

# Modellbezogene Use-Case-Identifikation für die UCP-basierte Aufwandsschätzung

*Stephan Frohnhoff<sup>1)</sup>, Karsten Kehler<sup>1)</sup>, Reiner Dumke<sup>2)</sup>*

<sup>1)</sup>sd&m AG, München, frohnhoff@sdm.de

<sup>2)</sup>Universität Magdeburg, dumke@ivs.cs.uni-magdeburg.de

## Inhalt:

<b>1. Einführung in die UCP-Methode</b> .....	2
1.1 Abgrenzungen und Anwendbarkeit .....	7
1.2 Zielgruppen und Granularität.....	8
<b>2. Strukturelle methodische Grundlagen</b> .....	8
2.1 Use Cases.....	9
2.2 Use-Case-Szenarien .....	10
2.3 Use-Case-Schritte .....	10
2.4 Use-Case-Dialoge .....	12
2.5 Aktoren der Spezifikation.....	12
<b>3. Empirische Grundlagen für Use-Cases-Strukturierung</b> .....	12
3.1 Zweckmäßige Granularitäten.....	13
3.2 Berücksichtigung impliziter Funktionalität.....	14
3.3 Use-Case-Strukturen.....	15
3.4 Verwendung von Anwendungsfunktionen.....	16
3.5 Berücksichtigung der Batch-Funktionalität .....	17
<b>4. Praxisrelevante Durchführung einer UCP-Schätzung</b> .....	18
4.1 Erfassung der Use Cases.....	18
4.2 Reuse und intuitive Äquivalenz .....	21
4.3 Erfassung von Aktoren für die Schätzung .....	23
<b>5. Spezifika der Use-Case-Beschreibungsarten für die UCP-Methode</b> .....	24
5.1 Grobe Use-Case-Beschreibungen .....	24
5.2 Textuelle oder tabellarische Beschreibungen.....	25
5.3 UML-basierte Ablauf- bzw. Aktivitätendiagramme .....	27
5.4 Interpretation von Zustandsdiagrammen.....	29
5.5 CRUD-Diagramme .....	31
5.6 Interpretation von Sequenzdiagrammen .....	33
5.7 Fazit .....	34
<b>6. Mapping von weiteren Spezifikationsformen</b> .....	35
6.1 Dialogorientierte Beschreibungen.....	35
6.2 Grobspezifikationen mit Geschäftsprozessen .....	38
6.3 Funktional orientierte Beschreibungen .....	41
6.4 Fazit .....	43
<b>7. Zusammenfassende Hinweise für die Anwendbarkeit und Evaluierung</b> .....	44
<b>8. Literatur</b> .....	45

## Zusammenfassung

Die Use Case Point Methode (UCP-Methode) erlaubt eine schnelle Schätzung von zu erwartenden Aufwänden für Software-Entwicklungsprojekte. Basis für eine solche Schätzung sind Grob-Spezifikationen unterschiedlichen Formats und unterschiedlicher Granularität. Entscheidend für den Erfolg der UCP-Methode und die Vergleichbarkeit der Ergebnisse ist vor allem, ob und wie es gelingt, die vorliegende Spezifikation auf Use Cases im Sinne der UCP-Methode abzubilden. Dafür beschreibt der vorliegende Preprint einen Vorschlag für einen Leitfaden, der aus der Spezifikations-Praxis des Softwarehauses sd&m abgeleitet wurde und damit wesentliche Grundlagen für allgemein empirisch vergleichbare Aufwandsschätzergebnisse darstellt.

### Keywords

Projektschätzung, Top-Down-Schätzung, Aufwandsschätzung, Use Case Points, White-Box-Estimation

## 1 Einführung in die UCP-Methode

Die Use Case Points UCP Methode (kurz: UCP-Methode) wurde speziell für eine UML-basierte Software-Entwicklung konzipiert ([7], [10], [11], [20], [25]) und berücksichtigt die empirischen Grundlagen für eine möglichst genaue Aufwandsschätzung ([2], [4], [6], [14], [23], [29]). Dabei hat diese Methode eine ähnliche Struktur, wie die klassischen Function Point Methoden ([8], [15], [17], [19]). Als Zählmaß werden bei der UCP-Methode die *Use Cases* und *Aktoren* einer Spezifikation oder Anforderungsdefinition eines Software-Systems herangezogen. Konkret werden die folgenden Aspekte bewertet: *Anzahl Use Cases*, *Anzahl Aktoren*, *technische Randbedingungen (nicht funktionale Anforderungen)* sowie *Projektumgebungsfaktoren*.

Für jeden Use Case wird die Komplexität anhand der einzelnen Transaktionen klassifiziert. Die Transaktionen sind die einzelnen Schritte, die innerhalb des System-Szenarios einzuhalten sind. Hierbei wird von Karner [22] eine Intervallskala vorgeschlagen, um den Umfang eines Use Case in einzelne Komplexitäts-Klassen einzuteilen.

Komplexität	Beschreibung	Gewicht $W_i$
einfach	A use case is simple if it has 3 or less transactions including alternative courses. You should be able to realise the use case with less than 5 analysis objects.	5
mittel	A use case is average if it has 3 to 7 transactions including alternative courses. You should be able to realise the use case with 5 to 10 analysis objects.	10
komplex	A use case is complex if it has more than 7 transactions including alternative courses. The use case should at least need 10 analysis objects to be realised.	15

**Tab. 1:** Gewichtung der Use Case Kategorien nach Karner [22]

Ein Use Case mit einem Umfang aus ein bis drei Transaktionen heißt „einfach“ und bekommt ein Gewicht  $W_i$  von 5 Use Case Points. Ein Use Case mit einem Umfang von vier bis sieben Transaktionen im Standardablauf ist „mittel“ und bekommt 10 Punkte und ein „komplexer“ Use Case, der mehr als sieben Transaktionen hat, bekommt ein Gewicht  $W_i$  von 15 Use Case Points. Tabelle 1 fasst dies zusammen.

Die Gewichte  $W_i$  aller Use Cases werden dann summiert und ergeben die **Unadjusted Use Case Weights (UUCW)**.

Im zweiten Schritt werden die Aktoren bewertet, die den Use Case zugeordnet sind ([11], [26]). Die Aktoren werden in drei Komplexitäts-Klassen eingeteilt:

- Aktoren, die über Schnittstellen mit anderen Systemfunktionen (API) kommunizieren, werden als "einfach" betrachtet und bekommen einen Use Case Point.
- Aktoren, die über ein Protokoll, (z.B. TCP/IP) mit einem anderen System angebunden sind, werden in die Kategorie "mittel" zugeordnet und bekommen zwei Use Case Points.
- Aktoren, die ein GUI benutzen, um mit dem System zu kommunizieren, werden als „komplex“ eingestuft und bekommen drei Use Case Points als Gewichtungsfaktor.

Tabelle 2 fasst die Gewichtungen zusammen.

Komplexität	Beschreibung	Gewicht $W_i$
einfach	An actor is simple if it represents another system with a defined application programming interface.	1
mittel	An actor is average if it is: 1. An interaction with another system through a protocol. 2. A human interaction with a line terminal.	2
komplex	An actor is complex if it interacts through a graphical user interface.	3

**Tab. 2:** Komplexitäts-Klassen der Aktoren nach Karner [22]

Die Gewichte  $W_i$  aller Aktoren werden dann summiert und ergeben die **Unadjusted Actor Weights (UAW)**. Die Summe der Unadjusted Use Case Weights und der Unadjusted Actor Weights ergibt die **Unadjusted Use Case Points (UUCP)**.

$$UUCP = UUCW + UAW$$

Beim letzten Schritt der Aufwandsabschätzung werden die *UUCP* durch technische Faktoren und Umgebungsfaktoren korrigiert. Um die Komplexität der einzelnen technischen Faktoren zu ermitteln sind einzelne Einflussgrößen wie Effizienz, Sicherheitsanforderungen usw. einzeln zu bewerten. Diese Komplexität wird anhand einer Ordinalskala von null bis fünf Punkten bewertet. Null bedeutet, dass das Kriterium irrelevant ist. Fünf Punkte bedeuten, dass das Kriterium sehr wichtig ist. Außerdem werden für die einzelnen Eigenschaften noch zusätzliche Gewichte vergeben. Dadurch findet eine Gewichtung der Eigenschaften je nach Projektspezifikation statt.

Die folgende Tabelle 3 gibt einen Überblick über die Kriterien für den technischen Faktor und deren Gewichtung.

Nr.	Einflussgröße	Ge- wicht	Beschreibung
T1	Distributed System	2	Wie stark verteilt ist die Architektur des Systems? 0: Monolithisches System 3: 3-tier mit Randsystem 5: Hochverteilte Systemarchitektur
T2	Response Time	1	Wie hoch sind Performanz- und Lastanforderungen an das System? 0: keinerlei Anforderungen 3: übliche Performance-Lastanforderungen 5: hohe Performance-Lastanforderungen, z.B. Lastverteilung
T3	End User Efficiency	1	Wie effizient muss die Benutzungsschnittstelle zu bedienen sein? 0: keinerlei Anforderungen 3: normale Benutzungsschnittstelle, z.B. Web-GUI 5: Hochintegrierte effiziente Benutzungsschnittstelle, z.B. Makros
T4	Complex internal Processing	1	Wie komplex sind die Geschäftsregeln / die Berechnungen im System? 0: nur einfache Regeln, keine Berechnungen 3: normale Komplexität 5: sehr komplex
T5	Code must be Reusable	1	Wie hoch sind die Anforderungen an die Wiederverwendbarkeit des Codes? 0: keine Anforderungen 3: normale Anforderungen 5: hohe Anforderungen, z.B. Framework
T6	Easy to install	0,5	Wie einfach muss die Software zu installieren sein? 0: keine Anforderungen 3: normale Installationsanforderungen (dedizierte Kunden- abteilung installiert wenige Instanzen) 5: hohe Installationsanforderungen (eigenständige Installation durch eine hohe Zahl von Endkunden)
T7	Easy to use	0,5	Wie hoch sind die Anforderungen an die Benutzerfreundlichkeit? 0: keine Anforderungen (keine Benutzer) 3: normale Anforderungen (GUI, Hilfesystem) 5: hohe Anforderungen (z.B. GUI-Varianten für Benutzer- gruppen, Internationalisierung, Wizards usw.)
T8	Portable	2	Wie hoch sind die Anforderungen an die Portabilität des Systems? 0: keine Anforderungen, z.B. Software, die nur einmal läuft 3: normale Anforderungen (eine Zielplattform, übliche Grad an Abstraktion) 5: hohe Anforderungen (z.B. Cross-Plattform, Windows und Unix)

T9	Easy to change	1	Wie variabel (änderungsfreundlich oder anpassbar) muss das System sein? 0: keine Anforderungen, z.B. Software, die nur einmal läuft 3: normale Anforderungen, z.B. Konfigurierbarkeit 5: hohe Anforderungen, z.B. Anpassbarkeit über Templates, Plugin-Schnittstelle
T10	Concurrent	1	Wie hoch sind die Verfügbarkeitsanforderungen an das System? 0: keine Anforderungen, (keine parallelen Transaktionen) 3: übliche Verfügbarkeitsanforderungen 5: hohe Verfügbarkeitsanforderungen (24/7-Betrieb, 99,x% Verfügbarkeit, hohe Zahl paralleler Transaktionen)
T11	Includes special Security objectives	1	Wie hoch sind die Sicherheitsanforderungen an das System? 0: keine Anforderungen 3: übliche Anforderungen (Authentifizierung und Autorisierung) 5: Hohe Verfügbarkeitsanforderungen (Zertifikate, verschlüsselte Kommunikation, Kryptografie)
T12	Provides direct access for third parties	1	Bietet das System direkte Zugänge für Dritte (andere als den Kunden) an? 0: keine Anforderungen 3: Systemnutzung durch Endkunden (B2C-Kunden des Kunden) 5: z.B. externe Serviceschnittstellen, B2B-systeme, Handelsplattformen
T13	Special user training facilities are required	1	Ist die Software so komplex, dass besondere Schulungen erforderlich sind? 0: keine Schulung erforderlich 3: übliche Anwenderschulung oder Selbststudium 5: Training erforderlich

**Tab. 3:** Technischen Faktoren mit Komplexitätswerten und Gewichtung

Durch die Bewertung der einzelnen technischen Kriterien erhält man den **Technical Factor (TFactor)**.

$$TFactor = \text{sum}(T1 \dots T13)$$

Aus dem *TFactor* kann über eine Formel der **Complexity Factor (TCF)** berechnet werden

$$TCF = 0,6 + (0,01 * TFactor),$$

wobei *TFactor* die gewichtete Multiplikation der zu schätzenden Komplexitätsfaktoren darstellt. Der TCF liegt im Wertebereich [0,6 – 1,3]. Analog wird der Einfluss der Projektorganisation und Projektrahmenbedingungen anhand von acht Umgebungsfaktoren ermittelt. Analog zum technischen Faktor werden die Umgebungsfaktoren anhand einer Skala von null bis fünf Punkten bewertet und durch Gewichte justiert.

Die folgende Tabelle 4 gibt einen Überblick über die Umgebungskriterien und deren Gewichtung.

Nr.	Umgebungsfaktoren	Ge- wicht	Beschreibung
E1	Familiarity with project	1,5	Wie gut ist das Team, das an Projekt arbeitet, mit Domäne, technischen Details des Projektes vertraut? 0: gering vertraut 3: Details sind bekannt 5: das Team kennt gut alle Details und Domäne
E2	Application experience	0,5	Wie gut kennt das Team die Anwendung? 0: komplettes Neuland 3: Anwendung und Umfeld sind dem Team zum Teil bekannt, bzw. einem Teil des Teams bekannt 5: Anwendung und Umfeld sind dem Team vertraut (typisch bei hohen Releasenummern und Monitor Releases)
E3	Object-oriented programming experience	1	Wie erfahren ist das Team mit OO-Programmierung? 0: relativ unerfahren 3: normale Erfahrung mit OO-Programmierung 5: sehr erfahren
E4	Lead analyst capability	0,5	Wie erfahren ist der Chefdesigner? 0: relativ unerfahren für die Aufgabe, wenig vertraut mit der Anwendungsdomäne 3: normale Erfahrung für die Aufgabe; Anwendungsdomäne ist bekannt 5: sehr erfahren für die Aufgabe; sehr gut vertraut mit der Anwendungsdomäne
E5	Motivation	1	Wie motiviert ist das Team? 0: unmotiviert 3: motiviert 5: ausgezeichnet motiviert
E6	Stable requirements	2	Wie stabil sind die Anforderungen an das System? 0: sehr hohe Änderungsrate auch grundlegender Anforderungen 3: normale Änderungsrate, übliches Änderungsmanagement 5: sehr stabile Anforderungen, kein Änderungsmanagement erforderlich
E7	Part-time Staff	-1	Sind Personen an Projekt beteiligt, die nur Teilzeit arbeiten? 0: keine 3: teilweise 5: alle
E8	Difficult programming language	-1	Wie komplex ist die Sprache/ Entwicklungsumgebung des Systems: 0: einfach, z.B. Perl, PHP 3: normal (z.B. Java, Cobol) 5: komplex, exotisch, schlecht verstanden, z.B. Assembler

**Tab. 4:** Gewichtung der Umgebungsfaktoren

Nachdem alle Kriterien, die einen Einfluss auf der Umgebung haben, betrachtet wurden, ist der **Environmental Factor (EF)** zu berechnen.

$$EF = 1,4 - 0,03 * EFactor$$

wobei der *EFactor* die Multiplikation von Gewichtungen an geschätzte Komplexitätsfaktoren darstellt. EF liegt im Wertebereich [0,425 – 1,7]. Nachdem alle Schritte durchgeführt wurden und die Faktoren feststehen, lassen sich die **Adjusted Use Case Points (AUCP)** berechnen zu:

$$AUCP = UUCP * TCF * EF$$

Um Durch Multiplikation mit einem an der jeweiligen Organisation zu eichenden **Produktivitätsfaktor (PF)** im erwarteten Wertebereich [20 – 35 Stunden/UCP] ergibt sich daraus der geschätzte Projekt-Aufwand in Bearbeiter-Stunden (Bh).

$$Aufwand = AUCP * PF$$

## 1.1 Abgrenzungen und Anwendbarkeit

Ein wesentlicher Kritikpunkt an der UCP-Methode ist, dass Use Cases in unterschiedlicher Granularität beschrieben werden können und dies unmittelbar Einfluss auf das Schätzergebnis hat (siehe auch [1], [3], [16], [27]). In dem vorliegenden Artikel wird auf Basis von zahlreichen Projektbeispielen eine Anleitung und Hilfestellung zum "Schneiden" von Use Cases für die Abschätzung des funktionalen Umfangs von Software bei der Anwendung der UCP-Methode geben.

Im Folgenden wird ausschließlich der Begriff Use Case verwendet, er gilt aber synonym auch für Anwendungsfall. Zur Einbindung und Unterscheidung von spezifizierten Anwendungsfunktionen wird später noch etwas gesagt, die UCP-Methodik spricht grundsätzlich von Anwendungsfällen und damit Use Cases.

Die Kenntnis über die UCP Methode an sich und ihre Anwendbarkeit im Allgemeinen wird im Folgenden vorausgesetzt. An dieser Stelle wird ausschließlich auf den Aspekt der Bewertung des funktionalen Umfangs einer Anwendung anhand von Use Cases eingegangen.

Basis für eine solche Abschätzung ist immer mindestens eine vorliegende (Grob-) Spezifikation der Anwendung. Dabei ist es unerheblich, in welcher Form und Granularität diese Spezifikation vorliegt. Entscheidend für den Erfolg der Schätzung und die Belastbarkeit des Ergebnisses ist vor allem, ob und wie es gelingt, die vorliegende Dokumentation auf die hier beschriebenen Use Cases abzubilden. Im Idealfall entspricht die Spezifikation den aufgeführten Anforderungen und es ist nur noch eine quantitative Bewertung erforderlich.

Es nicht Ziel dieses Preprints, ein einheitliches Vorgehen bei der Beschreibung von Use Cases zu definieren. Es wird davon ausgegangen, dass im Zuge der Schätzung keine Use Case Beschreibung neu erstellt wird. Daher werden auch formale Aspekte einer Use Case Beschreibung nicht betrachtet (siehe hierzu [5], [9], [18], [21]).

Eine konkrete Use Case Beschreibung kann und darf sowohl in Form als auch in der Granularität durchaus von den hier genannten Kriterien abweichen ohne dass die Anwendbarkeit der UCP Methode eingeschränkt wird.

Die Abschätzung des funktionalen Umfanges ist ein wesentlicher Schritt zur Bestimmung des Aufwandes zur Umsetzung eines Softwareprojektes. Daneben spielen andere Kostenfaktoren eine Rolle, auf die hier nicht eingegangen wird. Verschiedene dieser Einflussfaktoren können gegen eine Anwendbarkeit der UCP Methode sprechen oder zumindest das Ergebnis stark verfälschen oder zu einer mit großer Unsicherheit behafteten Schätzung führen. Dies trifft auch für einige später genannte Kriterien im Bereich der Use Cases zu.

Ganz allgemein kann an dieser Stelle bereits gesagt werden, dass Schwierigkeiten beim Finden der Use Cases oder ihrer Bewertung darauf hindeuten, dass nicht genügend Informationen zur Abschätzung des funktionalen Umfanges vorhanden sind. In diesen Fällen wird auch die UCP Methode keine zuverlässigeren Werte als andere Methoden liefern können.

## **1.2 Zielgruppen und Granularität**

Zielgruppen für einen Use Case Modellierung sind der Auftraggeber eines Software-Projektes, der Konstrukteur, der Tester und der Schätzer! Diese unterschiedlichen Zielgruppen erfordern eine unterschiedliche Detaillierung der Use Cases, das gilt natürlich ebenso für den Schätzer. Günstigerweise erfordert die Schätzung keine zusätzliche Detaillierung, sie kann und sollte auf einem rein fachlichen Level erfolgen.

Eine Use Case Beschreibung von bzw. für einen Fachbereich (Auftraggeber) sollte grundsätzlich ausreichend Informationen für eine Schätzung des fachlichen Umfanges nach UCP enthalten. Entscheidend ist hierbei nicht der Detaillierungsgrad der Beschreibung eines einzelnen Schrittes oder auch der Dialogoberfläche, sondern die Trennung und Benennung der verschiedenen Use Cases inklusive ihrer Schritte, Szenarien und Dialoge.

Damit kann eine Grobspezifikation mit bloßer Aufzählung und kurzer Benennung dieser Komponenten sogar eine bessere (weil schneller zu erfassende) Basis für eine UCP Schätzung als eine detaillierte Beschreibung sein.

## **2 Strukturelle methodische Grundlagen**

Im Folgenden wird versucht, eine einheitliche Sicht auf Use Cases aus Sicht der UCP Schätzmethodik zu beschreiben. Sie basiert auf Definitionen, die in der Industrie übliche Praxis sind (siehe auch [12], [13], [24], [27]). Wir wollen dabei bewusst keine neuen Begriffe und Ebenen einführen, sondern die Ebene und damit auch Granularität finden, die für eine Schätzung nach UCP erforderlich ist. In den folgenden Abschnitten wird immer wieder von der "Zählung" von Szenarien, Schritten und Dialogen die Rede sein. Diese bezieht sich bereits auf die Anwendung der hier beschriebenen Begriffe für die UCP Schätzung.



Wie diese gezählten Einheiten erfasst werden und in die Schätzung eingehen, wird in Abschnitt 4 noch näher beschrieben.

## 2.1 Use Cases

Die sd&m Spezifikationsmethodik definiert einen Use Case wie folgt und sie soll auch für die UCP Methode genauso gelten:

- Ein Use Case spezifiziert exakt eine Aktivität der systemunterstützenden Aktivitäten eines Geschäftsprozesses.
- Er beschreibt das Verhalten und die Interaktion eines Systems als Reaktion auf die zielgerichtete Anfrage oder Aktion eines Aktors. Die Beschreibung des Use Cases erklärt sowohl das extern sichtbare als auch das detaillierte interne Systemverhalten in der Sprache und aus der Sicht der Anwender.
- Mit einem Use Case wird durch das System ein für den Anwender sinnvoller Dienst erbracht oder ein benutzbares Ergebnis erreicht.

Ganz entscheidend ist dabei der Hinweis, dass mit einem *Use Case* immer ein bestimmtes *fachliches Ziel* erreicht werden soll [11]. Ohne das Erreichen dieses Zieles werden wir in der Regel nur von Anwendungsfunktionen oder Schritten eines Use Cases reden.

Beispiele dafür sind alle funktional orientierten Beschreibungen oder auch dialogorientierte Beschreibungen. So ist z.B. das Verwenden eines Suchdialoges häufig kein eigener abgeschlossener Use Case, da das eigentliche fachliche Ziel mit dem Finden und Anzeigen der Treffermenge noch nicht erreicht ist, weil z.B. mit einem bestimmten Datensatz nach dem Suchen weitere fachliche Schritte durchgeführt werden.

Use Cases sind dadurch charakterisiert, dass sie immer eine Interaktion des spezifizierten Systems mit mindestens einem externen Aktor beinhalten. Dieser Aktor kann entweder ein menschlicher Nutzer, ein anderer Use Case (wie z.B. ein Batch) oder ein Nachbar-/Partnersystem sein.

Die Interaktion selber kann dabei sowohl von dem Aktor als auch von dem System ausgehen. Der Auslöser kann demzufolge sowohl ein externes Ereignis (z.B. Aufruf durch den Benutzer) als auch ein internes Ereignis im spezifizierten System (u.a. Zeitereignis) sein.

Bei sd&m unterscheiden wir keine verschiedenen Arten von Use Cases, wie z.B. Business Use Case, Background Use Case (Batch), Technical Use Case, Geschäftsvorfall oder Systemfunktion. Damit wird stellenweise eine Kategorisierung nach der unterschiedlichen Interaktion, dem Detaillierungsgrad der Beschreibung oder der Einordnung in die Geschäftsprozesse vorgenommen.

Auch für die UCP Methode ist diese Unterscheidung bei der Bestimmung und dem Finden von Use Cases nicht erforderlich und wird nicht verwendet. Allerdings wird später auf Besonderheiten einiger dieser Arten hingewiesen, wenn es um das Abschätzen des funktionalen Umfangs dieser Use Cases geht.

## 2.2 Use-Case-Szenarien

Die Szenarien eines Use Cases sind ein Kernbestandteil einer Use Case Beschreibung. Sie beschreiben den Ablauf von Aktionen der Aktoren und des Systems, der notwendig ist, das Ziel des Use Cases zu erreichen. Dazu zählen auch nichttriviale Fehlerszenarien.

Wir unterscheiden zwischen Erfolgs- und Alternativszenarien, die jedoch bei der Gewichtung des Use Cases gleichberechtigt behandelt werden.

Ein **Szenario** eines Use Cases ist der fachliche Ablauf zum fehlerfreien Erfolgsfall, der das fachliche Ziel des Use Cases ist. Jeder Use Case hat in der Regel genau ein solches Erfolgszenario.

Bei sehr komplexen Use Cases kann es vorkommen, dass es auch mehrere gleichberechtigte Erfolgsszenarien gibt (im Sinne mehrerer Hauptszenarien). Im Allgemeinen handelt es sich aber um untergeordnete Alternativszenarien. Werden mehrere dieser Hauptszenarien identifiziert, sollte eine Aufteilung in mehrere eigenständige Spezialisierungen des Use Cases in Betracht gezogen werden.

Neben dem Erfolgszenario können **Alternativszenarien** spezifiziert werden. Diese beschreiben separat fachliche Spezialfälle, z.B. beim Vorliegen besonderer Randbedingungen, und Fehlerfälle.

**Fehlerszenarien** sind solche, die nicht zum Erfolg (Erreichen des Business Goal) führen, wobei wir für die Zählung bei der UCP Schätzung zwischen trivialen und nichttrivialen Fehlern unterscheiden.

- fachliche Fehlerszenarien werden gezählt (wenn fachliche Schritte zur Fehlerbehandlung durchlaufen werden, z.B. Autokorrekturen, Änderung statt Neuanlage, ...)
- triviale Fehlerszenarien, z.B. „Anzeige einer Meldung, dann Abbruch“ werden nicht gezählt.

## 2.3 Use-Case-Schritte

Entscheidend für die Gewichtung eines Use Cases ist die Anzahl der für die Erreichung des fachlichen Zieles erforderlichen Schritte. Auch dafür gibt es leider keine allgemein anerkannte Definition, in verschiedenen Quellen wird auch von Aktionen, Operationen bzw. Folge von Operationen oder Aktivitäten gesprochen.

Systemseitige Aktionen werden vor allem wenn die Beschreibung bereits eine gewisse funktionale Dekomposition der Use Cases enthält auch als Anwendungsfunktionen bezeichnet.

Ein **Schritt** im Ablauf eines Use Cases ist ein in sich geschlossener fachlicher Teil des Use Cases, der vom folgenden Schritt und davorliegenden Schritt eindeutig getrennt ist durch z.B.

- den Wechsel des Aktors, oder der verarbeitenden "Schicht" (z.B. Eingabe im Dialog durch den Nutzer => Verarbeitung der Eingabe am Server => Anzeige des Ergebnisses an der Oberfläche)
- Erzeugen eines definierten (Zwischen-) Ergebnisses (z.B. Erzeugen von Druckdokumenten)
- Aufspalten eines neuen Szenarios

Entscheidend für die Use Case Modellierung ist hierbei immer die Betrachtung der obersten Ebene der Zerlegung aus Sicht des Use Cases. Jede weitere Verfeinerung ist für die Schätzung nach UCP dann nicht mehr relevant. Wird also z.B. in einer Beschreibung eines Use Cases von Aktivitäten geredet, die wiederum aus Operationen bestehen, so werden hier nur die Aktivitäten gezählt. Die sd&m Spezifikationsmethodik verwendet den Begriff Aktion, in der UCP Methodik verwenden wir gleichwertig den Begriff Schritt. Bei der Zählung berücksichtigt werden müssen:

- die Anzahl aller Schritte in allen Szenarien, es handelt sich also um eine Aufsummierung über alle Szenarien, wobei Schritte, die in mehreren Szenarien verwendet werden, nur einmal in die Zählung eingehen
- gleichwertig sowohl Benutzeraktionen (-schritte) als auch systemseitige Aktionen

Typische Beispiele für Schritte sind:

- Eingabe eines oder mehrerer Werte in einen Dialog (ohne dass dazwischen ein Server-Roundtrip erfolgt)
- Aufrufe von Anwendungsfunktionen
- Server-Transaktionen
- Durchführung fachlicher Prüfungen
- Erzeugung von Ausgaben
- triviale Auswahlvorgänge durch einen Akteur aus einer Anzeige heraus werden nicht als eigener fachlicher Schritt gezählt (Adresse auswählen), es können aber je nach Auswahl neue Szenarien entstehen

Eine Besonderheit bei den Schritten stellen fachlich sehr komplexe Abläufe dar, die nach obigen Kriterien in einem einzigen Schritt abgearbeitet werden. Diese würden mit der genannten Definition potentiell unterschätzt werden. Darauf wird speziell im Abschnitt 3.4 eingegangen.

## 2.4 Use-Case-Dialoge

Als weiterer Parameter geht die Anzahl der unterschiedlichen Dialoge eines Use Cases in die Schätzung ein. Unter einem Dialog wird für die UCP Schätzung in einem erweiterten Sinn jede Interaktionsschnittstelle verstanden.

Dialoge werden wie folgt gezählt:

- Jeder Reiter bzw. jede Maske eines Dialoges (mit signifikanten fachlichen Unterschieden) wird als eigener Dialog gezählt,
- jeder Frame einer Webseite (mit signifikanten Steuerelementen) wird als eigener Dialog gezählt,
- Triviale Pop-Up-Meldungen, Bestätigungen und Menüs werden nicht gezählt

Neben den klassischen Dialogen werden aber auch noch folgende Interaktionsschnittstellen zu den Dialogen gezählt und damit bei der UCP Schätzung berücksichtigt:

- Komplexe Schnittstellen, die Nachbarsystemen (zusätzlich zur Verarbeitung) zur Verfügung gestellt werden
- Druckstücke bzw. sonstige speziell erzeugte formatierte Ausgaben

## 2.5 Aktoren der Spezifikation

Zusätzlich zu den eigentlichen Use Cases gehen auch die **Aktoren** in die Schätzung ein. Damit sind alle (menschlichen und technischen) Nutzer des Systems gemeint, die die Use Cases auslösen oder im weiteren Ablauf mit ihnen interagieren. Der Begriff Aktor wird hier gleichwertig zum Actor verwendet.

Falls verschiedene Aktoren bezüglich eines Use Cases fachlich verschiedene Abläufe benötigen oder unterschiedlich mit der Anwendung interagieren sollten in diesem Fall auch neue Use Cases dafür verwendet werden oder zumindest innerhalb eines Use Cases neue Szenarien eingeführt werden.

Eine weitere Besonderheit bei der Erfassung der Aktoren für UCP ist die einmalige Zählung über alle Use Cases hinweg. Ein Aktor wird also nicht pro Use Case, sondern über die gesamte Anwendung nur einmal erfasst. Damit ist es besonders wichtig, Aktoren als Rollen bzw. als in Gruppen kategorisiert zu betrachten.

## 3 Empirische Grundlagen für Use-Cases-Strukturierung

In den folgenden Abschnitten sollen weitere ergänzende und allgemeine Hinweise für das "Schneiden" und damit Strukturieren von Use Cases für die Anwendung der UCP-Methode gegeben werden.

Dabei geht es nicht darum, Gestaltungsrichtlinien für die Spezifikation vorzugeben.

### 3.1 Zweckmäßige Granularitäten

Das Finden des richtigen Schnittes der Use Cases ist für die Schätzung ein ganz entscheidender Punkt. Ein genaues Maß für eine ausgewogene Use Case Beschreibung kann aber nicht angegeben werden, daher hier einige Richtwerte.

Ein Zeichen von zu komplexen (oder auch zu einfachen) Use Cases kann die Größe der betreffenden Beschreibungen im Spezifikationsdokument liefern. Wir haben verschiedene Kriterien, um an der Beschreibung festzustellen, ob der Umfang eines Szenarios zu groß gewählt ist:

- Die textuelle Beschreibung eines Szenarios umfasst mehr als eine DIN-A4-Seite *oder*
- Ein Szenario enthält mehr als 12 Schritte *oder*
- Ein Use Case beinhaltet mehr als sieben Verzweigungen, Szenarien. In diesem Fall sind die einzelnen Szenarien von der Gewichtung her eher als eigene Use Cases zu betrachten.

Übersteigt ein Use Case diese Werte wesentlich, ist er für die Schätzung zu umfangreich und sollte zerlegt werden. Die angegebenen Werte sind Erfahrungswerte aus nachgeschätzten Projekten und sollte wie alle angegebenen Grenzwerte im konkreten Fall nicht blind befolgt, sondern anhand eigener Erfahrungen oder anderer Use Cases überprüft werden.

Ebenso sind sehr kleine Use Cases, die sich vor allem in einer geringen Anzahl von Schritten, Szenarien und Dialogen zeigen, ein Anzeichen für einen potentiell falschen Schnitt. Zeichen dafür sind:

- Die Beschreibung eines Use Cases umfasst nur wenige Zeilen (Vorsicht, in einer Grobspezifikation kann dieses Kriterium sehr häufig zutreffen, wenn in einem Satz viele Schritte einfach nur aufgezählt werden, in diesem Fall kann der Schnitt trotzdem korrekt sein)
- Der Use Case enthält keinen Dialog und
- Der Use Case hat nur ein Szenario und
- Der Use Case hat nur einen oder zwei Schritte

Hier sollte noch einmal überlegt werden, ob es sich eventuell nicht eher um eine Anwendungsfunktion oder einen Teilschritt eines übergeordneten Use Cases handelt (z.B. "Suchen").

Wichtig ist aber auch, dass grundsätzlich „Ausreißer“ erlaubt sind. Auch hier eine hilfreiche Maßzahl: Es sollten sich ca. 80%-90% aller vorhandenen Use Cases in diesen Grenzen bewegen.

Die restlichen sind Ausnahmen, die sehr groß oder sehr klein, insgesamt aber vom Schnitt her zu den anderen "passen". Insgesamt ist eine ausgewogene Verteilung von kleinen, mittleren oder großen Use Cases ein Zeichen für einen "guten" Schnitt.

Es sei hier noch angemerkt, dass die Kriterien für die Größe der textuellen Beschreibung nur für eine schon als detaillierte Use Case Beschreibung vorliegende Spezifikation gelten. Die Angaben bezüglich der Anzahlen gelten natürlich auch nach einem aus einer anderen Form der Spezifikation erfolgten Mapping (siehe 6).

Für die Wiederholbarkeit und Konsistenz bzgl. einer konkreten Schätzung ist entscheidend, dass die Kriterien zum Finden von Use Cases und -schritten innerhalb Leitfaden zum Finden von Anwendungsfällen für die UCP Methode einer Schätzung einheitlich sind. Eine Gefahr unterschiedlicher Gewichtung besteht vor allem

- wenn mehrere (Teil-)Dokumentationen aus unterschiedlichen Projektphasen vorliegen
- wenn unterschiedliche Autoren an (Teil-) Dokumenten gearbeitet haben
- Dokumente unter Zeitdruck erstellt wurden und gegen Ende der Grad der Detaillierung nachlässt

Es empfiehlt sich daher, im Verlaufe einer UCP Schätzung immer wieder den "Schnitt" von Use Cases und -schritten zu vergleichen.

Eine einheitliches Vorgehen und vergleichbare Ergebnisse über mehrere Schätzungen hinweg ist vor allem das Ziel des hier vorliegenden Dokumentes.

Das bewährte Vorgehen, analog einer Expertenschätzung mehrere Personen unabhängig eine UCP Schätzung durchführen zu lassen, gilt selbstverständlich auch hier.

### **3.2 Berücksichtigung impliziter Funktionalität**

Häufig kommt es vor, dass neben der reinen fachlichen Funktionalität in gesonderten Kapiteln einer Spezifikation oder eines Lastenheftes weitere "Nebenfunktionalität" beschrieben ist oder diese auf jeden Fall benötigt wird. Dazu gehören z.B.

- Bereinigungsverfahren, Migrationsprogramme
- administrative Funktionalität, Verwaltungs- und Konfigurationsoberflächen

Diese Funktionalität muss selbstverständlich auch berücksichtigt werden. Es gibt auch keinen Grund, warum sich solche Funktionen nicht als Use Cases formulieren lassen sollten. In diesem Fall sind das z.B. ein Use Case „Nächtliche Bereinigung der Kontodifferenzen“, „Migration der Adressdaten“ oder „Konfiguration der Metadaten“. Entscheidend ist die vollständige Abdeckung aller spezifizierten (und implizit angenommenen) Anwendungsfunktionalität mittels der erfassten Use Cases.

### 3.3 Use-Case-Strukturen

Bei der Verwendung von Use Cases zur Spezifikation einer Anwendung ist auch die Möglichkeit vorgesehen, durch eine hierarchische Ordnung eine Aufrufstruktur von Use Cases zu verwenden. Verboten im Sinne einer guten und lesbaren Spezifikation sind Verzweigungen auf einzelne Szenarien oder Schritte eines anderen Use Cases. Wir wollen uns bei der UCP Methode daran halten.

Der aufrufende Use Case sorgt dafür, dass

- die Vorbedingung des aufgerufenen Use Cases erfüllt ist,
- der gesamte (Unter) Use Case durchlaufen wird

Nach vollständigem Durchlaufen des aufgerufenen Use Cases muss dessen Ergebnis gelten. Anschließend setzt der aufrufende Use Case seine Bearbeitung fort.

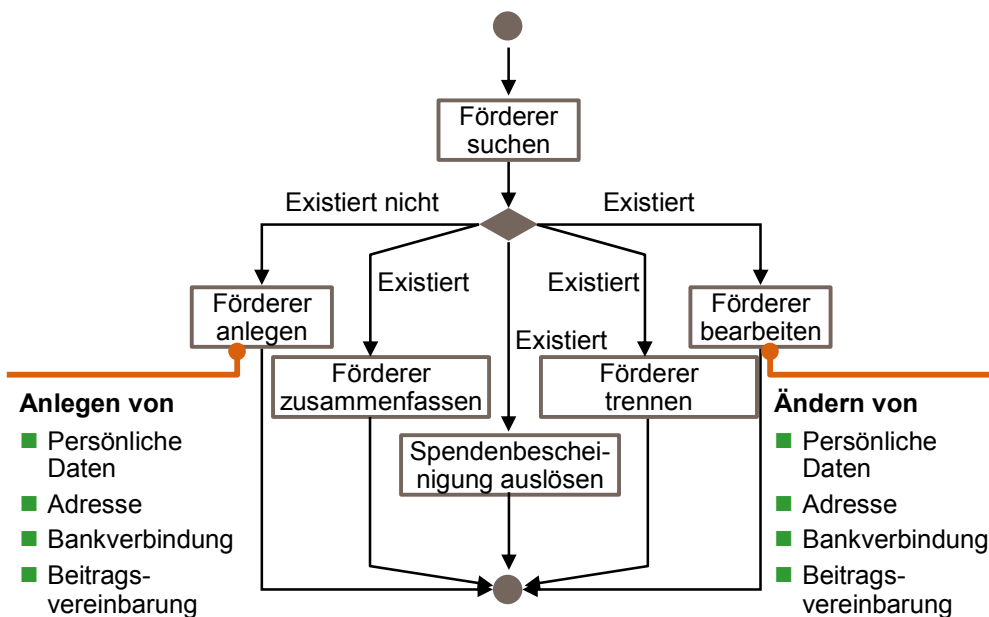
Für die UCP Methode ist der Umgang mit solchen aufzurufenden Use Cases relativ einfach folgendermaßen geregelt:

- der Aufruf des untergeordneten Use Cases und die Schaffung der Vorbedingungen für diesen zählt als ein Schritt im aufrufenden Use Case
- die Szenarien, Schritte und Dialoge des aufgerufenen Use Cases werden nur dort und nicht im aufrufenden Use Case gezählt (und damit über die gesamte Anwendung hinweg betrachtet auch nur einmal)
- eine Wiederverwendung nach Abschnitt 4.2 kommt bei den aufrufenden Use Cases nicht zur Anwendung

Eine extensive Verwendung von Aufrufstrukturen von Use Cases in einer Spezifikation kann ein Hinweis auf einen noch nicht passenden "Schnitt" der Use Cases für die Schätzung sein (muss es aber nicht!). Wir unterscheiden zwei Fälle:

- Es liegt eigentlich eine Beschreibung von Geschäftsprozessen auf höherer Ebene vor, die obere Ebene/Ebenen beschreiben keine einzelnen Use Cases, sondern eher organisatorische Prozesse oder ganze Workflows, die dann wiederum "echte" Use Cases verwenden. Diese sind dann in der Regel auch nicht mehr so stark verzahnt bzw. hierarchisch gegliedert. In diesem Fall können bei der Schätzung diese oberen Ebenen als Use Cases weggelassen werden und die eigentlichen Use Cases der weiteren Ebenen betrachtet werden. Allerdings sind dann auch die Hinweise bzgl. des Umganges mit Grobbeschreibungen aus 5.1 und 6.2 zu beachten.
- Der zweite (entgegen gesetzte) Fall ist eine bereits vorgenommene starke Dekomposition der Use Cases, viele der Use Cases werden also eher Schritte (in diesem Fall Anwendungsfunktionen) und keine Use Cases sein. Hier entsteht die Hierarchie und Verzahnung durch die Wiederverwendung von Funktionalität. In diesem Fall gelten die Hinweise aus 4.2

Abbildung 1 zeigt ein typisches Beispiel einer oft anzutreffenden Konstellation mit einem zentralen Use Case zum „Suchen“, von dem aus wiederum andere Use Cases aufgerufen werden. Dieser aufrufende Use Case führt selbst noch zu keinem fachlichen Ziel. Man könnte diesen ersten Teil auch als Schritt jedes einzelnen aufgerufenen Use Cases auffassen. Dann hätte man aber das Problem der Zählung der Wiederverwendung dieses ersten Schrittes und außerdem entspricht bei solchen komplexen Suchfunktionen die Gewichtung als eigener Use Case eher dem wirklichen Aufwand. In diesem Fall besteht der Use Case „Förderer Suchen“ neben den hier noch nicht ersichtlichen Schritten zum eigentlichen Suchen dann letztendlich aus den 5 Szenarien zum Aufrufen der Unter Use Cases und somit auch aus 5 zusätzlichen Schritten (neben dem Suchen).



**Abb. 1:** Beispiel für einen Use Case „Suchen“ als Ablaufdiagramm

### 3.4 Verwendung von Anwendungsfunktionen

Der Begriff Anwendungsfunktion soll hier nicht als getrennter Begriff oder Gegenkonzept zum Use Case eingeführt werden. Grundsätzlich ist eine Anwendungsfunktion ein Schritt eines Use Cases, hat also ein nicht so stark definiertes fachlich abgeschlossenes Ziel. Nichtsdestotrotz können solche Anwendungsfunktionen eine sehr hohe Komplexität haben, daher wollen wir hier den Umgang mit ihnen erläutern.

Wir unterscheiden dabei zwei Fälle:

#### 1. unabhängig von Use Cases spezifizierte Anwendungsfunktionen

Komplexere systemseitige Funktionen, vor allem wenn sie häufig wieder verwendet werden oder wenn die vorliegende Beschreibung bereits funktional zerlegt ist, sind in der Regel separat beschriebene Abläufe von fachlichen Berechnungen oder z.B. Prüfungen und Batchabläufe (siehe auch nächster Abschnitt).



Hier kann es sein, dass es sich um eine eigentlich in anderen Use Cases bereits verwendete aber nicht explizit referenzierte Funktionalität oder um eigenständige Funktionen handelt. Der erste Fall muss zuerst einmal erkannt werden und dann bei sehr komplexen Anwendungsfunktionen überlegt werden, ob es sich nicht eher um einen eigenen Use Case und damit um einen aufgerufenen Unter-Use Case entsprechend dem vorherigen Abschnitt handelt. Falls die beschriebene Funktion für sich alleine steht sollte sie als eigener Use Case in die Schätzung aufgenommen werden. Grundsätzlich soll die gesamte Funktionalität der Anwendung mit der UCP Methodik erfasst werden. In der Regel ist eine einzelne, fachlich komplexe Anwendungsfunktion immer auch als Use Case darstellbar.

## **2. *in Use Cases aufgerufene komplexe Anwendungsfunktionen***

Der zweite Fall sind bereits in Use Cases verwendete oder referenzierte Anwendungsfunktionen, die aufgrund ihrer Komplexität eigentlich zu groß für einen Schritt eines Use Cases sind. Dazu zählen z.B. in einem einzigen Schritt ausgeführte Prüfungen, danach komplexe Berechnungen und evtl. noch die Erzeugung/ Veränderung einer größeren Anzahl von Entitäten. In diese Kategorie fallen ebenso umfangreiche Regelwerke oder auch Funktions- und Entscheidungsmatrizen. In solchen Fällen schlagen wir eine Vergleichswichtung gegenüber anderen ("klassischen") Use Cases aus dem Projekt vor. Das bedeutet, dass die komplexen Schritte noch einmal aufgesplittet werden können in einzelne kleinere Schritte. Diese sollten vom fachlichen Umfang her den anderen (serverseitigen) Anwendungsfunktionen entsprechen und zumindest ein fachlich abgrenzbares Teilergebnis produzieren. Bei sehr komplexen Anwendungsfunktionen sollte überlegt werden, ob es sich nicht eher um einen eigenen Use Case und damit um den Aufruf eines Unter Use Cases entsprechend dem vorherigen Abschnitt handelt.

Anwendungsfunktionen werden häufig dann beschrieben, wenn es um Wiederverwendung geht. Es soll eine mehrfache Beschreibung gleicher Funktionalität in mehreren Use Cases vermieden werden. In solchen Fällen wird eine Anwendungsfunktion als einzelner Schritt nur einmal gezählt. Wird eine solche Anwendungsfunktion in mehreren Use Cases referenziert, soll sie nur bei einem dieser Use Cases (üblicherweise dem ersten) gezählt werden. In diesem Fall kann die Auslagerung in einen eigenen (Unter-) Use Case eine einfache Lösung sein.

Die Schwierigkeit liegt hierbei vor allem im Erkennen, dass eine Anwendungsfunktion bereits als Schritt in einem anderen Use Case gezählt wurde. Eine Möglichkeit der Berücksichtigung von Anwendungsfunktionen ist der Einsatz des Parameters "Wiederverwendung" in der UCP-Schätzmethodik, darauf wird im Abschnitt 4.2 noch einmal eingegangen.

## **3.5 Berücksichtigung der Batch-Funktionalität**

Häufig werden serverseitig komplexe Funktionen oder auch Batchabläufe, die ohne Interaktion mit anderen Abläufen oder Aktoren durchgeführt werden nicht als klassische Use Cases spezifiziert. Trotzdem handelt es sich oft um Use Cases, sie haben einen definierten Start mit Vorbedingungen (das ist häufiger sogar einfacher zu bestimmen als bei anderen Use Cases), es gibt Ablaufszenarien und ein bestimmtes fachliches Ziel.

Die Herausforderung besteht in diesen Fällen eher darin, den Ablauf für die Schätzung in einzelne Schritte der richtigen Größe zu zerlegen. Im Prinzip gelten hier alle Hinweise über die Schritte wie auch bei anderen Use Cases.

- wenn sehr komplex Zerlegung in Unter Use Cases entsprechend fachlicher Blöcke
- Schritte können ebenfalls gefunden werden anhand fachlicher Blöcke oder dem Zugriff auf Schnittstellen, anhand der verarbeiteten Entitäten
- ein einzelner Schritt kann hier eine geringfügig größere Mächtigkeit (=abgedeckte Funktionalität) haben als ein Schritt eines Nicht-Batch Use Cases, da es in der Regel keinen Aktor-Eingriff und damit keine komplexe Schnittstelle oder z.B. einen Schichtenwechsel innerhalb der Anwendungsarchitektur geben wird
- generell gilt, dass Schritte innerhalb eines Use Cases, über alle Use Cases einer Schätzung hinweg und (idealerweise) über alle UCP Schätzungen hinweg etwa gleich groß sein sollten.

Ist für einen Batch Use Case eine solche Zerlegung gefunden worden, wird er genauso gezählt wie ein "normaler" Use Case, d.h. es werden Szenarien, Schritte und Dialoge (hier wahrscheinlich eher in der Form von Druckstücken oder komplexen Ausgabedateien) erfasst.

**Fazit:** Das in den letzten beiden Abschnitten beschriebene Vorgehen bezüglich des Umganges mit komplexen Anwendungsfunktionen weicht zum Teil die Definition des Use Cases (fachliches Ziel wird erreicht) oder aber des Schrittes im Use Case auf (Abgrenzung zum nächsten Schritt ist nicht so deutlich vorhanden). Das gesamte Thema von komplexen Anwendungsfunktionen wird in zukünftigen Versionen der UCP Methodik noch einmal untersucht und eventuell als eigener Faktor in die Formel eingehen.

## **4 Praxisrelevante Durchführung einer UCP-Schätzung**

### **4.1 Erfassung der Use Cases**

Die so gefundenen Use Cases werden in einer für sd&m einheitlich vorgegeben Excel-Tabelle (Version 1.0) erfasst (siehe Tabelle 5). Sie wird an dieser Stelle kurz erläutert. Die Tabelle besteht aus weiteren Blättern zur Erfassung von Kostenfaktoren und natürlich dem Ergebnis der Schätzung, hier wird allerdings nur auf das Blatt zur Zählung und Dokumentation der Use Cases eingegangen.

Die Semantik der Felder ist dabei folgende:

- **Gruppe:** Anhand dieses Feldes können Use Cases gruppiert werden (in größere funktionale Gruppen). Es handelt sich lediglich um eine beschreibende Zusatzinformation.
- **Nr.:** Eine Nummer oder Referenz für den Use Case. Das Arbeitsblatt enthält eine fortlaufende Nummerierung, die durch projektspezifische Nummern oder Referenzen ersetzt werden kann.
- **Name:** Hier den Namen des Use Case eintragen.
- **Anzahl Szenarien:** Die Anzahl der unterschiedlichen Erfolgs-Szenarien und nichttrivialen Fehlerszenarien im Use Case. Es werden hierbei nicht alle möglichen Kombinationen von Szenarien gezählt, sondern eine Aufsummierung der Szenarien über den gesamten Ablauf hinweg
- **Anzahl Schritte:** Die Anzahl der einzelnen unterschiedlichen Schritte im Use Case in allen Szenarien.
- **Anzahl Dialoge:** Die Anzahl der unterschiedlichen Dialoge/Masken im Use-Case in allen Szenarien

Use-Case-Beschreibung			Maße			Use-Case-Points			Re-Use in %	Bemerkung
Gruppe	Nr	Name	Anzahl Szenarien	Anzahl Schritte	Anzahl Dialoge	Be-rechnet	Intuitiv	Effektiv		
<b>Summen:</b>			<b>31</b>	<b>72</b>	<b>45</b>	<b>145</b>		<b>125,0</b>		
BaufiLight	1	Kooperations-partner verwalten	2	3	3	5		5,0		
	2	BaufiLight aufrufen	1	2	2	5		5,0		
	3	Kunden und Anfragen verwalten	3	9	2	15		15,0		
	4	Dokumentenerzeugung und Versand aus Baufi Light	1	4	1	10		10,0		
Haushaltsrechnung	5	Haushaltsrechnung verwalten	1	3	1	5		5,0	Lesen, ändern, speichern, Vorschau ist nächster Use-Case	
	6	Haushaltsrechnung darstellen	3	4	3	10		10,0	Preview, echte Web-Ansicht, Druckansicht	

**Tab. 5:** Beispiel einer UCP-Zählung

Auch wenn es für die Berechnung des Gewichtes eines Use Cases nicht zwingend erforderlich ist, wird dringend empfohlen, bei einer konkreten Schätzung alle zur Verfügung stehenden oder ermittelbaren Werte einzutragen.

Aus diesen Informationen wird dann mit folgender Formel das Gewicht des Use Cases berechnet und in der Spalte *berechnet* angezeigt:

$$UCP = MIN (15, ((MAX ( ANZ_{Schr}, ANZ_{Sz}, ANZ_{Dia} ) DIV 5) +1 ) * 3)$$

wobei UCP = Gewicht des Use Cases  
ANZ<sub>Schr</sub> = Anzahl der Schritte  
ANZ<sub>Sz</sub> = Anzahl der Szenarien  
ANZ<sub>Dia</sub> = Anzahl der Dialoge

Damit ergibt sich eine diskrete Einteilung der Use Cases in die 3 Kategorien leicht (5), mittel (10) und komplex (15).

An dieser Stelle soll nur eine kurze Erläuterung und Motivation dieser Kategorisierung gegeben werden. Eine Formel, die die Anzahl der erfassten Kenngrößen kontinuierlich berücksichtigt, würde eine Genauigkeit suggerieren, die nicht vorhanden ist. Etwaige Fehler in der Zählung z.B. von Schritten, werden über eine gewisse Anzahl von Use Cases statistisch gesehen wieder ausgeglichen. Die Beschränkung auf eine gewisse maximale Größe von Use Cases soll bei der Schätzung einen Hinweis darauf liefern, das die gewählte Granularität in diesen Fällen evtl. nicht mit der sonst gewählten Dimensionierung übereinstimmt. Beide Beschränkungen dieses Modells können in zukünftigen Stufen jedoch angepasst und modifiziert werden.

**Fazit:** Aus der Formel ist auch ersichtlich, dass die Anzahl der Schritte, Szenarien und Dialoge gleichwertig in die Berechnung eingeht. Das führt momentan in der Version 1.0 dazu, dass in der Regel bei einer konkreten Schätzung die Anzahl der Schritte für das Gewicht des Use Cases ausschlaggebend sind, da die beiden anderen Größen normalerweise kleiner sind. Diese Gewichtung kann und wird in zukünftigen Versionen der UCP Methodik noch mittels der dann erweiterten Datenbasis angepasst werden. Momentan ist es daher wichtig, auf jeden Fall die Schritte korrekt zu erfassen, da hierfür die Formel bereits justiert ist.

Zusätzlich können in dem Blatt die berechneten Werte überstimmt werden, indem in der Spalte *intuitiv* ein Wert eingetragen wird. Dieser wird dann anstelle des berechneten Wertes verwendet. Allerdings ist auch hier nur eine Angabe der schon erwähnten Werte für leicht, mittel und schwer möglich.

Das Gewicht eines Use Cases kann außerdem durch die Angabe eines bestimmten Grades von Wiederverwendung in der Spalte Re-Use beeinflusst (verkleinert) werden. Es handelt sich dabei um eine prozentuale Angabe, wie viel Funktionalität dieses Use Cases nicht mehr neu entwickelt werden muss.

Auf die Verwendung des Überstimmens und der Wiederverwendung wird im nächsten Abschnitt näher eingegangen.

In der Spalte effektiv wird dann der in die Schätzung eingehende Wert angezeigt.

Bei der Erfassung der Use Cases sollten alle beschriebenen auch in die Schätzung aufgenommen werden.

Das gilt auch für solche, die aus verschiedenen Gründen nicht bewertet wurden (z.B. nicht umzusetzen, nicht schätzbar oder 100% Bestandteil eines anderen Use Cases).

Dieses Vorgehen sichert eine Vergleichbarkeit verschiedener UCP Schätzungen untereinander oder mit einer Expertenschätzung.

## 4.2 Reuse und intuitive Äquivalenz

Für die Bewertung der Use Cases hat der Schätzer neben der Erfassung der Szenarien, Schritte und Dialoge noch weitere Möglichkeiten, das Gewicht eines Use Cases zu beeinflussen.

Zum einen kann er manuell (= intuitiv) den nach oben angegebener Formel berechneten und im Excel-Tabelle auch angezeigten Wert überstimmen, indem er in die entsprechende Spalte „intuitiv“ in der Excel-Tabelle einen Wert einträgt. Dies wird und soll der Schätzer insbesondere dann tun, wenn:

- er nicht genug Informationen hat, um die Parameter Szenarien, Schritte und Dialoge quantitativ genau zu bestimmen
- ein Use Cases zwar nach der Formel korrekt berechnet vorliegt, im Vergleich zu anderen gleich berechneten Use Cases jedoch die Funktionalität der Szenarien, Schritte, Dialoge deutlich komplexer oder einfacher ist

Dieses "Überstimmen" ist ein durchaus legitimes Mittel und gibt insbesondere dem erfahrenen Schätzer eine gute Möglichkeit, korrigierend einzugreifen. Allerdings sollte diese Entscheidung auch in der Bemerkungsspalte dokumentiert werden.

Tritt das zweite Kriterium allerdings im Verlaufe einer Schätzung sehr häufig auf (> 30%) so liegt auch hier ein Hinweis auf einen nicht korrekten "Schnitt" der Use Cases oder aber der Schritte innerhalb der Use Cases vor. Dies gilt insbesondere dann, wenn die Korrekturen immer in die gleiche Richtung (höher oder niedriger) erfolgen. Auch wenn es einfacher (und schneller) ist, einen Wert zu überschreiben sind die Schätzer aufgefordert, in diesem Fall den Mehraufwand einer Korrektur des Schnittes der Use Cases auf sich zu nehmen. Dadurch ist die Schätzung einfacher nachvollziehbar und damit auch besser zu validieren.

Zum zweiten sieht unsere UCP Schätzmethodik den Einsatz von Re-Use, also die Erfassung von wiederverwendeter Funktionalität vor. Hier kann der Schätzer den prozentualen Anteil des Use Cases angeben, dessen Funktionalität bereits anderweitig abgedeckt ist und damit nicht mehr umgesetzt und getestet werden muss.

Damit ist insbesondere gemeint:

- der Use Case wird im Projekt nur erweitert, nicht vollständig neu entwickelt, dieser Fall tritt insbesondere bei Folge- oder Wartungsprojekten häufig ein.
- der Use Case wird fachlich zum Teil bereits durch eine Produktsoftware oder ein Framework realisiert.

- der Use-Case ist eine Spezialisierung (im OO-Sinne) eines anderen Use Cases.
- bestimmte Schritte des Use Cases (speziell wenn es sich um Anwendungsfunktionen handelt) werden bereits in anderen Use Cases verwendet.

Die Abschätzung des Grades der Wiederverwendung ist in der Regel nicht einfach. Ein hoher Anteil an mit Re-Use geschätzten Use Cases ist ebenfalls ein Hinweis auf eine potentiell kritische Schätzung. Zu berücksichtigen ist ebenfalls, dass sich Re-Use auf alle Phasen beziehen muss. Falls es nur um die reine Realisierung geht, die wiederverwendete Funktionalität aber getrennt spezifiziert und getestet werden muss, ist der Grad der anzusetzenden Wiederverwendung geringer.

Speziell das letzte angeführte Kriterium kann in Schätzungen problematisch werden, wenn die Spezifikation bereits einen hohen Detaillierungsgrad aufweist. Im Falle einer starken "Verzahnung" der Use Cases und damit auch einer gehäuften Wiederverwendung von Anwendungsfunktionalität empfehlen wir, diese Schritte dann nur einfach beim ersten verwendenden Use Case zu zählen und nicht über den Parameter Re-Use zu steuern.

**Beispiel:** Eine Anwendungsfunktion "Kunde suchen" wird in vielen verschiedenen Use Cases als Schritt verwendet. Für die Schätzung gibt es jetzt verschiedene Möglichkeiten:

- es handelt sich um einen eigenen Use Case (eigener Dialog, mehrere Szenarien, Schritte), in diesem Fall wird es auch als eigener Use Case geschätzt und in den anderen als verwendetet Use Case berücksichtigt (siehe auch ...)
- es ist kein eigener Use Case und der Schritt "Kunde suchen" wird beim ersten verwendenden Use Case gezählt, in den weiteren wird er als Re-Use prozentual abgezogen (z.B. ein Use Case mit 5 Schritten, von denen dann einer die Suche ist, hat dann 20% Re-Use, einer mit 4 Schritten 25%). Dieses Verfahren kann für einzelne Fälle schnell angewendet werden, bei häufigem Auftreten ist es aber sehr mühsam und bei weiteren wieder verwendeten Anwendungsfunktionen besteht die Gefahr der Unübersichtlichkeit
- es ist kein eigener Use Case und der Schritt "Kunde suchen" wird beim ersten verwendenden Use Case gezählt, in den weiteren wird dieser Schritt nicht mehr gezählt (z.B. ein Use Case mit 5 Schritten, von denen dann einer die Suche ist, hat dann noch 4 gezählte Schritte). Dieses Verfahren hat sich in der Praxis bewährt und wird bei Wiederverwendung einzelner Schritte empfohlen. Es entspricht auch dem Vorgehen, wenn innerhalb eines Use Cases Schritte mehrfach verwendet werden

Der Faktor Re-Use geht immer nach allen berechneten und auch intuitiv ermittelten Werten in die Schätzung ein. Ein nach bereits genannter Formel berechneter Use Case mit 10 UCP und einer Wiederverwendung von 20% wird also mit 8 UCP in die Schätzung eingehen. Werden die berechneten 10 UCP zusätzlich mit 5 UCP intuitiv überstimmt, bleiben im Fall einer weiter eingetragenen 20% Wiederverwendung also noch 4 UCP übrig. Das bedeutet: Wiederverwendung geht grundsätzlich in die Schätzung ein, durch intuitives Überstimmen wird die Wiederverwendung nicht überschrieben!

### 4.3 Erfassung von Aktoren für die Schätzung

Aktoren werden analog zu den Use Cases auf einem gesonderten Blatt der Tabelle erfasst. Wie bei den Use Cases können hierbei Gruppierungen vorgenommen, laufende Nummern vergeben und Bezeichnungen eingetragen werden (siehe Beispiel in Tabelle 6).

Anschließend werden die Aktoren einer bestimmten Komplexitätsklasse zugeordnet, wobei nur folgende unterschieden werden:

- einfach, z.B. ein Randsystem das über eine (einfache) Schnittstelle angebunden ist
- mittel, z.B. ein Randsystem das über eine (komplexeres) Protokoll angebunden ist
- komplex, z.B. ein Endnutzer, der über ein Frontend mit dem System agiert

Aktoren					
Aktoren-Beschreibung			Bewertung		Bemerkung
Gruppe	Nr.	Name	Komplexität	Punkte	
<b>Summe:</b>				<b>16</b>	
	1	Regeladmin	komplex (Endnutzer über Frontend)	3	
	2	Finanzierungsberater	komplex (Endnutzer über Frontend)	3	
	3	CallCenter-Mitarbeiter	komplex (Endnutzer über Frontend)	3	
	4	Mitarbeiter der Kooperationspartner	komplex (Endnutzer über Frontend)	3	
	5	Comod Schnittstelle	einfach (System über Schnittstelle)	1	
	6	Bankenschnittstelle	einfach (System über Schnittstelle)	1	

**Tab. 6:** Beispiel einer Aktorenerfassung bei der UCP-Schätzung

Endnutzer mit fachlich unterschiedlichen Rollen, werden als separate Aktoren gezählt (wie in der OO-Analyse üblich).

Bei der Erfassung der Aktoren kommen im Prinzip folgende Varianten vor:

- es liegt bereits eine Liste aller Aktoren als Zusammenfassung der Spezifikation bei, in diesem Fall kann diese direkt übernommen werden.
- für jeden einzelnen Use Case sind Aktoren angegeben. In diesem Fall muss darauf geachtet werden, dass gleichartige Aktoren zusammengefasst werden. Es kann allerdings vorkommen, dass gleich bezeichnete Aktoren in unterschiedlichen Use Cases als getrennte Aktoren betrachtet werden müssen, so kann z.B. ein Administrator im Kontext eines bestimmten Use Cases eine andere Rolle darstellen als in einem anderen Use Case.
- Aktoren sind nicht angegeben. In diesem Fall sind sie im Laufe der Use Case Zählung "aufzusammeln" und analog wie bei der Variante mit der einzelnen Beschreibung einzuordnen.

## 5 Spezifika der Use-Case-Beschreibungsarten für die UCP-Methode

Für die Dokumentation von Use Cases gibt es keine einheitlich definierten formalen Vorgaben. Es haben sich allerdings im Laufe der Zeit und unter anderem auch bei sd&m einige Formen bewährt und durchgesetzt. Im Folgenden wird erläutert, wie aus diesen verschiedenen Arten der Beschreibungen eine Zählung für die UCP Schätzung gefunden werden kann.

### 5.1 Grobe Use-Case-Beschreibungen

In der Praxis kann es häufig vorkommen, dass eine Use Case Beschreibung erst in einer sehr groben Form vorliegt. Das wird in der Regel in frühen Projektphasen z.B. als Lastenheft, Vorstudie oder speziell für eine Ausschreibung erstellte Unterlagen der Fall sein. In diesen Fällen kann z.B. eine Übersicht von Use Cases mit einer kurzen textuellen Beschreibung oder eine Geschäftsprozessbeschreibung inklusive der Benennung der Use Cases vorhanden sein.

Die so benannten Use Cases geben bereits eine gute Basis für die Schätzung nach UCP. Sie können als erster Schritt direkt in die Schätztabelle übernommen werden. Im zweiten Schritt ist dann je nach Detaillierung der Schätzung eine Gewichtung nur nach intuitiver Einschätzung möglich oder es können aus der textuellen Beschreibung Hinweise auf die Schritte entnommen werden. In diesem Fall sind die Kriterien aus Abschnitt 3.1 zu beachten.

Liegt eine Grobspezifikation ohne Benennung von Use Cases vor, ist der erste Schritt auf jeden Fall das Finden und Aufzählen der Use Cases selbst. Je nach Detaillierung der Spezifikation kann dabei eine der im Abschnitt 6 beschriebenen Mapping-Methoden helfen. Häufig wird der zweite Schritt aber in der Durchführung einer intuitiven Schätzung bestehen.

Ähnlich wie bei einer Expertenschätzung gibt es auch hier einen Zustand, in dem zu wenig Information für eine Schätzung vorliegt und somit eine Schätzung nicht mehr möglich ist.

Besteht zumindest Konsens oder eine gewisse Sicherheit, dass alle Use Cases gefunden und damit benannt und gezählt werden können und die Szenarien und Schritte können nicht benannt werden, so kann ein erster grober Entwurf durchaus in einer manuellen durchschnittlich mittleren Wichtung der Use Cases liegen. In der Regel hat man damit eine Obergrenze. Dieser Wert hat sich vor allem bei kleineren bis mittleren Projekten (bis ca. 50 Use Cases) als relativ treffend erwiesen.

Bei mehr Use Cases tritt in der Regel ein gewisser Effekt von Wiederverwendung von Funktionalität ein, so dass der einzelne Use Case eher geringer gewichtet werden kann (ca. in der Mitte zwischen leicht und mittel für eine Obergrenze). Bei noch stärkerer Wiederverwendung von fachlicher Funktionalität in unterschiedlichen Use Cases sind dann die einzelnen Use Cases im Durchschnitt eher leicht.



## 5.2 Textuelle oder tabellarische Beschreibungen

Eine textuelle oder tabellarische Beschreibung enthält in der Regel die folgenden Informationen bzw. mindestens eine Untermenge davon:

Titel, Kurzbeschreibung, Ziel, Aktoren, Auslöser, Vorbedingung, Szenarien, Ergebnis, Ausführungshäufigkeit, NFA, Bemerkungen

Diese Form der Beschreibung in mehr oder weniger starker Anlehnung an die genannte Strukturierung der Informationen wird zum Zeitpunkt einer Aufwandsschätzung nach UCP die am häufigsten anzutreffende sein bzw. als Ergänzung zu einer der anderen später genannten Spezifikationsformen auftreten.

Aus Sicht der UCP Methodik liegt damit schon eine sehr gute Basis für eine Schätzung vor. In diesem Fall kann ein sehr einfaches Mapping in die Schätztabelle erfolgen.

- Bereits vorhandene fachliche Gruppierungen (z.B. Kapitel) können direkt in die Gruppenspalte übernommen werden
- Titel oder Kurzbeschreibung können direkt in die Namensspalte übernommen werden
- Die Anzahl der Szenarien kann aus der Beschreibung der Szenarien entnommen werden
- Die Anzahl der Schritte ist als Summe der Schritte (Aktionen) aus den einzelnen Szenarien zu betrachten und kann damit aus der Beschreibung der Szenarien entnommen werden.

Zur Evaluierung kann bei einem angegeben Ziel in der Szenariobeschreibung überprüft werden, ob zumindest das Hauptszenario komplett beschrieben ist. Eventuell aufgeführte Hinweise auf Anwendungsfunktionen oder Referenzen auf andere Use Cases helfen bei der Erkennung von Wiederverwendung. Sollten diese Referenzen nicht explizit vorhanden sein, liegt es an den Fähigkeiten des Schätzers, sich an diese zu erinnern und sie zu erkennen. Auch hier sei mit Tabelle 7 ein Beispiel angegeben.

<i>Titel</i>	AF_LeistungBuchen	
<i>Kurzbeschreibung</i>	Der Use Case "AF_LeistungBuchen" der Komponente Buchung führt für eine Reservierungsliste die Buchungen durch. Im Use Case "Buchung::AF_LeistungReservieren" wurde durch den Sachbearbeiter eine Reservierungsliste mit Reisen erstellt, die gebucht werden können, also vakant sind. Der Use Case "AF_LeistungBuchen" führt jetzt für alle Reservierungen eine Buchung durch.	
<i>Ziel</i>	Durchführen einer Buchung	
<i>Szenarien</i>	Leistung buchen (Erfolgsszenario)	<ol style="list-style-type: none"> <li>1. Der Anwender wählt eine Reservierungsliste aus (Use Case "Buchung::Reservierungsliste suchen")</li> <li>2. Der Anwender wählt alle Leistungen aus, die gebucht werden sollen.</li> <li>3. Das System führt für alle gewählten Leistungen eine Buchung durch. <ol style="list-style-type: none"> <li>3.1 Zuerst geprüft das System, ob für die Leistung ein Kontingent vorliegt. <ol style="list-style-type: none"> <li>3.1.1 Liegt ein Kontingent vor, so bucht das System von diesem Kontingent die Leistung.</li> <li>3.1.2 Wenn kein Kontingent vorliegt, bucht das System die Leistung direkt über das Buchungssystem des Leistungsanbieters.</li> </ol> </li> </ol> </li> </ol>
	Kontingent erschöpft	<ol style="list-style-type: none"> <li>1 Ist das Kontingent erschöpft, wird der Anwender gefragt, ob eine direkte Buchung durchführen möchte. <ol style="list-style-type: none"> <li>1.1 Will der Anwender eine direkte Buchung durchführen, wird mit Punkt 3.1.2 fortgefahren.</li> <li>1.2 Will der Anwender keine direkte Buchung durchführen, wird die Buchung mit einer Fehlermeldung abgebrochen.</li> </ol> </li> </ol>
	Buchung nicht möglich	<ol style="list-style-type: none"> <li>1 Wenn eine Buchung nicht möglich ist, weil die Leistung nicht verfügbar ist, wird die Buchung mit einer entsprechenden Meldung abgebrochen.</li> </ol>
<i>Ergebnis</i>	Wurde die Buchung erfolgreich durchgeführt, ist die Buchung am Ende geschlossen Es werden keine Daten zurückgegeben	

**Tab. 7:** Beispiel einer tabellarischen Use-Case-Beschreibung

Interessant ist in diesem Beispiel, dass der beschriebene Use Case ein Unter Use Case von Reservierungsliste suchen ist. Dies ist dann sinnvoll, wenn mit der gefundenen Reservierungsliste außer der Buchung auch noch verschiedene andere fachliche Use Cases durchgeführt werden können. Ansonsten wäre das Suchen der Reservierungsliste besser als erster Schritt für den hier beschriebenen Use Case modelliert und damit auch gezählt worden. Bei der hier vorgegebenen Modellierung würde man das Auswählen einer Reservierungsliste noch zum aufrufenden Use Case zählen.

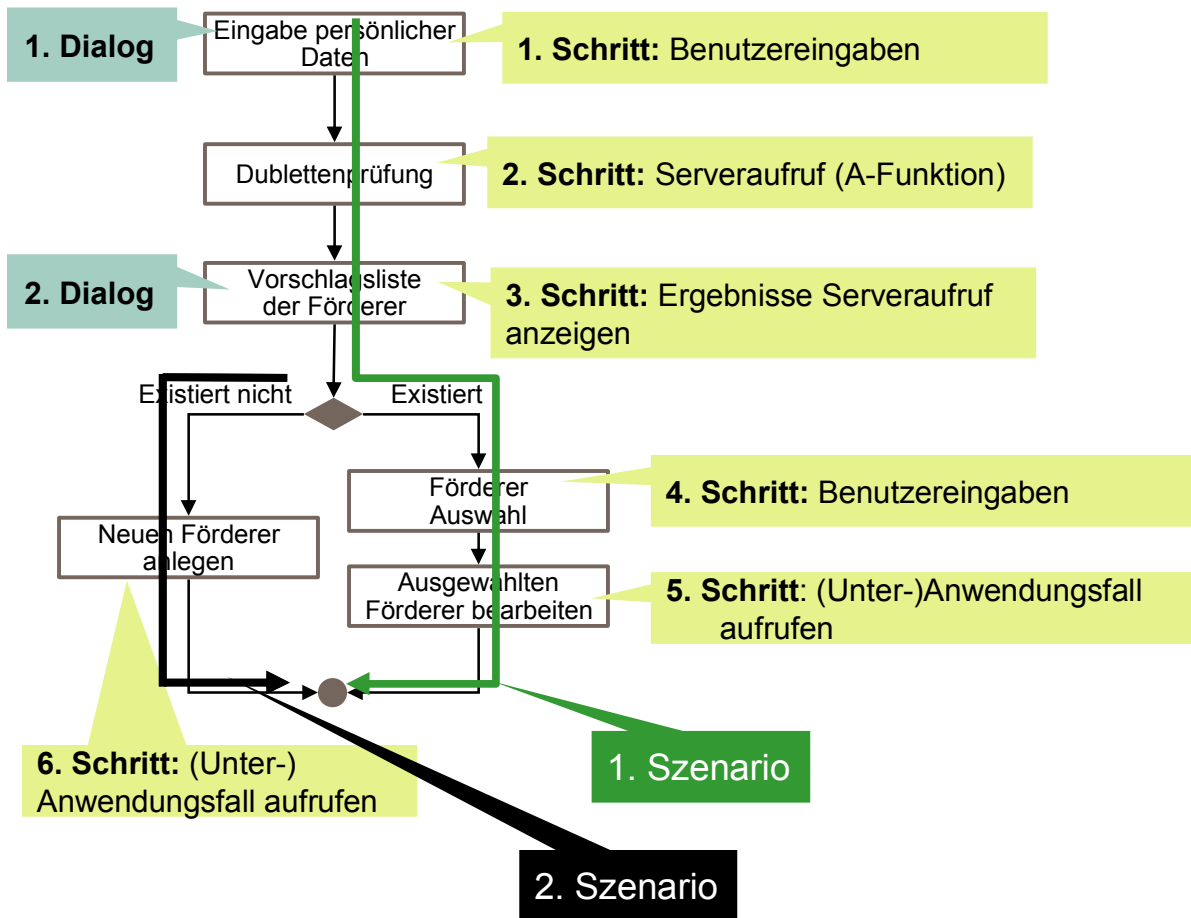
Damit ist der erste Schritt die Auswahl der zu buchenden Leistungen, der zweite Schritt ist die Prüfung auf ein vorliegendes Kontingent und der dritte die eigentliche Buchung. Das (Haupt-) Erfolgsszenario besteht also aus 3 Schritten. Alternativ (2. Szenario) erfolgt bei nicht vorliegendem Kontingent eine direkte Buchung über das Buchungssystem des Leistungsanbieters (4. Schritt). Außerdem gibt es ein 3. Szenario, bei dem ein an sich vorhandenes Kontingent ausgeschöpft ist und eine Abfrage erfolgt, ob eine direkte Buchung erfolgen soll (5. Schritt). Die darauf folgende Buchung ist bereits gezählt worden, ebenso werden die Fehlermeldungen über nicht buchbare Leistungen als trivial eingestuft. Wir zählen also insgesamt 3 Szenarien mit 5 Schritten und 1 Dialog (Auswahl der Leistungen; das Pop-Up für die Abfrage, ob direkt gebucht werden soll, zählt nicht).

### 5.3 UML-basierte Ablauf- bzw. Aktivitätendiagramme

Liegen Ablauf- oder Aktivitätendiagramme als Form der Use Case-Beschreibung vor, können hieraus sehr schnell und einfach direkt die erforderlichen Informationen für eine UCP Schätzung entnommen werden. An dieser Stelle betrachten wir ausschließlich Aktivitätendiagramme für Use Cases. Auf die Beschreibung von Geschäftsprozessen wird in 6.2 noch eingegangen. Es wird allerdings empfohlen, den richtigen "Schnitt" stichprobenartig anhand der zusätzlichen Beschreibung zu überprüfen. In diesen Diagrammen werden normalerweise Schritte und Dialoganzeigen in unterschiedlichen Symbolen dargestellt und gerichtete Kanten kennzeichnen den fachlichen Ablauf des Use Cases. Dabei sind folgende Hinweise zu beachten:

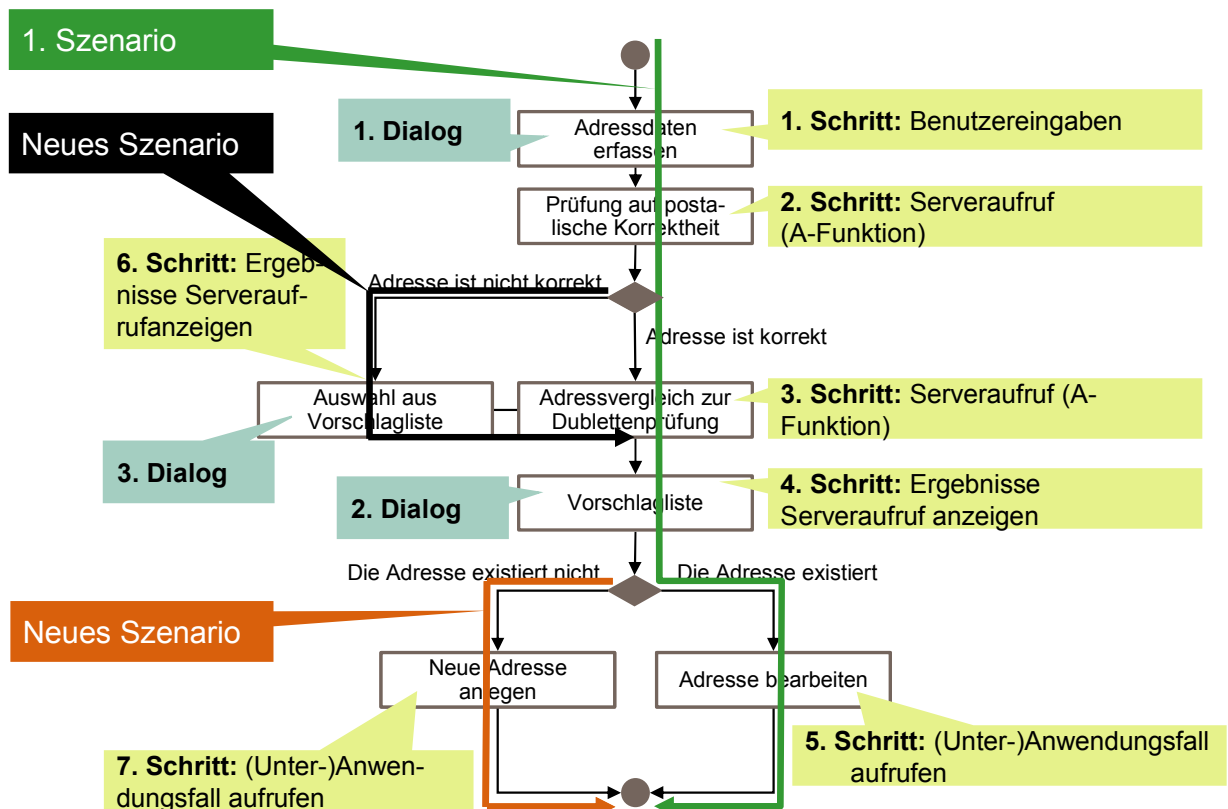
- Anzeigen (Dialoge) sind auch als Schritt zu zählen
- Entscheidungsalternativen werden manchmal nicht gesondert als Schritt dargestellt wenn sie systemseitig erfolgen, können aber durchaus fachlich komplexe Abläufe beinhalten, die eine Zählung als eigenen Schritt rechtfertigen.
- die Aufspaltung bei einer Alternative führt zu neuen Szenarien entsprechend der Anzahl der neuen Verzweigungen (keine Kombinationen mit vorherigen Szenarien).
- Wenn ein Algorithmus (z.B. ein Zuordnungsalgorithmus) als Aktivitätendiagramm dargestellt ist, handelt es sich bereits um eine starke funktionale Dekomposition. In diesem Fall sind die Verzweigungen und Schritte möglichst weitgehend zusammenzufassen, bevor die Zählung beginnt.
- Triviale Fehlerszenarien können als eigene Szenarien mit Schritten dargestellt sein, sollten aber nicht als solche gezählt werden

In den folgenden Beispielen soll dieses Vorgehen veranschaulicht werden. Die gezählten Szenarien, Dialoge und Schritte sind dabei farblich unterschiedlich dargestellt und nummeriert.



**Abb. 2:** Aktivitätsdiagrammbezogene Interpretation

Zu beachten ist hierbei insbesondere, dass der Aufruf des Unter-Use Cases als ein einzelner eigener Schritt gezählt wird, der eigentliche Ablauf des Unter-Use Cases in die Zählung aber nicht eingeht. Die Entscheidung durch den Benutzer, ob ein neuer Förderer angelegt werden soll oder ein bestehender bearbeitet wird eröffnet ein neues Szenario, ist aber an sich kein eigener Schritt.



**Abb. 3:** Erweiterte Aktivitätsdiagrammbezogene Interpretation

In diesem Beispiel werden die Schritte - wie bereits beschrieben - ganz normal gezählt, hierbei sind die Dialoganzeigen als eigene Schritte mitgezählt. Ebenso ist die Prüfung der Adresse auf postalische Korrektheit ein eigener Schritt. In diesem Fall ein eigener fachlicher Ablauf, da bei nicht korrekter Adresse ein neues Szenario mit Benutzerinteraktion ausgelöst wird. Im Gegensatz dazu wird die Entscheidung, das eine neue Adresse angelegt oder eine bestehende bearbeitet wird, nicht als eigener Schritt gezählt. Die Anzahl der Szenarien ergibt sich als Aufsummierung aller Alternativen zum Hauptablauf über den Gesamtablauf hinweg plus den Hauptablauf. In diesem Fall gibt es eine Alternative nach dem Schritt der Prüfung auf postalische Korrektheit und eine Alternative bei der Entscheidung, ob eine Adresse existiert oder nicht. Damit haben wir insgesamt zweimal Alternativen und mit dem Hauptszenario 3 Szenarien.

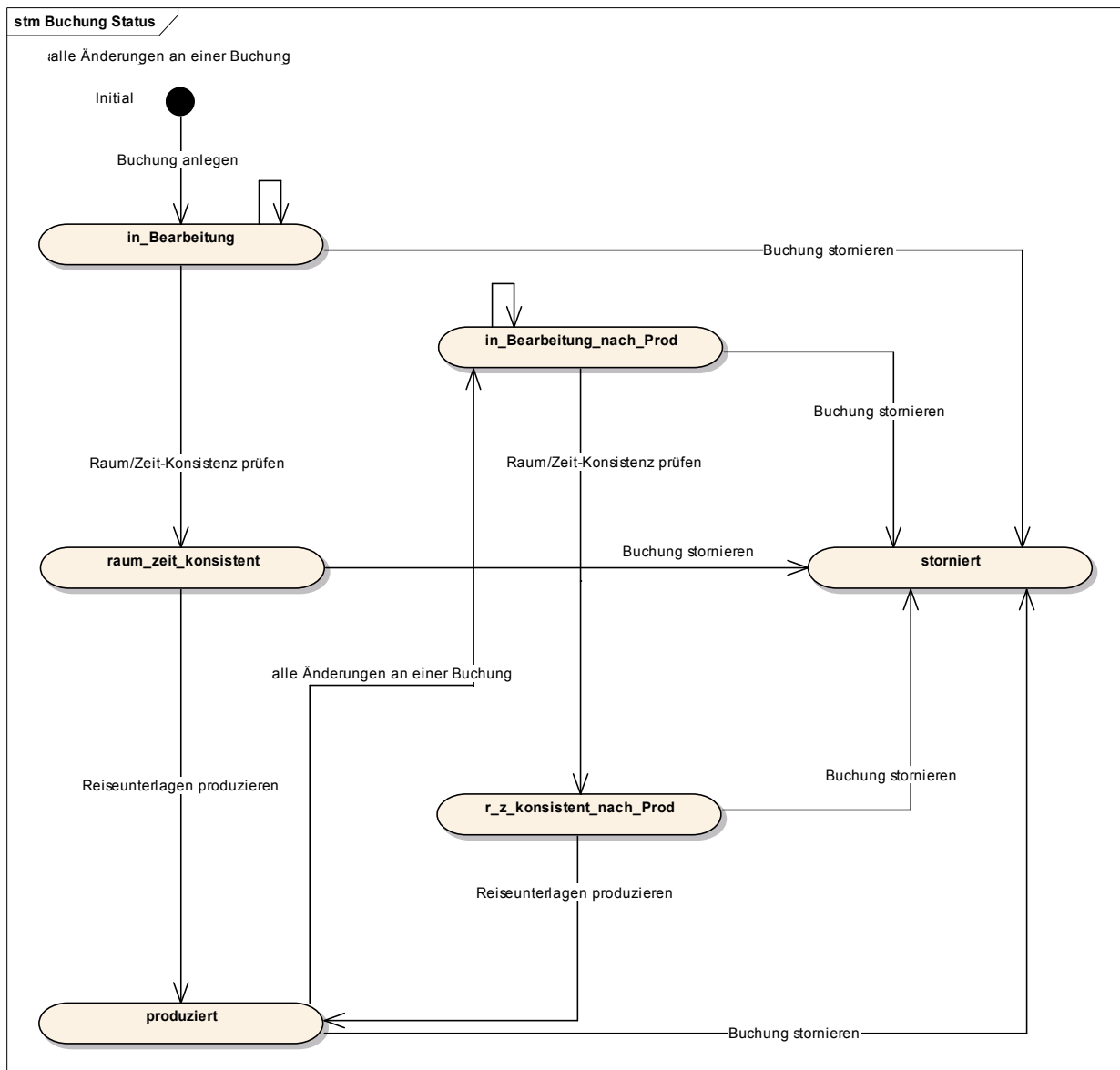
#### 5.4 Interpretation von Zustandsdiagrammen

Bei bestimmten Anwendungen spiegelt sich ein großer Teil der fachlichen Komplexität im Lebenszyklus eines Objektes wieder. In diesem Fall durchlaufen diese Entitäten im Laufe ihrer Existenz im System verschiedene Zustände und Zustandswechsel.

Handelt es sich dabei um fachlich relevante Vorgänge, kann man sie mit Hilfe von Zustandsdiagrammen beschreiben und dokumentieren.

In der Regel werden Zustandsdiagramme nicht isoliert vorkommen, trotzdem können auch aus Ihnen relevante Informationen für die UCP Schätzung gefunden werden.

Zustandsdiagramme decken üblicherweise mehrere Use Cases, diese aber nicht zwangsläufig vollständig ab, stellen also einen anderen "Schnitt" durch die Anwendung dar. Bei einer vollständigen Abdeckung aller Entitäten und ihrer Zustandsübergänge (insbesondere auch der trivialen) sollten auch alle Use Cases abgedeckt sein. Dies soll hier am Beispiel eines Zustandsdiagramms erläutert werden.



**Abb. 4:** Beispiel für Use Case bezogenes Zustandsdiagramm

Die Zustände sind hierbei durch Ellipsen und die Zustandsübergänge durch Kanten (Pfeile) zwischen diesen dargestellt. Die Kanten des Diagramms sind entweder beschriftet oder textuell beschrieben. Hier finden sich dann die Use Cases oder Schritte von Use Cases.

Im Beispiel in Abbildung 4 ergeben sich so mindestens die Use Cases Buchung anlegen, Buchung ändern und Buchung stornieren. Der Use Case Buchung anlegen beinhaltet mit Sicherheit die Eingabe von Daten und in diesem Beispiel laut Diagramm auch den Schritt Raum/Zeit Konsistenz prüfen. Aus dem Diagramm alleine ist nicht ersichtlich, ob „Reiseunterlagen produzieren“ ein eigener Use Case oder auch ein Schritt von „Buchung anlegen“ ist. In obigem Beispiel liefert das Zustandsdiagramm außerdem einen Hinweis darauf, das es für den Use Case Buchung stornieren in Abhängigkeit vom Status mehrere Erfolgsszenarien gibt (in diesem Fall 5 verschiedene aufgrund 5 verschiedener möglicher Ausgangszustände). Je nach Komplexität der durchzuführenden fachlichen Abläufe sind das dann auch unterschiedliche Schritte im Use Case.

Eine UCP Schätzung ausschließlich auf Basis von Zustandsdiagrammen wird häufig nur einen groben Hinweis auf die Aufteilung der Anwendung in Use Cases ermöglichen. Die zu zählenden Schritte können und müssen dann ergänzend aus dem Text gefunden werden oder es erfolgt eine intuitive Schätzung.

Bei einer Schätzung von UCP anhand von Zustandsdiagrammen sollte auf jeden Fall darauf geachtet werden, das triviale Zustände und ihre Übergänge nicht übersehen werden, da diese in der Regel nicht in der Form von Diagrammen spezifiziert sind, durchaus aber eigene Use Cases sein können. Dazu gehören z.B. das simple "Erschaffen" eines Objektes und dessen spätere Löschung, d.h. es existiert nur ein Zustand.

## 5.5 CRUD Diagramme

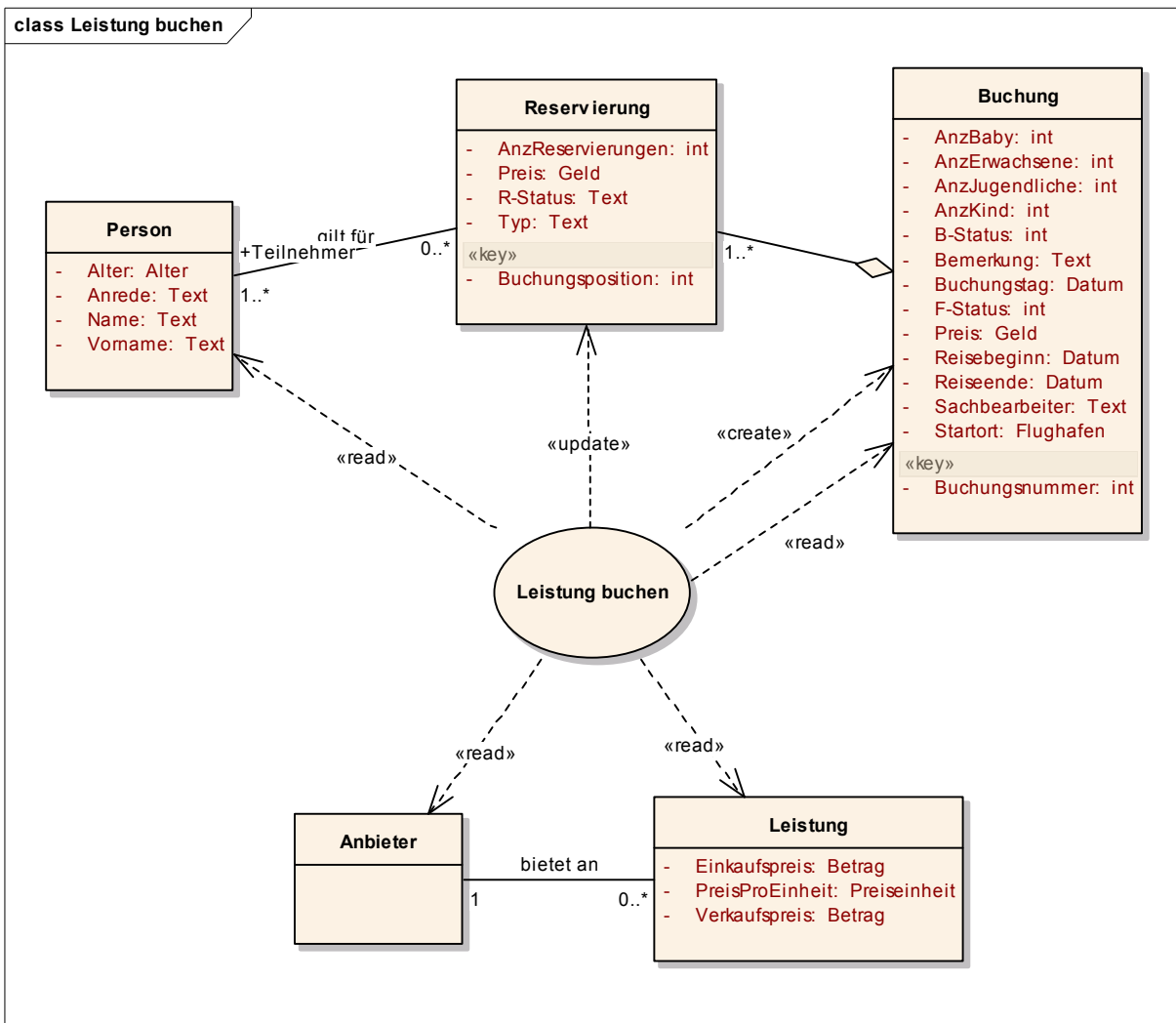
Auch CRUD (Create/Read/Update/Delete) Diagramme als eine spezielle Form der Beschreibung von Use Cases liefern eine Basis für eine Schätzung nach UCP. In einem solchen Diagramm ist dokumentiert, welche Lese- und Schreibzugriffe ein Use Case direkt auf die Entitäten des fachlichen Datenmodells ausführt. Damit gehen die Use Cases selbst direkt aus diesen Diagrammen hervor (sie stehen in der Regel im Mittelpunkt). Auch hier soll die Verwendung für die UCP Schätzung an einem Beispiel (Abbildung 5) vorgenommen werden.

Die Kanten geben dann Auskunft darüber, welche Entitäten im Zusammenhang mit diesem Use Case auf welche Art und Weise verändert werden.

Leider kann in diesem Fall nicht direkt aus der Anzahl der Kanten auf die Anzahl der Schritte des Use Cases geschlossen werden! Es fehlen die Hinweise auf Interaktionen mit den Aktoren zwischen den einzelnen CRUD Operationen und damit der eigentliche "innere Schnitt" des Use Cases.

In unserem Beispiel für den Use Case Leistung buchen werden z.B. das Lesen der Leistungen und der Anbieter keine getrennten Schritte sein. Wird der Use Case wie dargestellt als eigener Use Case modelliert (das ist durchaus berechtigt, da eine sofortige Buchung nicht immer Bestandteil einer Reservierung ist), so werden wahrscheinlich in einem ersten Schritt Personen- oder Reservierungsdaten vom Benutzer eingegeben, anschließend werden im System diese Informationen gesucht/gelesen und dann dem Benutzer angezeigt. Für die UCP Methode würden bis dahin 3 Schritte anfallen.

Der 4. und letzte Schritt im (Haupt-) Erfolgsszenario ist dann die Durchführung der eigentlichen Buchung. Ob und inwieweit das Lesen der Leistungen und Anbieter in diesem Erfolgsszenario überhaupt zu berücksichtigen sind oder ob sie Bestandteil eines Alternativszenarios oder eines anderen Schrittes des Use Cases sind (Suchen und Lesen der Reservierung oder eigentliches buchen) kann aus diesem Diagramm nicht entnommen werden.



**Abb. 5:** Beispiel für ein CRUD-Diagramm

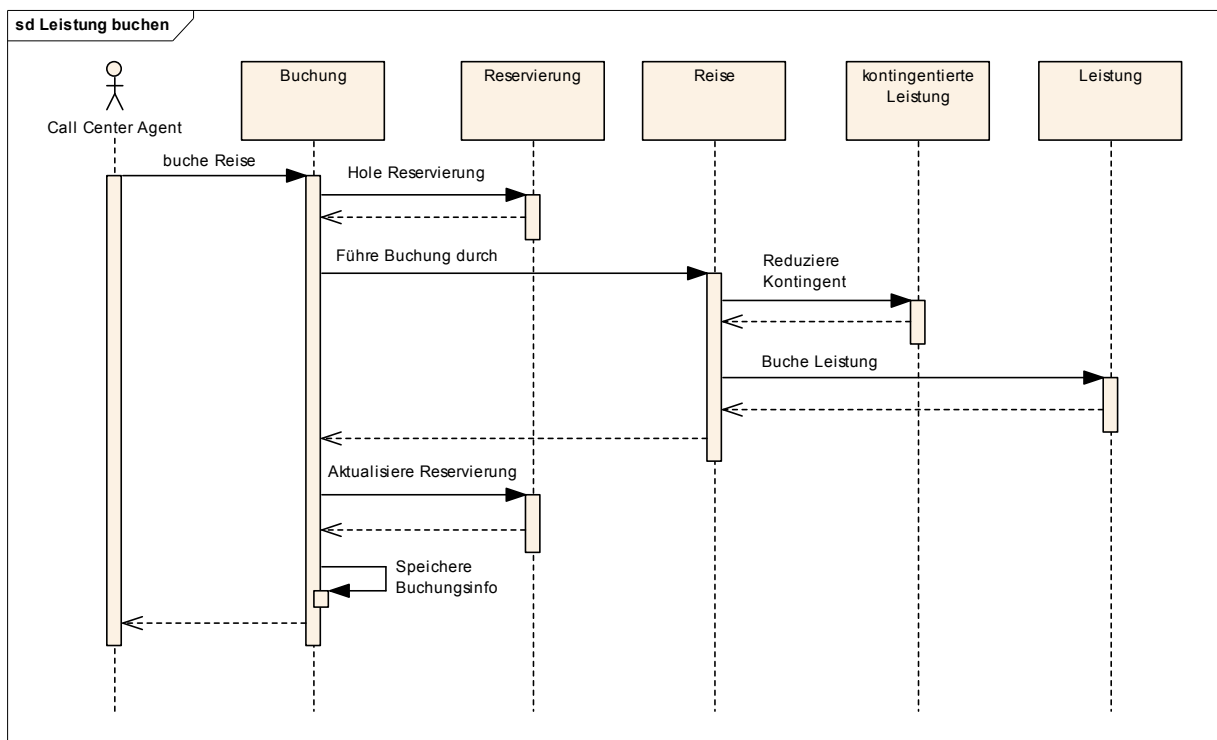
CRUD Diagramme alleine werden uns höchstens den Schnitt der Anwendung in Use Cases liefern, nicht aber ausreichende Hinweise auf die einzelnen Schritte, hier muss grundsätzlich die Beschreibung oder eine der anderen Formen der Darstellung von Use Cases zusätzlich berücksichtigt werden. Generell besteht bei CRUD Diagrammen die Gefahr einer zu feinen Granularität und damit einer zu starken Dekomposition der Use Cases in Funktionen.



## 5.6 Interpretation von Sequenzdiagrammen

Sequenzdiagramme zur Veranschaulichung des Austausches von Nachrichten zwischen Aktoren und Objekten können ebenfalls als Basis für eine UCP Schätzung dienen. In der Praxis wird dies nicht allzu häufig vorkommen, da der Grad der erforderlichen Detaillierung stärker ist als in der Regel in einer solch frühen Phase möglich ist. An dieser Stelle sollen trotzdem Hinweise gegeben werden, wie mit einer solchen Spezifikation umgegangen werden kann.

Falls in der ersten Spalte die Aktoren des Systems die Auslöser der Nachrichten sind, ist eine Strukturierung nach Use Cases direkt möglich. Falls hier schon Objekte genannt werden, ist die Granularität bereits zu fein (auf Ebene von Schritten oder noch detaillierter). Dann müssen die übergeordneten Use Cases vor der Zählung erst wieder zusammengesetzt werden. Gezählt werden dann als Schritte nur die Nachrichten zwischen den Aktoren (1. Spalte) und den Objekten der 1. Hierarchie (2. Spalte). Ausnahmen können Prüfungen (tiefere Ebenen) sein, die Alternativszenarien auslösen, aber auch diese sind in der Regel dann wieder als Interaktion mit dem Actor zu erkennen.



**Abb. 6:** Beispiel für ein Sequenzdiagramm

Im obigen Ausschnitt eines Sequenzdiagramms (Abbildung 6) ist z.B. nur ein einzelner, im Sinne der UCP Schätzung gezählter, Schritt eines Use Cases dargestellt, nämlich „buche Reise“! Die weiteren Schritte sind dann im Ablauf des Diagramms weiter unten dargestellt oder werden sogar in gesonderten Diagrammen beschrieben (häufiger Fall). Im letzteren Fall ist dann also ein einzelner Use Case auf mehrere Diagramme verteilt beschrieben.

Ähnlich wie bei CRUD Diagrammen besteht hier grundsätzlich die Gefahr einer zu starken Dekomposition des Use Cases für die UCP Schätzung.

## 5.7 Fazit

Eine vorliegende Use Case Beschreibung ist erstmal grundsätzlich in jedem Fall geeignet, eine UCP Schätzung für den fachlichen Umfang einer Anwendung durchzuführen und belastbare Ergebnisse zu erhalten. Ob dann auch genug Informationen für die Beurteilung der Kostenfaktoren vorliegen wird hier nicht betrachtet.

Art der Use Case Spezifikation	Eignung für UCP Schätzung	Vorteile	Gefahren
Grobe (textuelle) Beschreibung	gut	Aufzählung kann direkt übernommen werden Szenarien und Schritte sind grob beschrieben	Unterschätzung durch zu wenig Informationen über die Szenarien/Schritte Unterschätzung durch vergessene ganze Use Cases
Textuelle bzw. tabellarische Beschreibung	sehr gut	Aufzählung kann direkt übernommen werden Szenarien und Schritte sind bereits beschrieben	
Aktivitäten-diagramm	sehr gut	Direkte Übernahme von Use Cases und Zählung der Szenarien und Schritte einfach und schnell	Gefahr der Vernachlässigung der Überprüfung gegen textuelle Beschreibung Aktivitätendiagramme können zu fein sein.
Zustands-diagramm	mittel		anderer "Schnitt" durch die Anwendung; Übersehen von Use Cases nur im Zusammenhang mit textuellen Ergänzungen verwendbar!
CRUD Diagramm	mittel	die meisten Use Cases sind bereits gefunden	Schritt müssen erst gefunden werden Übersehen von Use Cases nur im Zusammenhang mit textuellen Ergänzungen empfohlen
Sequenzdiagramm	gut, wenn alle Use Cases erfasst sind, sonst mittel	Schnitt der Use Cases bereits vorgegeben Schritte und Szenarien einfach ersichtlich	Gefahr zu feiner Granularität in der Regel keine vollständige Abdeckung der Anwendung Überprüfung der textuellen Ergänzung empfohlen

**Tab. 8:** Beschreibungsarten und ihre Relevanz für die UCP-Schätzung

Je nach Detaillierung und Art der Use Case Beschreibung ist unterschiedlich viel Aufwand in die Schätzung selbst zu investieren, um eine gleiche Qualität der Schätzergebnisse zu erreichen. Die folgende Tabelle fasst die in den letzten Abschnitten dazu gemachten Äußerungen noch einmal zusammen und basiert auf der Annahme, dass mit den betrachteten Verfahren eine die Anwendung vollständig abdeckende UCP Schätzung erreicht werden soll.

## **6 Mapping von weiteren Spezifikationsformen**

Liegt eine (Grob-)Spezifikation nicht als Use Case Beschreibung in der bisher dargestellten Form vor, muss im Vorfeld der Schätzung abgewogen werden, ob und mit welchem Aufwand das Finden geeigneter Use Cases möglich ist. Insbesondere wenn das Erstellen der Use Cases ausschließlich für die Schätzung (also keine Weiterverwendung im Projekt absehbar) erfolgt, kann dieser initiale Teil der Schätzung einen potentiell hohen Aufwand bedeuten. Dieser Aufwand sollte aber auch unter dem Aspekt der Qualitätssicherung sowohl für das Verständnis der Fachlichkeit als auch das Ergebnis und die Belastbarkeit der Schätzung betrachtet werden. Ein mit der Fachlichkeit vertrauter oder/und erfahrener Use Case Modellierer kann diesen Schritt durchaus in vertretbarer Zeit durchführen (wenige Stunden bis einige Tage je nach Umfang der Anwendung).

Können Use Cases aufgrund mangelnder zur Verfügung stehender Informationen nicht oder nur sehr schwer gefunden werden, wird auch eine Schätzung auf Basis dieser Use Cases nicht belastbar sein. In diesem Fall also besser: Finger davon lassen oder zumindest die Ergebnisse mit wohlwollender Toleranz betrachten! Das muss nicht zwangsläufig bedeuten, dass eine Expertenschätzung nicht möglich ist, aber auch diese dürfte dann nicht wesentlich belastbarer sein. Ein gutes Kriterium ist auch die Überprüfung, ob sich Testfälle relativ einfach bestimmen lassen.

Speziell im Fall von nicht als Use Case Beschreibung vorliegender Spezifikation wird empfohlen, eine Schätzung nach UCP ebenfalls als eine unabhängige Schätzung mindestens zweier verschiedener Personen durchzuführen. Dies gilt insbesondere für das Finden der Use Cases selbst (den "Schnitt"), die eigentliche Gewichtung ist dann nicht mehr ganz so schwierig.

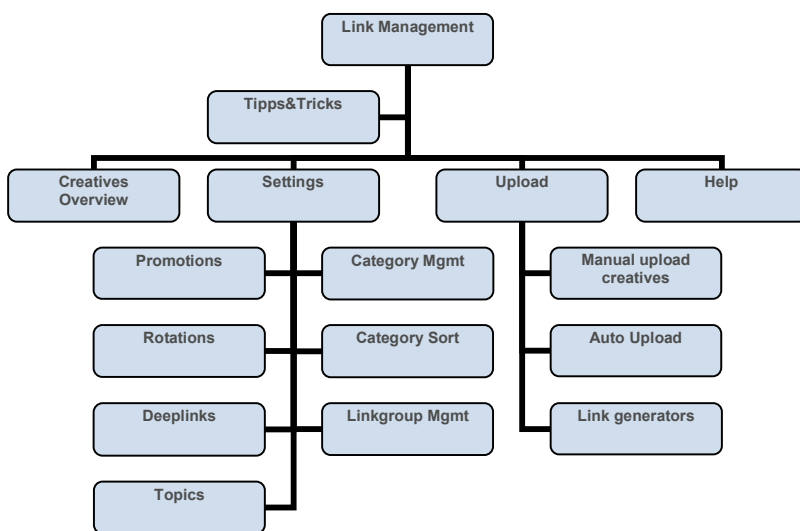
### **6.1 Dialogorientierte Beschreibungen**

Häufig wird in den frühen Phasen von besonders dialoglastigen Projekten nicht mit Use Cases oder textuellen Beschreibungen von Funktionalität gearbeitet. Der Fachbereich sind auch nur Menschen und diese sind stark visuell gesteuert. Daher treffen wir dann zum Zeitpunkt der Schätzung auf Screenshots, Mock-ups oder ähnliche Beschreibungen von Dialogen.

Aus diesen Darstellungen lassen sich in der Regel Use Cases gut finden. Wir möchten dazu an dieser Stelle einige Hinweise geben:

- über den Einstieg des Dialogsystems und das Menusystem lassen sich meist direkt (Haupt) Use Cases oder zumindest fachliche Blöcke ableiten.
- den Workflow im Dialog dann soweit verfolgen, bis "finale" Aktionen ausgeführt werden (z.B. speichern, prüfen, senden, ...).
- ein Dialog kann von mehreren Use Cases verwendet werden, entscheidend ist der Workflow durch das Dialogsystem bis zum Erreichen eines bestimmten fachlichen Ziels.
- Crosslinks aus der Aufrufhierarchie heraus in eine andere (parallele) Ebene der Hierarchie heraus sind Hinweise auf die Verwendung eines anderen Use Cases.
- Buttons bzw. durch andere Interaktionselemente mit der GUI ausgelöste Aktionen sind ein guter Indikator für einen neuen Schritt.
- weitere (auch "verdeckte") Schritte gehen dann entweder aus dem Gesamtkontext oder aus zusätzlichen textuellen oder graphischen Darstellungen hervor oder müssen erfragt werden. *Ein Screenshot alleine reicht für eine UCP Schätzung nicht aus.*

Bei der Zählung können die Dialoge (Reiter, Frames beachten!) direkt übernommen werden. Szenarien und Schritte ergeben sich wie beschrieben aus dem Workflow und der Folge von verwendeten Dialogen für ein bestimmtes fachliches Ziel.



**Abb. 7:** Menüstruktur eines Dialogteilsystems

Dies ist die Menüstruktur für ein Dialogteilsystem aus dem auf jeden Fall eine grobe fachliche Strukturierung ersichtlich ist. Hier liefert die Hauptmenüstruktur nur zum Teil Hinweise auf Use Cases (fehlende Verben in den Menüpunkten!). Ausnahmen sind die Dialoge "Category Sort", "Manual Upload" und "Auto Upload", die sich im Übrigen später auch als echte Use Cases heraus gestellt haben.

Als Beispiel sei hier der Punkt "Category Mgmt" angeführt, zu dem es folgendes Mock-Up gibt.

**Create a new Category** Status  active

Title

Super category

**Manage existing categories**

Tree	ID	Title	Active/Overall Creatives	Status active	Power link	restricted	Action
<input checked="" type="checkbox"/>	1234	<input type="text" value="Marry Christmas Campaign"/>	2/17	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="button" value="Details"/>
<input type="checkbox"/>	3456	<input type="text" value="Anniversaries"/>	0/3	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="button" value="Details"/>
<input checked="" type="checkbox"/>	2344	<input type="text" value="10th Anniversary"/>	4/4	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="button" value="Details"/>
<input type="checkbox"/>	1236	<input type="text" value="10th Anniversary"/>	0/0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="button" value="Details"/>

select all

Action Activate  
Action Deactivate  
Action Delete

**Tab. 9:** So genanntes Mock-Up eines Category Managements

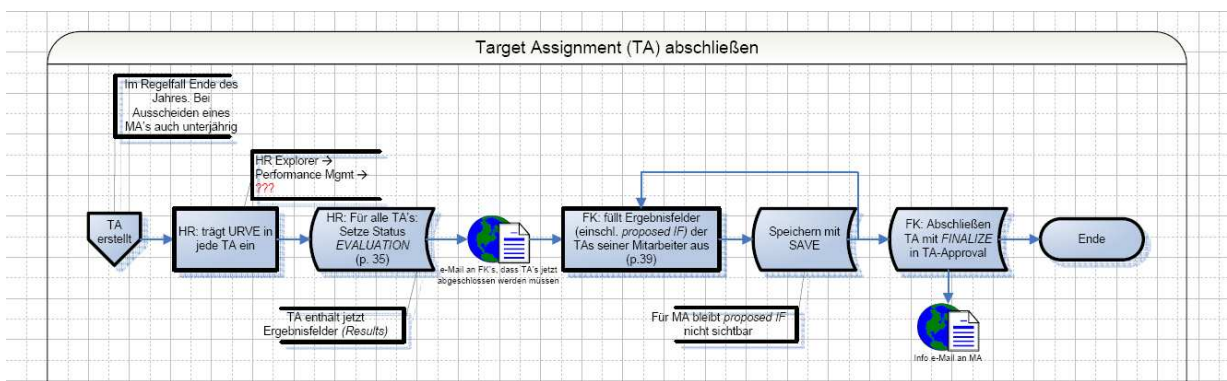
Aus den Buttons können hier direkt die Use Cases "Create Category", "Activate Category", "Deactivate Category", "Delete Category", "Update Category" und "Download as csv" abgeleitet werden. Ob sich hinter den Buttons weitere Schritte und Dialoge verbergen oder der Use Case damit abgeschlossen ist, muss dann aus weiteren Mock-Ups und den textuellen Erläuterungen hervorgehen.

Details sind z.B. hier ein spezielles Szenario von „Update“, aber das geht auch noch nicht alleine und direkt aus dem Mock-up hervor. Die Aktion „Delete“ enthält auch mehrere fachliche Schritte, die erst aus der ergänzenden Beschreibung hervorgehen. Man kann auch vermuten, dass allen diesen Use Cases ein gemeinsamer Schritt „Category suchen“ bzw. „lesen“ vorausgeht, da kein spezieller Button dafür auf diesem Mock-Up zu ersehen ist.

Zusätzlich können Interaktionsdiagramme vorliegen, aus denen ebenfalls Use Cases und Hinweise auf Schritte entnommen werden können (siehe Abbildung 8). Hierbei sind die Dialoge (Screens) als Ellipsen dargestellt und die Aktionen (Buttons) als Kästchen. Es lassen sich wieder die Use Cases anhand der Buttons erkennen: „Create Rotation“, „Update Rotation“, „Deactivate Rotations“, „Activate Rotations“ und „Assign Rotation Creatives“, wobei „Assign Rotation Creatives“ noch mit einem weiteren Dialog fortgesetzt wird.



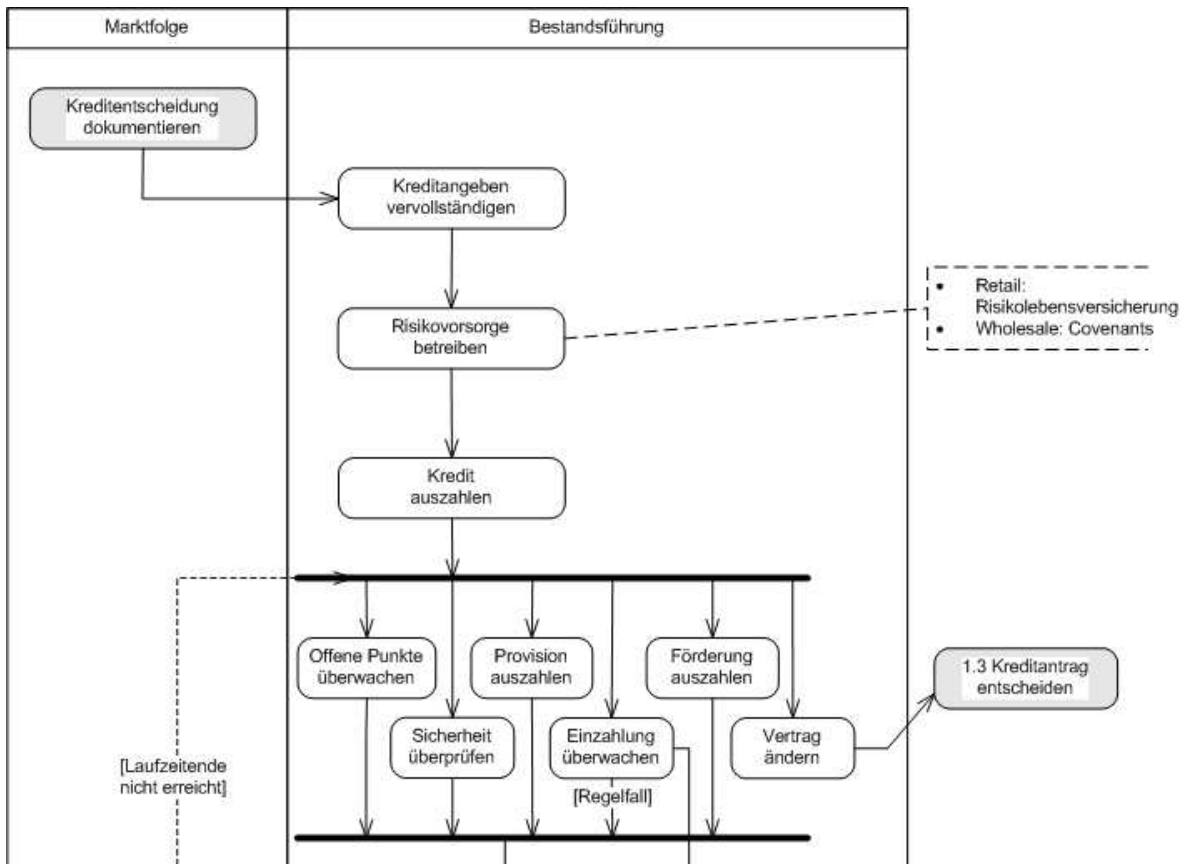
- Use Cases in Prozessbeschreibungen können durch einen Wechsel des Aktors oder Prozessverantwortlichen, durch Erzeugung von permanenten Zwischenergebnissen oder auch durch einen Medien- oder Systemwechsel voneinander getrennt und damit zu finden sein
- falls zusätzliche Informationen zu den Schritten vorliegen, daraus die Use Cases bewerten, im Notfall intuitiv ohne Szenarien und Schritte, evtl. vergleichbare Use Cases aus ähnlichem fachlichen oder technischen Umfeld heranziehen
- falls keine weiteren Informationen vorliegen, hat man außer der Auflistung der Use Cases nicht viel. Daraus kann dann maximal eine grobe Abschätzung gemacht werden, die dann aber nur einen Richtwert darstellt.



**Abb. 9:** Beispiel einer Prozessbeschreibung

In Prozess in Abbildung 9 werden die einzelnen Use Cases durch den (wahrscheinlich manuell durchgeführten) Versand von E-Mails voneinander getrennt. Außerdem wechselt der Akteur von „HR“ zu „FK“. In diesem Fall haben wir 2 Use Cases, die Vorbereitung der „TA“ durch „HR“ und den Abschluss der „TA“ durch „FK“. Es gibt auch Hinweise auf mindestens durchzuführende Schritte, wie das Ausfüllen bestimmter Felder und eine Evaluierungsfunktion.

Ob das Versenden der E-Mail systemseitig unterstützt wird und damit noch zum Use Case gehört, geht aus der Darstellung nicht hervor. Es liegt die Vermutung nahe, dass beide Use Cases in die Kategorie einfach fallen.



**Abb. 10:** Ausschnitt einer Geschäftsprozessbeschreibung

Dieser Ausschnitt zeigt die Geschäftsprozesse schon auf einem größeren Level, die einzelnen Schritte des Prozesses sind eigene Use Cases und sind auch nicht mehr weiter erläutert. Nur auf Basis dieser Informationen ist eine Schätzung sehr schwer möglich, hier helfen nur fachlich ähnliche und bereits bekannte Fälle. Im konkreten Beispiel liegen noch die Informationen gemäß Tabelle 10 vor.

Die textuellen Ergänzungen zeigen, dass es sich hierbei durchaus nicht um triviale Use Cases handelt. Der Use Case Kreditangaben vervollständigen mag noch als einfacher Pflege Use Case betrachtet werden können, aber auch können komplexe Folgeschritte wie Prüfungen unterschätzt werden. Beim Use Case Kredit auszahlen sieht es aber mindestens nach 2 Unter Use Cases Kreditkonto anlegen und Kredit bereitstellen aus. Liegen keine weiteren Detailinformationen vor, erscheint eine Use Case Schätzung hier wenig belastbar.



Nr.	Name	Beschreibung
1	Kreditangaben vervollständigen	<p>Kreditunterlagen kontrollieren auf Vollständigkeit (alle Dokumente, Legitimation, Unterschriften). Fehlende bzw. regelmäßig zu stellende Dokumente auf Wiedervorlage legen.</p> <p>Systemunterstützung:</p> <ul style="list-style-type: none"> <li>• maschinelle Kontrollen durch Dokumentenabgleich / Plausibilitäten etc.</li> <li>• elektronische Prüfvermerke/Unterschrift</li> </ul>
2	Risikovorsorge betreiben	<p>Anlegen einer (Risiko-)Lebensversicherung, Kreditrahmenversicherung bzw. Einholen von Nebenabreden (covenants) etc.</p>
3	Kredit auszahlen	<p>Kreditkonto anlegen, Kredit bereitstellen.</p> <p>Bis zur Ausnutzung der Kreditlinie durch den Kunden: Bereitstellungszinsen einziehen, Kreditkonditionen prüfen (auf Kündigungsgründe bei Nichtabnahme).</p> <p>Wird der Kredit genutzt: Limitausnutzung nachführen.</p> <p>Schufa-Eintrag vornehmen bzw. ergänzen.</p>

**Tab. 10:** Beispiel einer textuellen Ergänzung zur Geschäftsbeschreibung

### 6.3 Funktional orientierte Beschreibungen

Häufig finden sich Spezifikationen auch als Ansammlung oder Auflistung von Funktionen, die ohne Gruppierung in Use Cases vorliegen. Die Vielfalt der Beschreibungsmöglichkeiten ist dabei sehr groß, damit kann eine allgemeine Richtlinie für den Umgang mit diesen Beschreibungen nur sehr grobe Hinweise enthalten.

- die später für die Schätzung zu zählenden Schritte sind in der Beschreibung je nach Detaillierungsgrad bereits in der richtigen Granularität vorhanden. Bei zu feingranularer Beschreibung (siehe auch CRUD-Diagramme, Abschnitt 5.5) müssen Funktionen erst wieder zu Schritten zusammengefasst werden.
- in einem zweiten Schritt sind dann diese Schritte den Use Cases zuzuordnen (das kann man sich ruhig daneben schreiben oder in einer Matrix erfassen), bei Schwierigkeiten orientiert man sich am Hauptschritt der Verarbeitung oder an Entitäten. Auf jeden Fall gelten die Kriterien aus dem Abschnitt 2.3.
- ist nicht bekannt, ob es später weitere Use Cases geben wird, die die Funktionalität nutzen, gehen wir erstmal nur von einem aus.

- es muss für jede angeführte Funktion auch einen Use Case geben, der diese verwendet!
- Querschnittsfunktionalität kann als extra Use Case erfasst werden, um nicht ständig das Problem der Wiederverwendung zu haben (z.B. Hilfe, login, Hauptmenu-Auswahl)

**Funktionsmatrizen:** Ein spezieller Fall von funktional orientierten Beschreibungen sind tabellarische Aufstellungen, in denen bestimmten fachlichen Abläufen oder Objekten über Matrizen Funktionen zugeordnet sind.

Man läuft leicht Gefahr eine solche, eigentlich orthogonal zu Use Cases angelegte Beschreibung zu übernehmen und damit durch den hohen Grad der funktionalen Dekomposition einen zu feinen "Schnitt" der Use Cases zu erhalten und damit Function Points zu zählen. Dieser Hinweis gilt generell für funktionale Beschreibungen.

Funktion	Objekt					
	Textbaustein	Referenz	Feature	Produkttext	Cluster-Objekt	VIB-Template
Suchen	x	x	x	x	x	x
Anzeigen (Bearbeiten)	x	x	x	x	x	x
Ausführen						x
Anlegen	x	x	x	x	x	x
Prüfen eines Features gegen eine VIB			x			
Struktur anzeigen	x		x	x	x	x
Löschen	x	x	x	x	x	x
Übersetzung anlegen	x	x	x	(x)	(x)	x
Übersetzung editieren	x	x	x	(x)	(x)	x
Status wechseln			x	x	x	x
Exportieren	x	x	x	x	x	x
Importieren	x		x			x

**Tab. 11:** Beispiel einer Funktionsmatrix

In Beispiel in Tabelle 11 sind für verschiedene Entitäten die darauf ausführbaren Funktionen aufgeführt. Je nach Ähnlichkeit der Objekte muss entschieden werden, ob jedes Kreuz ein eigener Use Case wird oder ob z.B. ähnliche Objekte gleich bearbeitet werden und dadurch nur unterschiedliche Szenarien eines Use Cases sind, d.h. Spalten zusammengefasst werden können.

Dies geht hier z.B. mit den Funktionen „exportieren“ und „importieren“. Des Weiteren können evtl. spezielle Funktionen wie z.B. „Status wechseln“ als Schritt eines Szenarios des Use Cases „Anzeigen (Bearbeiten)“ betrachtet werden oder einzelne Funktionen sind keine eigenen Use Cases sondern immer Bestandteil eines anderen Use Cases, hier z.B. „Suchen“.

## 6.4 Fazit

Auch aus „Nicht-Use-Case“ Beschreibungen heraus kann eine UCP Schätzung durchgeführt werden. Dem geht allerdings immer ein mehr oder wenig aufwendiger Schritt des Mappings voraus. In diesem Schritt müssen die Use Cases gefunden werden und eine Aufteilung in Szenarien und Schritte vorgenommen werden. Mit zunehmender "Entfernung" der vorhandenen Spezifikation von einer Use Case Beschreibung handelt es sich dabei um die hohe Schule der UCP Schätzung.

Es können hier sicherlich nicht alle Arten von Spezifikationen betrachtet und erfasst werden, für häufig vorkommende sind jedoch Hinweise für den Umgang im Rahmen einer UCP Schätzung gegeben worden.

Art der Spezifikation	Eignung für UCP Schätzung	Vorteile	Gefahren
Geschäftsprozesse als Grob-spezifikationen	Mittel, je nach Grad der Detaillierung	Aufzählung der Use Cases kann direkt übernommen oder einfach gefunden werden, guter Überblick der Anwendung	Unterschätzung durch zu wenig Informationen über die Szenarien/Schritte
Dialog-beschreibungen	gut	Use Cases und Szenarien sind relativ einfach zu finden, gute Abdeckung der Anwendung	verdeckte Schritte, Anwendungslogik ist nicht erfasst, daher immer textuelle Ergänzungen beachten
Funktionale Beschreibungen	je nach Umfang gut	Beschreibung der Schritte und Funktionen, gute Abdeckung der geforderten Funktionalität der Anwendung	Komposition zu Szenarien und Use Cases u.U. aufwändig, Gefahr in Funktionen und damit zu detailliert zu schätzen

**Tab. 12:** Eignung der weiteren Spezifikationsformen für die UCP-Methode

## 7 Zusammenfassende Hinweise für die Anwendbarkeit und Evaluierung

Zusätzlich zu den Bemerkungen über das Mapping gibt der letzte Abschnitt noch einmal in Form eines tabellarischen Überblicks Hinweise und Kriterien für die Anwendbarkeit der UCP Methode und Möglichkeiten der Evaluierung einer durchgeführten Schätzung. Zum großen Teil sind diese bereits in vorherigen Abschnitten genannt, begründet und erläutert worden.

Erscheinung	Ursache	Auswirkung	Korrekturmöglichkeit
Schwierigkeit beim Finden der Use Cases	Beschreibung zu grob, nicht genug Informationen vorhanden	Schätzung nicht durchführbar, nicht plausibel	Vorsicht, hier hilft auch UCP nicht!
Schwierigkeit beim Finden der Use Cases	keine Use Case Beschreibung und Mapping von bestehender Beschreibung schwierig	Schätzung nicht oder schwierig durchführbar, nicht plausibel	Neu nachdenken ☹, zweiten Schätzer fragen
sehr viele kleine Use Cases (häufig nur 1 oder 2 Schritte)	Genereller Schnitt wahrscheinlich zu fein granular	potenziell Überschätzung	Die zugrundeliegende Use Beschreibung oder das für eine andere Art der Beschreibung verwendete Mapping "passt" nicht und sollte überdacht werden.
Verhältnismäßig (zu den kleinen und mittleren) sehr viele große Use Cases	Genereller Schnitt wahrscheinlich zu grob granular	potenziell Unterschätzung	
Sehr große Use Cases (>12 Szenarien, Schritte, Dialoge)	Schnitt wahrscheinlich zu grob granular	potenziell Unterschätzung	Dekomposition der betroffenen Use Cases, evtl. Unter Use Cases finden
Hoher Anteil von Re-Use	Wartungs- oder Integrationsprojekt, Einsatz von Frameworks, Produkten	hohe Schätzunsicherheit	
Hoher Anteil von Re-Use	starke Kopplung der Use Cases, starke Berücksichtigung von Anwendungsfunktionen über Re-Use	Verfälschung der Schätzung	Wiederverwendete Anwendungsfunktionen/Schritte nur beim ersten Use Case zählen

**Tab. 13:** Mögliche Problembehandlungsformen bei der UCP-Anwendung

Damit stellt dieser Preprint insgesamt einen Leitfaden für die zweckmäßige Auffindung von Anwendungsfällen dar, der empirisch qualifiziertere Aussagen im Rahmen der UCP-Methode ermöglicht.

## 8. Literatur

1. Abran, A.; Ndiaye, I.; Bourque, P.: *Evaluation of Black-box Estimation Tool: A Case Study*. Software Process Improvement and Practice, 12(2007), pp. 199-218
2. Albrecht, A.J.: *Measuring Application Development Productivity*. In Proc. IBM Applications Development Symposium. GUIDE Int. and Share Inc., IBM Corp., Monterey, CA, 1979
3. Anda, B.; Dreiem, H.; Sjoberg, D.I.K.; Jorgensen, M.: *Estimating Software Development Effort based on Use Cases - Experiences from Industrie*. In Gogolla, M., Kobryn, C., Hrsg., 4th International Conference on the Unified Modeling Language (UML2001). Springer-Verlag, Toronto, Canada, 2001.
4. Basili, V.R.; Zelkowitz, M. V.: *Empirical Studies to Build a Science of Computer Science*. Comm. of the ACM, 50(2007)11, pp. 33-43
5. Boehm, B. W.: *Software Engineering*. IEEE Computer Society, Los Altimos, 2007
6. Boehm, B. W.: *Software Cost Estimation with COCOMO II*. Prentice Hall, 2000.
7. Booch, G.; Rumbaugh, J.; Jacobson, I.: *The Unified Modeling Language User Guide*, Addison-Wesley Object Technology Series, 1999
8. Bundschuh M.; Fabry A.: *Aufwandschätzung von IT-Projekten*, 2.Aufl., mitp-Verlag, Bonn 2004.
9. Clem, R. (2005): *Project Estimation with Use Case Points*. <http://www.codeproject.com/gen/design/usecasep.asp>, 15 März 2007
10. Cockburn ,A.(2003): *Writing effective use cases*. The Agile Software Development Series. 8. Auflage. Addison-Wesley, 2003.
11. Cockburn, A.(1995): *Structuring Use Cases with Goals*. <http://alistair.cockburn.us/crystal/articles/sucwg/structuringucswithgoals.htm>, 15. Januar 2007
12. DASMA (2005): *Metrik-Glossar*, <http://www.dasma.org/contray/html/informationen/metrik-glossar/e>, 15. März 2007
13. Dumke, R.: *Software Engineering*. 4. Auflage; Vieweg Verlag, 2003
14. Ebert, C.; Dumke, R.: *Software Measurement – Establish, Extract, Evaluate, Execute*. Springer Verlag, 2007
15. Fraunhofer IESE: *Virtuelles Software Engineering Kompetenzzentrum*. <http://www.software-kompetenz.de/?4784>, 15 März 2007
16. Frohnhoff, S.; Jung, V.; Engels, G. (2006): *Use Case Points in der industriellen Praxis*. In “Applied Software Measurement - Proceedings of the International Workshop on Software Metrics and DASMA Software Metrik Kongress”, Abran, A. et al. Eds. Shaker Verlag, 2006, pp. 511-526.
17. Garmus, D.; Herron, D.: *Measuring the Software Process – A Practical Guide to Functional Measurement*. Prentice Hall PTR, 1996
18. Hericko, M.; Rozman, I.; Zivkovic, A. *A formal representation of functional size measurement methods*. The urnal of Systems and Software, 79(2006), pp. 1341-1358
19. IFPUG: *Function Point Counting Practices Manual*, Release 4.0. International Function Point Users Group. Westerville, Ohio (USA), 1994

20. Jacobson, I.; Christerson, M.; Jonsson, P. und Overgaard, G.: *Object-Oriented Software Engineering. A Use Case Driven Approach*. 4. Auflage. Addison-Wesley, 1993
21. Junior, O.; Farias, P.; Belchior, A.: *Fuzzy Modeling for Function Point Analysis*. Software Quality Journal, 1(2003), pp. 149-166
22. Karner, K: „Metrics for Objectory“. Diploma thesis, University of Linköping, Sweden, No. LiTHIDA-Ex-9344:21, December 1993
23. Koirala, S. (2004): *How to Prepare Quotation Using Use Case Points*, <http://www.codeproject.com/gen/design/usecasepoints.asp>, 15 Januar 2007
24. Lothar, M.; Dumke, R.: *Points Metrics – Comparison and Analysis*. Proc. of the 11<sup>th</sup> IWSM, Shaker Verlag, 2001, pp. 228-267
25. McConell, S.: *Software Estimation*. Microsoft Publ., 2006
26. Schneider, G.; Winters, J.: *Applying Use Cases - A Practical Guide*. Addison Wesley Longman, Inc., 1998
27. Siedersleben, J.: *Moderne Software-Architektur*. dpunkt-Verlag, 2004
28. Sneed, H.: *Software-Projektkalkulation*. Hanser Verlag, 2005