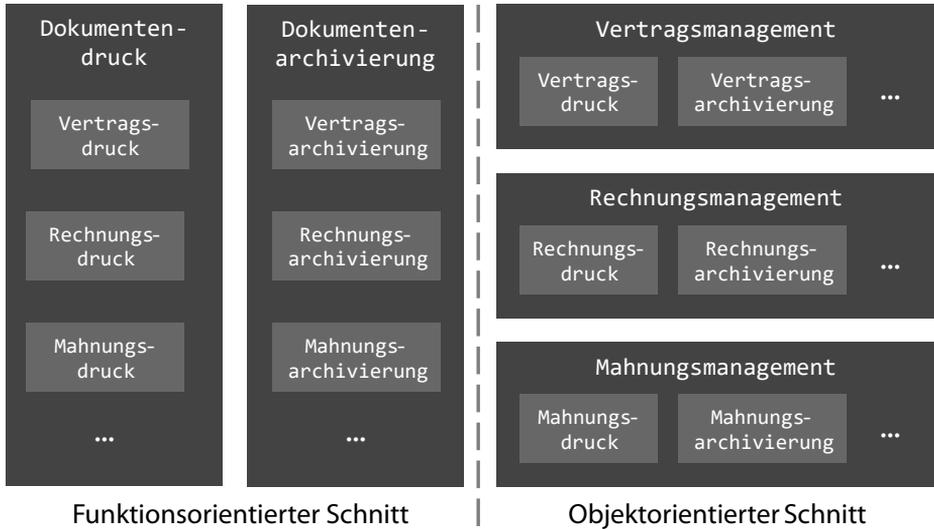
- Kundenbeziehungsmanagement
- Projektzeiterfassung
- Beschwerde- und Regressbehandlung
- Preisgestaltung
- Betrugserkennung
- Dokumentendruck

Auch wenn die Capability Dokumentendruck ebenso als objektorientiert geschnitten (im Hinblick auf den sehr allgemeinen Informationsobjektyp Dokument) gelten kann, so lässt sie sich dennoch auch als funktionsorientiert betrachten, da Drucken eine spezifische funktionale Fähigkeit ist. Die untergeordnete Ebene würde dann möglicherweise für verschiedene Informationsobjektypen spezialisierte Capabilities definieren:

- Vertragsdruck
- Rechnungsdruck
- Mahnungsdruck
- ...

Damit hat man die Fähigkeit, etwas drucken zu können, für alle Informationsobjektypen in einer gemeinsamen, übergeordneten Capability gebündelt. Bei einem objektorientierten Schnitt würde man ebenfalls einzelne für den jeweiligen Informationsobjektyp spezifische Druck-Capabilities modellieren, diese jedoch nicht in einer übergeordneten Dokumentendruck-Capability zusammenfassen, sondern z.B. in separaten Management-Capabilities je Informationsobjektyp modellieren.

Abbildung 7–2 zeigt das Beispiel in grafischer Form. Die Abbildung stellt beide Modellierungsansätze gegenüber.



**Abb. 7–2:** Funktions- vs. objektorientierter Capability-Schnitt am Beispiel

Welche Variante die bessere ist, lässt sich nicht pauschal beurteilen. Der funktionsorientierte Ansatz eignet sich besser, wenn man den Fokus auf die Gemeinsamkeiten bzgl. der Funktionen (hier des Druckens bzw. der Archivierung) legen möchte. Dies ist insbesondere dann eine gute Wahl, wenn das Vorhaben stärker auf die Technik fokussiert und – bezogen auf das Beispiel – passende Druck- bzw. Archivierungssysteme gesucht werden. Die objektorientierte Variante betont eher die fachlichen Gemeinsamkeiten in Bezug auf die Informationsobjekttypen. Dies ist vermutlich die bessere Wahl, wenn die Orientierung stärker in Richtung Fachlichkeit geht. Allerdings weiß man bei der Modellierung manchmal noch nicht, wo der Fokus der Anwendungsfälle liegen wird. Natürlich muss das Capability-Modell unabhängig von den gewählten Schnittkriterien alle notwendigen Eigenschaften erfüllen, sodass es sich grundsätzlich für alle Anwendungsfälle eignet.

### Vorteile des funktionsorientierten Schnitts

Der funktionsorientierte Schnitt hat einige Vorteile, die die folgende Zusammenstellung beschreibt:

**Gute Identifizierbarkeit von Synergien und Redundanzen** – Wegen der Fokussierung auf die funktionalen Aspekte lassen sich (funktionale) Synergien leichter identifizieren und Redundanzen (oder gar Überschneidungen) vermeiden. Im Beispiel oben lässt sich erkennen, für welche Informationsobjekttypen das

Unternehmen die Druck- und Archivierungsfähigkeiten benötigt. Dies zeigt sofort das Potenzial gemeinsamer Realisierungen auf, sodass nur jeweils ein technisches System benötigt wird, um alle Druck- und Archivierungs-Capabilities umzusetzen.

**Erhöhung der Wiederverwendbarkeit** – Eine funktionsorientiert geschnittene Capability wie Dokumentendruck ist unabhängig von konkreten Informationsobjekttypen definiert und ermöglicht damit einen höheren Wiederverwendungsgrad in verschiedenen fachlichen Kontexten bzw. für verschiedene Informationsobjekttypen.

### **Nachteile des funktionsorientierten Schnitts**

Der funktionsorientierte Schnitt birgt auch Nachteile, die im Folgenden kurz beschrieben sind:

**Risiko zu hoher Generizität** – Abstrahiert das Modell beim funktionsorientierten Schnitt in zu hohem Maße von Informationen und fachlichem Kontext, ergeben sich schnell zu stark verallgemeinerte Capabilities, die dann nur noch schwer zu greifen sind. Mögliche Synergien sind dann eventuell so allgemein, dass sie für die praktische Umsetzung unrealistisch sind. Beispiel hierfür könnte die schon mehrfach als Beispiel herangezogene Capability Auftragsmanagement sein. Wenn es in einem Unternehmen mehrere Arten von Auftragsmanagement für verschiedene Auftragstypen in verschiedenen Kontexten gibt, dann scheint der funktionsorientierte Schnitt zunächst naheliegend, um die funktionalen Gemeinsamkeiten zu erfassen. So kann es z.B. bei einem Automobilhersteller (neben weiteren möglichen Varianten) ein Kundenauftragsmanagement, ein Produktionsauftragsmanagement und ein Logistikauftragsmanagement geben. Sind diese Arten des Auftragsmanagements in der Praxis so unterschiedlich, dass eine Synergie trotz der auch vorhandenen funktionalen Gemeinsamkeiten nicht realistisch ist, so ist es sinnvoller, die Capabilities getrennt voneinander (im Sinne eines kontextorientierten Schnitts) zu modellieren.

**Geringere fachliche Kohärenz** – Der Grund für diesen Nachteil ist derselbe wie bei dem Risiko zu starker Generizität. Fokussiert man zu wenig auf den fachlichen Kontext und die fachlichen Spezifika der Informationen, so sinkt unter Umständen die fachliche Kohärenz und somit auch die fachliche Verständlichkeit. Es ist ggf. ein höherer Aufwand zur Abstimmung und Einführung des Capability-Modells erforderlich.

### Hinweise zum Vorgehen

Ein möglicher Ausgangspunkt für einen funktionsorientierten Schnitt ist das Geschäftsprozessmodell, aus dem sich die enthaltenen Aktivitäten / Prozessschritte extrahieren und auf ihren funktionalen Gehalt hin untersuchen lassen. Eine weitere Informationsquelle können die fachlichen Applikationen sein, deren Funktionalitäten Hinweise auf Fähigkeiten liefern.

Ein genereller Ansatz ist das Aufsammeln von Capability-Kandidaten als erster Schritt (auf Basis welcher Quellen auch immer), bevor man diese anhand von funktionalen Gemeinsamkeiten zu übergeordneten Capabilities bündelt. Hier kann auch eine Betrachtung des Synergiepotenzials erfolgen, indem man von fachlichen Kontexten und Informationen abstrahiert, um auf den gemeinsamen funktionalen Kern zu fokussieren. Dabei sollte man darauf achten, dass weder die fachliche Kohärenz zu gering wird noch dass die sich ergebenden Capabilities zu abstrakt / zu generisch sind.

### 7.1.3 Kontextorientierter Schnitt

Die Gruppierung von Capabilities zu einer übergeordneten Capability erfolgt beim kontextorientierten Schnitt auf Basis des gemeinsamen fachlichen Kontexts. Beispiele für solche Kontexte können z.B. die Produktion eines Unternehmens oder sein Personalwesen sein. Diese fachlichen Kontexte können (annähernd) deckungsgleich mit Organisationseinheiten sein, müssen dies jedoch nicht. Die Deckungsgleichheit mit Organisationsstrukturen ist explizit keine geforderte Eigenschaft und somit nicht gleichbedeutend mit dem kontextorientierten Schnittkriterium.

Beim kontextorientierten Schnitt können sich bei den untergeordneten Capabilities sowohl verschiedene Informationsobjekttypen als auch verschiedene Arten von funktionalen Aspekten wiederfinden. Das Element, das für Kohärenz sorgt, ist der fachliche Kontext. Diese Art des Schnitts eignet sich insbesondere für die oberste Ebene des Capability-Modells, da die Capabilities auf dieser Ebene besonders mächtig sind und viele Unterfähigkeiten für verschiedene Informationsobjekttypen umfassen.

**Beispiel.** Tabelle 7–1 zeigt einige Beispiele für kontextorientiert geschnittene Capabilities. Der fachliche Kontext differenziert die Capabilities und findet sich ggf. auch direkt in deren Namen wieder.

Die beiden Capabilities *Produktionslogistik* und *Absatzlogistik* des Beispiels haben denselben funktionalen Kern der modellierten Fähigkeit. Mit einem funktionsorientierten Ansatz würde man diese in der nächsthöheren Ebene z.B. zu einer Capability *Unternehmenslogistik* zusammenfassen. Bleibt es beim kontextorientierten Schnitt, so würde sich die Capability *Produktionslogistik* bei-

spielsweise in einer übergeordneten Capability Produktion wiederfinden. Die Absatzlogistik würde in eine übergeordnete Capability Vertrieb passen.

Capability	Funktionaler Kern	Fachlicher Kontext
Produktionslogistik	Logistik	Produktion (z.B. eines Fahrzeugherstellers)
Absatzlogistik	Logistik	Absatz von Produkten im Sinne der Distribution hergestellter Waren
Vertriebssteuerung	Steuerung	Vertrieb eines Unternehmens

Tab. 7-1: Beispiele für Capabilities mit fachlichem Kontext

Der kontextorientierte Schnitt fokussiert weder auf funktionale Gemeinsamkeiten noch auf die Gruppierung anhand gemeinsam genutzter Informationsobjekttypen. Abbildung 7-3 zeigt an weiteren Beispielen den Unterschied zwischen kontext- und funktionsorientiertem Capability-Schnitt.

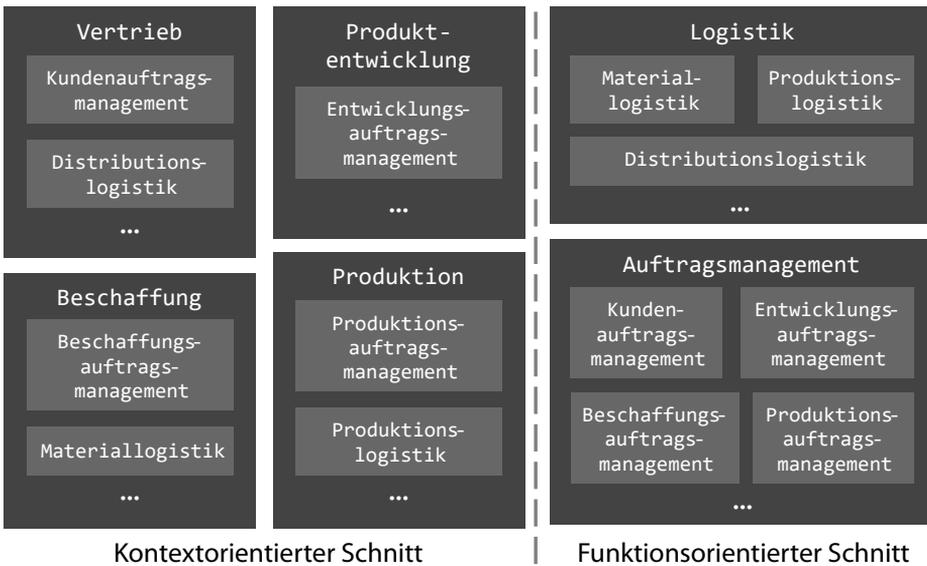


Abb. 7-3: Kontext- vs. funktionsorientierter Capability-Schnitt am Beispiel

Beim kontextorientierten Schnitt ist jede der Capabilities einem der vier fachlichen Kontexte Vertrieb, Produktentwicklung, Beschaffung und Produktion zugeordnet. Verschiedene, funktional ähnliche Capabilities (wie z.B. Kundenauftragsmanagement, Beschaffungsauftragsmanagement und Produktionsauftragsmanagement)

sind auf die verschiedenen Kontext-Capabilities verteilt. Im funktionsorientierten Schnitt sind diese wegen ihrer funktionalen Ähnlichkeit gebündelt. Dafür verteilen sich dort die zum selben fachlichen Kontext gehörigen Capabilities auf verschiedene übergeordnete Capabilities. Im kontextorientierten Modell umfassen die übergeordneten Capabilities Unter-Capabilities, die verschiedene funktionale Aspekte abdecken.

### **Vorteile des kontextorientierten Schnitts**

Der kontextorientierte Schnitt bietet verschiedene Vorteile, die im Folgenden zusammengefasst sind:

**Wiedererkennbare Ordnungsstrukturen** – Oftmals entsprechen die zum Schnitt verwendeten Kontexte bereits vorhandenen Ordnungsstrukturen, wie z.B. Unternehmensbereichen oder Teilen der Wertschöpfungskette. Dieser Wiedererkennungseffekt kann eine verbesserte Verständlichkeit und Akzeptanz des Modells zur Folge haben. Achtsamkeit ist hier dennoch geboten, um die Verwechslungsgefahr von Capabilities mit anderen Elementen wie Organisationseinheiten oder Prozessen aufgrund ihrer Namensgleichheit zu minimieren.

**Kontextspezifische Differenzierungsmöglichkeit** – Die Hinzunahme des fachlichen Kontexts in die Definition einer Capability ermöglicht mit dem kontextorientierten Schnitt, eine rein funktionale Zusammenfassung zu vermeiden, die ggf. zu einem zu hohen Grad an Generizität führen würde. (Zur Erinnerung: Dies ist einer der möglichen Nachteile eines funktionsorientierten Schnitts.) Zwar minimiert man damit nicht die funktionale Redundanz des Modells, ermöglicht jedoch eine kontextspezifische Differenzierung der Fähigkeiten im Modell, die bei der späteren Verwendung des Capability-Modells nützlich sein kann. Beispielsweise lassen sich Funktionalitäten von Applikationen feiner durch die verschiedenen Kontexte abgrenzen (siehe Beziehungstyp 2 (Capability ◀ unterstützt Applikation) in Anhang A.3). Das Modell wird insgesamt fachlich reicher und differenzierter. Dennoch ist auch beim kontextorientierten Schnitt Aufmerksamkeit erforderlich, damit nicht zu viele verschiedene Kontexte im Capability-Modell entstehen. Dies vermeidet Redundanzen bzw. stellt die Überschneidungsfreiheit des Modells sicher.

### **Nachteile des kontextorientierten Schnitts**

Zwei Nachteile des kontextorientierten Schnitts beschreibt die folgende Aufstellung:

**Übernahme von Altlasten in das Modell** – Die Auswahl der für den Schnitt verwendeten Kontexte muss mit Bedacht erfolgen, denn ansonsten übertragen sich möglicherweise Nachteile von vorhandenen Ordnungsstrukturen wie z.B.

der Aufbauorganisation auf das Modell. So können sich ggf. unpassende Kontexte ergeben, in die (implizit oder explizit) ungeeignete Kriterien wie Standorte, spezifische Produktbezeichnungen o.Ä. einfließen.

**Erschwerte Erkennung funktionaler Synergien** – Auch wenn die Capability-Modelle im Resultat anders aussehen, so gilt dieser Nachteil in analoger Weise wie beim objektorientierten Schnitt. Capabilities, für die funktionale Synergien möglich sind, werden nicht nach diesem Kriterium gruppiert, sondern sind aufgrund der anderen Kontextzugehörigkeit möglicherweise »quer« über das Capability-Modell verteilt. Synergiemöglichkeiten in der Realisierung sind damit schwerer zu erkennen.

### Hinweise zum Vorgehen

Es gibt verschiedene Ansatzpunkte zur Identifizierung fachlicher Kontexte. Wichtig ist, dass die im Folgenden genannten Ausgangsgrößen Hinweise für fachliche Kontexte liefern, jedoch nicht mit diesen bzw. den abgeleiteten Capabilities gleichzusetzen sind.

Die Elemente der Wertschöpfungskette auf oberster Ebene sind Kandidaten für fachliche Kontexte, denn die Wertschöpfungskette erfasst wichtige Tätigkeiten zur Erstellung von Marktprodukten bzw. Erbringung von Dienstleistungen. Hinter diesen Tätigkeiten verbergen sich dazugehörige Fähigkeiten, und die Kriterien, nach denen die Wertschöpfungskette in Elemente unterteilt ist, können auch für das Capability-Modell geeignete Schnittkriterien darstellen. Dies gilt zumindest für die oberen Ebenen des Modells.

Auch die Aufbauorganisation ist grundsätzlich ein geeigneter Ausgangspunkt für die Identifizierung von fachlichen Kontexten. Der Schnitt von Organisationsstrukturen richtet sich ebenfalls – zumindest partiell – nach der Gleichartigkeit auszuführender Aktivitäten. Somit können auch die dafür erforderlichen Fähigkeiten ähnlicher Natur sein. Allerdings gibt es für Organisationseinheiten noch verschiedene andere mögliche Schnittkriterien, die sich nicht oder nur bedingt für den Schnitt von Capability-Modellen eignen. Dazu gehören z.B. geografische Kriterien (Region, Standort usw.), die Unterscheidung nach Produktausprägungen oder Zuständigkeiten für Technologietypen. Jede Organisationseinheit lässt sich daraufhin untersuchen, welche Fähigkeiten diese umsetzen und ob sich ein fachlicher Kontext zur Gruppierung daraus ableiten lässt.

#### 7.1.4 Wahl der Schnittkriterien

In den vorangegangenen Abschnitten wurden drei wesentliche Schnittkriterien für Capability-Modelle eingeführt und miteinander verglichen. Neben den grundsätzlichen Vor- und Nachteilen unterscheidet sich die Anwendbarkeit durch die Ebene des zu erstellenden Capability-Modells. Tabelle 7-2 zeigt Empfehlungen für

die Wahl der Schnittkriterien je Ebene des Capability-Modells. Die Anzahl der modellierten bzw. betrachteten Ebenen hängt vom Anwendungsfall ab.

Ebene	Schnittkriterien	Anzahl (Unter-)Capabilities
1	■ kontextorientiert	6 - 15
2	■ kontextorientiert ■ funktionsorientiert ■ objektorientiert	3 - 10
3	■ funktionsorientiert ■ objektorientiert	3 - 10
4	■ funktionsorientiert ■ objektorientiert	3 - 10

**Tab. 7-2:** Geeignete Schnittkriterien der Modellebenen

Der kontextorientierte Schnitt eignet sich vor allem für die obersten beiden Ebenen eines Capability-Modells, da Kontexte inhaltlich »breit« sein können und so zu der groben Granularität passen, die in den höheren Ebenen (insbesondere auf der obersten Ebene) benötigt wird. Der objektorientierte Schnitt ist für die oberen Modellebenen in der Regel nicht geeignet, da die zugrunde liegenden Informationsobjekte bereits zu feingranular sind und eine stärkere Aggregation der Informationen auf konzeptioneller Ebene nicht sinnvoll ist. Auch die Funktionen des funktionsorientierten Schnitts sind für die obersten Modellebenen oftmals schon zu detailliert / zu feingranular, um sie dort sinnvoll als Schnittkriterium einzusetzen. Umgekehrt verhält es sich auf den tieferen Ebenen des Capability-Modells. Hier kann der Schnitt zwar auch kontextorientiert erfolgen, wenn der jeweilige Kontext entsprechend feingranular gewählt ist. Besser eignen sich jedoch der funktions- oder objektorientierte Schnitt, da Funktionen bzw. Daten in der Regel ohnehin feingranularer sind.

Die in der Tabelle angegebenen Anzahlen von Unter-Capabilities sind Orientierungswerte, die sich in der Praxis bewährt haben. Generell gilt, dass bei mehr als 10 untergeordneten Capabilities die Übersichtlichkeit des Modells leiden kann. Ergeben sich im Laufe des Modellierungsprozesses einmal mehr als 10 Unter-Capabilities, so ist eine Prüfung angebracht, ob sich die übergeordnete Capability sinnvoll in zwei oder drei Capabilities aufteilen lässt.

Hat eine Capability weniger als drei Unter-Capabilities, dann ist die Frage, ob bereits alle zugehörigen Unter-Capabilities vollständig modelliert sind oder ob das Modell noch ergänzt werden sollte. Ansonsten passen vermutlich entweder die Granularität der übergeordneten Capability oder deren Ebene noch nicht. In je-

dem Fall sollte man sich das Modell noch einmal genau anschauen und die Gründe für die geringe Anzahl untergeordneter Capabilities herausfinden. Eine hohe oder geringe Zahl untergeordneter Capabilities muss nicht zwingend »pathologisch« sein, doch wenn es solche Fälle gibt, sollten sie gut begründbar sein.

### 7.1.5 Ungeeignete Entwurfs-/Schnittkriterien

Es gibt eine Reihe von Unternehmenselementen, bei denen die Idee naheliegend erscheinen mag, diese ebenfalls als Schnitt- bzw. Gruppierungskriterien für Capabilities einzusetzen. Tatsächlich eignen sie sich jedoch nicht dafür. Zusammengefasst lassen sich die in Tabelle 7–3 angegebenen wesentlichen Verhinderungsgründe identifizieren, weshalb sich Elemente *nicht* als Strukturierungskriterium für Capability-Modelle eignen. Die nachfolgenden Abschnitte benennen und diskutieren einige solcher Elemente und verweisen auf die jeweils zutreffenden Verhinderungsgründe aus der Tabelle.

Nr.	Grund
VG 1: Überlappung	Die Ausprägungen des potenziellen Gruppierungselements sind bzgl. der benötigten Fähigkeiten nicht überschneidungsfrei.
VG 2: Inkohärenz	Die Ausprägungen des möglichen Gruppierungselements benötigen eine Auswahl an Fähigkeiten, die (funktional) sehr verschiedenartig sind, sodass sich inkohärente Capability-Modelle ergeben würden.
VG 3: Veränderlichkeit	Das potenzielle Gruppierungselement ändert sich häufig, sodass die gewünschte Stabilität des Capability-Modells nicht erreicht werden könnte.
VG 4: Fähigkeitsbezug	Das mögliche Gruppierungselement ist so beschaffen, dass seine Definition bzw. seine Eigenschaften keinen sinnvollen Bezug zu Capabilities erlauben. Das bedeutet, es fehlt eine ausreichende bzw. verständliche Interpretation als <i>Fähigkeit</i> , da das Gruppierungselement von seiner Natur her vollkommen andere Aspekte erfasst als Fähigkeiten.
VG 5: Realisierungsbezug	Das fragliche Gruppierungselement umfasst in seiner Definition zu stark das »Wie«, d.h. die Art und Weise der Realisierung, was der Natur der Capabilities zuwiderlaufen würde. Das Element abstrahiert zu wenig von der Umsetzung und konkreten Gegebenheiten, sodass Aspekte einfließen würden, die die Beschränkung auf die reine Geschäftsfähigkeit (das »Was«) aushebeln würden.

**Tab. 7–3:** Gründe für die fehlende Eignung als Schnittkriterium