

Duale Hochschule Baden-Württemberg

Mannheim

## **Erste Projektarbeit**

### **Erarbeitung eines Lösungsentwurfes für eine IT-Lösung zur Kapazitätsplanung der Kabinencrew**

Studiengang Wirtschaftsinformatik - Sales & Consulting

Bearbeitungszeitraum: 15.05.2017 - 29.08.2017

Verfasser:	Julian Garske
Matrikelnummer:	6728241
Kurs:	WWI SCA16
Studiengangsleiter:	Prof. Dr. Frank Koslowski
Wissenschaftlicher Betreuer:	Günter Stumpf
Telefon:	01511 8237778
Mailadresse:	guenter.stumpf@esosec.de
Ausbildungsbetrieb:	Lufthansa Systems GmbH & Co. KG Am Prime Parc 1 D 65479 Raunheim
Unternehmensbetreuer:	Iwan Berger
Telefon(Firma):	+49 (0)69 696 74135
Mailadresse(Firma):	iwan.berger@lhsystems.com

# Inhaltsverzeichnis

<b>Abkürzungsverzeichnis</b>	<b>I</b>
<b>Abbildungsverzeichnis</b>	<b>II</b>
<b>1 Einleitung</b>	<b>1</b>
1.1 Motivation . . . . .	1
1.2 Problemstellung und -abgrenzung . . . . .	2
1.3 Ziel der Arbeit . . . . .	2
1.4 Vorgehen . . . . .	3
1.5 Überblick über COMPAS Cockpit . . . . .	4
1.6 Bisherige Vorgehensweise der Kapazitätsplanung für die Kabinenbesatzung	4
<b>2 Problemanalyse</b>	<b>6</b>
2.1 Erhebung der Pilotendaten von der Crew Database . . . . .	6
2.2 Integration in das COMPAS-Umfeld . . . . .	7
2.3 Erfassen der Anforderungen . . . . .	7
2.4 Mehrfachqualifikationen der Kabinenbesatzung . . . . .	7
<b>3 Grundlagen/Methodischer Ansatz</b>	<b>9</b>
3.1 Der Software-Lebenszyklus . . . . .	9
3.2 Vorgehen bei der Beschreibung und Analyse der Anforderungen . . . . .	10
3.3 Qualitätssicherung der Anforderungen . . . . .	11
3.3.1 Qualitätsaspekt Inhalt . . . . .	13
3.3.2 Qualitätsaspekt Dokumentation . . . . .	13
3.3.3 Qualitätsaspekt Abgestimmtheit . . . . .	14
3.4 Anforderungskategorisierung . . . . .	14
3.5 Abgrenzung des Systems und Systemkontextes . . . . .	16
3.6 Arten und Ziele des Prototypings . . . . .	17
<b>4 Istanalyse</b>	<b>19</b>
4.1 Bestandsrechnung in COMPAS Cockpit . . . . .	19
4.2 Anforderungsanalyse- und kategorisierung . . . . .	19
4.3 Die Schnittstelle: Das Crew Management System . . . . .	20

<b>5</b>	<b>Konkrete Modernisierung und Anpassung von COMPAS Cockpit</b>	<b>21</b>
5.1	Veränderungen bei der Datenaufbereitung . . . . .	21
5.2	Veränderungen in der Bestandsrechnung . . . . .	21
5.3	Abbilden der Mehrfachqualifikationen und Einbindung der Kleingruppen	22
<b>6</b>	<b>Zusammenfassung und Ausblick</b>	<b>23</b>
6.1	Änderungen an COMPAS Cockpit um es für eine automatisierte Kapazitätsplanung der Kabinenbesatzung zu nutzen . . . . .	23
6.2	Probleme bei der Anpassung der Bedarfsrechnung der Kabinenbesatzung nach dem Vorbild von COMPAS Cockpit . . . . .	23
6.3	COMPAS Cockpit als Vorlage - kritische Auseinandersetzung . . . . .	24
6.4	Weitere Vorgehensweise . . . . .	25
 <b>Literaturverzeichnis</b>		
<b>Glossar</b>		<b>III</b>
<b>Anhang</b>		<b>IV</b>

# Abkürzungsverzeichnis

**BT** Beschäftigungstage

**CAB** Compas Cabin

**CDB** Crew Database

**CMS** Crew Management System

**COC** Compas Cockpit

**KG** Kleingruppe

**PU** Planungseinheit

# **Abbildungsverzeichnis**

# 1 Einleitung

## 1.1 Motivation

Bereits seitdem es kommerzielle Passagierflüge gibt, ist eine konkrete Zuteilung der Besatzung für die Flüge notwendig. Dabei müssen nicht nur Fehlzeiten wie Urlaub oder Krankheit, sondern auch andere Faktoren wie z.B. Teilzeit berücksichtigt werden. Außerdem werden für unterschiedliche Flugzeuge auch unterschiedliche Qualifikationen benötigt. Bei immer größer werdenden Fluggesellschaften, wie der Lufthansa mit insgesamt ca. 20.000 Besatzungsmitgliedern der Kabine, stellt die mengenmäßige Einteilung der Kabinenbesatzung aus diesen Gründen oft eine Herausforderung dar. Die sogenannte Kapazitätsplanung ist deshalb für viele Airlines ein wichtiger Bestandteil des Flugbetriebs. Mit der Kapazitätsplanung hängt die Schulungsplanung eng zusammen. Schulungen dauern oft einige Wochen und müssen daher langfristig vorher geplant werden, sodass Fehlzeiten ausgeglichen werden können und die Qualifikationen nach der Schulung aktualisiert werden. Das Ziel der Kapazitäts- und Schulungsplanung ist, „durch rechtzeitige Neueinstellungen und Umschulungen die richtige Menge an Piloten mit der richtigen Qualifikation zum richtigen Zeitpunkt auf einer Flotte bereitzustellen“ (vgl. Berger, 2016, S.19).

Für die Kapazitäts- und Schulungsplanung der Cockpit-Besatzung im Lufthansa Konzern gibt es seit 2000 das von Lufthansa Systems entwickelte System COMPAS (Crew Operation Manpower Planning Advanced System), im folgenden Compas Cockpit (COC) genannt. Dadurch ist eine Planung der Einteilung und Schulungen für die etwa 5.000 Piloten bis zu 15 Monate in Zukunft möglich. Bis heute wird die Funktionalität dieses Programms regelmäßig erweitert.

Mit der Entwicklung von COC entstand der Wunsch, ein ähnliches Programm auch für die Kabinenbesatzung zu entwickeln. Durch diese systematische Lösung sollen aufwendige manuelle Prozesse abgelöst und damit Zeit gespart und Fehler minimiert werden (vgl. Winkel, 2017, S.4). Ein Auftrag für das Projekt wurde von dem Kunden, der Lufthansa Passage Airline, noch nicht vergeben, weshalb es noch nicht zu einem Projekt mit einer konkreten Analyse oder Entwicklung kam.

Im Jahr 2016 hatte sich die Firma M2P Consulting bereits mit der bisherigen Kapazitäts-

planung der Kabinenbesatzung auseinandergesetzt und geprüft, wie man diese verbessern könne und ob COC dafür in Frage käme. Sie kamen zu dem Ergebnis, dass kurzfristig zwar die bisherige Vorgehensweise der Kapazitätsplanung (siehe 1.6) optimiert werden könne, um die Qualität zu verbessern, langfristig solle es aber durch eine systemseitige Lösung abgelöst werden (vgl. M2P, 2016, S.8-10).

## **1.2 Problemstellung und -abgrenzung**

Zurzeit werden die mengenmäßigen Kapazitäten für die Einsätze der Kabinenbesatzung mithilfe von verschiedenen Excel-Tabellen geplant. Diese ca. 200 Tabellen enthalten Daten aus unterschiedlichen Quellen, die für die Zuteilung der Besatzung erforderlich sind. Aufgrund der großen Datenmenge und den Verflechtungen der Tabellen untereinander kommt es oft zu Problemen und Fehlern bei der Kapazitätsplanung (vgl. Berg u. a., 2017).

Nach dem Vorbild von COC soll eine automatisierte Kapazitäts- und Schulungsplanung jetzt auch für die Kabinenbesatzung unter dem Titel Compas Cabin (CAB) ermöglicht werden. Um den Umfang dieses Projektes einzugrenzen, wird mit der Kapazitätsplanung begonnen, die, sobald sie fertiggestellt wurde, um die Schulungsplanung ergänzt wird. Dabei gilt es, zu ermitteln wie weit COC dafür als Vorlage genutzt werden kann, da dieses Tool den geforderten Zweck bereits für die Cockpitbesatzung erfüllt.

Im Vordergrund steht dabei zuerst die Anpassung der bereits vorhandenen Bestandsrechnung in COC an die Kapazitätsplanung der Kabinenbesatzung. Diese informiert über den aktiven, verfügbaren Bestand an Piloten, indem von einem Ausgangswert, den „brutto Beschäftigungsköpfen“, unterschiedliche An- und Abwesenheiten verrechnet werden. Die Bestandsrechnung ist die spätere Grundlage für die Kapazitätsplanung, in der sie mit dem Bedarf ins Verhältnis gesetzt wird, um ein Delta zu ermitteln, welches eine Über- oder Unterdeckung in Beschäftigungstagen anzeigt.

## **1.3 Ziel der Arbeit**

Zunächst war das Ziel der Arbeit, einen Lösungsentwurf für die Entwicklung eines Prototypes zur Kapazitätsplanung der Kabinenbesatzung zu erarbeiten. Darüber hinaus sollte

diskutiert werden, wie weit COC als Vorlage genutzt werden kann und welche Unterschiede es zur Kapazitätsplanung der Cockpitbesatzung gibt.

Relativ schnell wurde klar, dass der Aufwand, diesen Lösungsentwurf zu erarbeiten, den Umfang dieser Arbeit überschreitet. Das Hauptelement der Kapazitätsplanung ist die Delta-rechnung, in der Bestand und Bedarf zueinander ins Verhältnis gesetzt werden und das Delta ermittelt wird. Besonders eine Lösung für die Automatisierung der Bedarfsrechnung der Kabinenbesatzung zu finden, in der für einen bestimmten Zeitraum der Bedarf an Flugpersonal der einzelnen Flotten ermittelt wird, stellte sich schnell als kompliziert heraus. Deswegen fokussiert sich diese Arbeit nur auf den ersten Schritt der Kapazitätsplanung, die Bestandsrechnung.

Während offiziell noch nicht entschieden wurde, ob das Programm ein Tool von COC wird oder ob ein eigenständiges und moderneres Programm entwickelt wird, lässt sich bereits eine Tendenz erkennen. Die einfachste Lösung für alle Stakeholder ist, COC als Vorbild zu nutzen, da es viele Gemeinsamkeiten mit dem gewünschten Programm bereits beinhaltet. Daher wird dieses ähnliche Tool in dieser Arbeit als Ausgangspunkt genutzt. Diese Projektarbeit fokussiert sich damit auf die Unterschiede zwischen der Bestandsrechnung in COC und der gewünschten automatisierten Bestandsrechnung für die Kabinenbesatzung. Es werden Lösungsansätze für Probleme, die aus erforderlichen Veränderungen entstehen, gefunden und beschrieben, wie COC angepasst werden sollte, um eine automatisierte Bestandsrechnung für die Kabinenbesatzung zu ermöglichen.

## **1.4 Vorgehen**

Zunächst ist es nötig, die Anforderungen zu ermitteln und abzuklären. Dafür muss zuerst die aktuelle Vorgehensweise der Kapazitätsplanung von der Cockpit- und der Kabinenbesatzung verstanden werden. Danach werden Anforderungen an die automatisierte Kapazitätsplanung der Kabinenbesatzung erfasst. Die Anforderungen werden in diesem Projekt hauptsächlich von den momentanen Planerinnen, die das spätere Programm anwenden, gestellt.

Danach werden Gemeinsamkeiten und Unterschiede zwischen CAB und COC ermittelt, um damit bewerten zu können, wie COC als Vorlage geeignet ist. Eine Anwenderdokumentation dient als Grundlage für die Entwicklung des Prototyps. Sie beschreibt aus fachlicher Sicht, wie Nutzer mit dem Programm interagieren und welche Funktionen es beinhaltet.



## **1.5 Überblick über COMPAS Cockpit**

COC (Crew Operation Manpower Planning Advanced System) dient als Ausgangspunkt für das Projekt und nur einige Besonderheiten müssen bei dem aktuellen Programms angepasst werden, um die Kapazitätsplanung der Kabinenbesatzung dadurch zu automatisieren (vgl. Wagner u. a., 2017b). Deshalb sind ein Überblick über COC und Kenntnisse der wichtigsten Funktionen davon wichtig. Dieser Überblick stammt aus einem betriebs-internen Dokument (vgl. Berger, 2016).

COC ist das zentrale Tool für die Kapazitätsplanung des Cockpitpersonals. Ziel dieser Planung ist es, die richtige Anzahl an Piloten mit passenden Qualifikationen zu einem bestimmten Zeitpunkt auf eine Flotte bereitstellen zu können. Das kann kurzfristig gesteuert oder langfristig prognostiziert werden. Diese Planung ist sehr komplex, da sehr viele Einflussfaktoren dort berücksichtigt werden müssen.

Um während der Planung den Überblick zu behalten, teilt COC jeden Piloten eindeutig in eine Planungseinheit (PU) ein, die durch Flugzeugtyp, Flotte, Fluggesellschaft, Standort, Funktion und Unterfunktion definiert ist. Dadurch kann für jede PU für einen bestimmten Zeitraum tagesgenau z.B. der Bestand und Bedarf ermittelt und verglichen werden.

COC besteht nach einigen Erweiterungen mittlerweile aus folgenden drei Hauptmodulen:

- Deltarechnung: Setzt Pilotenbestand und -bedarf ins Verhältnis zueinander
- Schulungsplan: Planung und Einteilung von Schulungskursen
- Reportfunktionen: Detailabfragen bestimmter Daten

Als Zusatzmodule kommen Schnittstellen zu der COMPAS Datenbank und zu dem Bewerbersystem für das Cockpit hinzu (vgl. Berger, 2016, S.19).

## **1.6 Bisherige Vorgehensweise der Kapazitätsplanung für die Kabinenbesatzung**

Bisher gibt es für die Kapazitätsplanung der Kabine noch kein spezialisiertes Tool. Zwei Angestellte der Lufthansa Passage planen die Kapazitäten mithilfe von Excel-Tabellen. Dafür verknüpfen sie ca. 220 Tabellen miteinander (vgl. Berg u. a., 2017). Da es kaum möglich ist, jede einzelne Person konkret einzuteilen, wird jede einer Kleingruppe (KG)

zugewiesen. Diese werden ähnlich wie die PUs durch Funktion, Homebase, Unterfunktion, Fluggesellschaft und ein oder mehreren Flugzeugtypen definiert. Auf Grundlage dieser KGs wird die Kapazitätsplanung durchgeführt und nur bei Bedarf können einzelne Personen genauer betrachtet werden.

Nach M2P Consulting sei „die Handlungsfähigkeit der Kapazitätsplanung in Bezug auf die zukünftige Herausforderungen stark eingeschränkt“ (M2P, 2016, S.5) und „deckt keine der definierten Soll-Funktionalitäten der Kapazitätsplanung ausreichend ab“ (M2P, 2016, S.6). Bei den Soll-Funktionalitäten handelt es sich nach M2P um langfristige Bereederung und Budgetplanung und um mittelfristige Bereederung (vgl. M2P, 2016, S.6).

## **2 Problemanalyse**

### **2.1 Erhebung der Pilotendaten von der Crew Database**

Im Vergleich zu COC, welches mit Daten von etwa 6.000 Piloten umgeht, muss CAB in der Lage sein, Daten von etwa 20.000 Personen ohne große Schwierigkeiten verarbeiten zu können (vgl. Wagner u. a., 2017b). Um diese Daten aktuell zu halten, werden alle zwei Wochen Quelldaten vorbereitet, was bereits in COC einige Stunden dauert, für CAB wäre es also bei einer linearen Steigung fast das Vierfache. Es besteht dabei jedoch das Risiko, dass die Laufzeit sogar exponentiell ansteigt. Bei einer langen Laufzeit erhöht sich zudem auch die Fehleranfälligkeit, da Abstürze wahrscheinlicher werden.

Hinzu kommt, dass die Datenbank nicht lange von CAB ausgelastet werden darf, weil auch andere Schnittstellen darauf zugreifen. Eine zu starke Auslastung kann dabei zu großen Problemen in unterschiedlichen Bereichen und Systemen bei der Lufthansa Passage führen (vgl. Wagner u. a., 2017b).

Bei der Erhebung der Daten in COC werden von der Crew Database (CDB), der zentralen Datenbank, alle benötigten Daten herausgesucht und daraus neue Views erstellt, sodass die Daten tagesgenau für die nächsten 450 Tage vorliegen. Dieses Erstellen der Views wird in dieser Arbeit auch Datenaufbereitung genannt. Daraus entstehen drei Tabellen. Sie enthalten Bestandsdaten sowohl für jede Person (personenbasiert) als auch für jede PU (kumuliert) und Stammdaten jedes Mitarbeiters, die für die Kapazitätsplanung benötigt werden. Die hohe Anzahl an Datensätzen führt dazu, dass die Tabellen der Views bis zu drei Millionen Einträge enthalten, die alle von der CDB aufbereitet werden (vgl. Hentschel u. a., 2017).

Aus diesem Problem resultiert, dass für die Datenaufbereitung der CDB eine neue Architektur benötigt wird, um das fast 20 Jahre alte Design von COC abzulösen (vgl. Hentschel u. a., 2017). Für eine Veränderung des Architekturansatzes zur Datenaufbereitung wird ein grobes Verständnis der bisherigen Architektur von COC benötigt.

## **2.2 Integration in das COMPAS-Umfeld**

Sobald CAB weit genug entwickelt worden ist, soll es, den aktuellen Anforderungen nach, in das bereits bestehende COC integriert werden. Der Kunde möchte momentan kein weiteres Programm haben. Im Gegensatz dazu soll aber auch ein modernes Programm entwickelt werden. Deshalb ist es unklar, ob man dieses im Endeffekt in COC integriert oder doch ein neues Programm entwickelt, was sogar als Vorlage für eine Modernisierung von COC genutzt werden kann. Das Ergebnis dieses Projektes soll auch als Entscheidungshilfe dienen.

## **2.3 Erfassen der Anforderungen**

Bei der Anforderungsermittlung geht es darum, die vollständige und korrekte Anforderungen von den Stakeholdern zu ermitteln. In diesem Projekt werden die Anforderungen größtenteils von den Kabinenplanerinnen gestellt, dem sogenannten Fachbereich.

Herausforderungen bei der Ermittlung sind unterschiedliche und häufig wechselnde Anforderungen. Die Stakeholder können sie oft selber nicht genau benennen und ausdrücken, sodass die Wünsche sich zu widersprechen scheinen oder durch neue Ideen und Vorschläge ändern. Ein anderer problematischer Teil ist das Sprachverständnis während der Anforderungsermittlung. Der „Fachbereich [kennt] in den seltensten Fällen die Fachbegriffe des Entwicklers [...] und umgekehrt [...]. Somit ist eine Art 'Übersetzungsprozess' zwischen der Sprache des Fachbereichs und der des Entwicklers notwendig“ (Alpar u. a., 2016, S.319) . Für den Auftragnehmer bedeutet das eine ständige Hinterfragung aller Details und Fachbegriffe sowie die Absprache jeder Kleinigkeit mit den Stakeholdern, sodass sie vollständig ihren Anforderungen entsprechen.

## **2.4 Mehrfachqualifikationen der Kabinenbesatzung**

Jede Person der Kabinenbesatzung kann mehreren Flotten zugeordnet sein. Eine Person wird zwar genau einer KG zugeteilt, aber diese KG kann bis zu drei, aber auch weniger Flugzeugtypen haben, für die die zugeordneten Flugbegleiter und -begleiterinnen qualifiziert sind (vgl. Berg u. a., 2017).

Auch in COC ist die Zuteilung jeder Person zu einer PU eindeutig. Jede PU wird aber, im Gegensatz zu der Kabinenbesatzung, durch einen eindeutigen Flugzeugtypen definiert. Da die PUs Basis für die Berechnungen sind, basiert COC auf der Tatsache, dass eine Person, anders als die Kabinenbesatzung, in genau einer Rolle auf genau einer Flotte fliegt. Deshalb muss das gewünschte Tool sinnvoll mit den Mehrfachqualifikationen der Kabinenbesatzung umgehen können.

Eine Übersicht aktueller KGs und PUs befindet sich im Anhang C.

### 3 Grundlagen/Methodischer Ansatz

#### 3.1 Der Software-Lebenszyklus

Software Projekte lassen sich in einzelne Phasen unterteilen. Diese „Phasen, die ein Softwareprodukt bei seiner Herstellung und dem späteren Einsatz durchläuft“ (Brich u. Hasenbalg, 2013, S.173) nennt man den Software-Lebenszyklus eines Produktes.

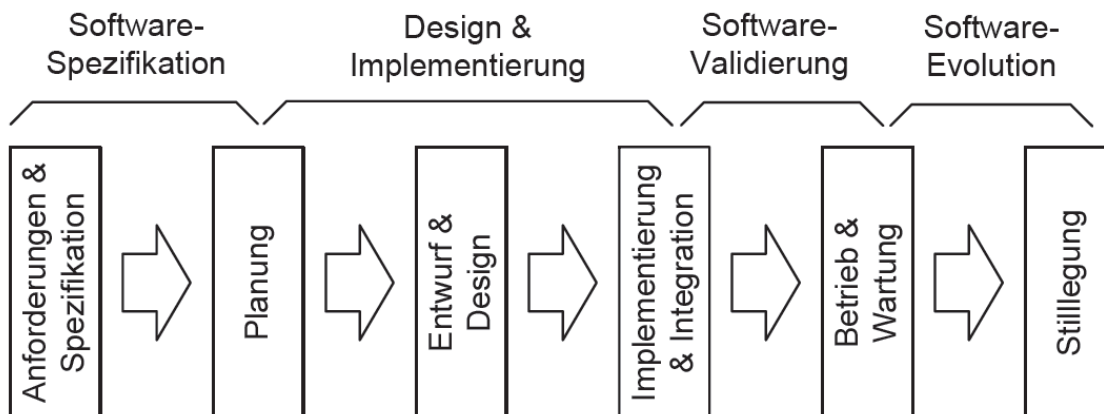


Abbildung 1: Lebenszyklus eines Softwareprojekts

(Schatten u. a., 2010, S.13)

In Abbildung 1 lassen sich die wesentlichen Schritte des Software-Lebenszyklus erkennen. Diese sind hier entweder grob in vier Stufen oder differenzierter in sechs Stufen unterteilt. Es gibt keine Festlegung über die Anzahl oder die Bezeichnungen der Abstufungen, sodass es sehr viele unterschiedliche Versionen des Software-Lebenszyklus und damit Vorgehensmodelle für die Softwareentwicklung gibt. Grundlegende Dinge, die hier durch *Software-Spezifikation*, *Design und Implementierung*, *Software-Validierung* und *Software-Evolution* dargestellt werden, „finden sich [jedoch] nahezu in allen Projekten wieder“ (Schatten u. a., 2010, S.13).

Detaillierter lässt sich der Software-Lebenszyklus in die sechs Phasen *Anforderung und Spezifikation*, *Planung*, *Entwurf und Design*, *Implementierung und Integration*, *Betrieb und Wartung* und *Stilllegung* aufteilen.

Die Einteilung eines Softwareprojekts in Phasen „kann den Entwicklerteams als Leitlinie dienen, um ein Software-Projekt im Team erfolgreich zu strukturieren und abzuwickeln“ (Schatten u. a., 2010, S.11). Dabei können die Phasen je nach Vorgehensmodell variieren oder wahrgenommen werden.

### **3.2 Vorgehen bei der Beschreibung und Analyse der Anforderungen**

Die Beschreibung und Analyse von Anforderungen ist der erste Schritt bei fast jedem IT-Projekt. Zunächst wird „aus der in einer systematischen Ermittlung gewonnenen Information [...] bei der Dokumentation eine präzise Anforderungsspezifikation erstellt“ (Partsch, 2010, S.44). In diesem Fall wird in Zusammenarbeit mit dem Kunden eine Anwenderdokumentation erstellt, die die Anforderungen beschreibt. Diese Anwenderdokumentation dient als Pflichtenheft und enthält „die vom Auftragnehmer erarbeiteten Realisierungsvorgaben“ (Pohl u. Rupp, 2015, S.41).

Statt dem Lastenheft werden für die Erstellung der Anwenderdokumentation nur mündliche Quellen genutzt, da es sich in diesem Fall um einen Entwurf für die Entwicklung eines Prototyps handelt. Mit dessen Hilfe soll es dem Nutzer vereinfacht werden, ein Lastenheft für die Entwicklung des endgültigen Programms zu erstellen. Anders als gewöhnlich wird in diesem Projekt also eine Art Pflichtenheft mit einem Realisierungsvorschlag vom Auftragnehmer geliefert und danach erst das Lastenheft erstellt. Der Auftraggeber formuliert dann darin, wie weit er mit dem Lösungsvorschlag, also dem Prototypen, einverstanden ist und welche Änderungen und Ergänzungen benötigt werden.

Das Ziel der Anforderungsanalyse ist, „möglichst vollständige Kundenanforderungen in guter Qualität zu dokumentieren und dabei Fehler möglichst frühzeitig zu erkennen und zu beheben“ (Pohl u. Rupp, 2015, S.11). Dafür gibt es unterschiedliche Methoden, die jedoch alle „vor allem gesunden Menschenverstand voraus[setzen]“ (Partsch, 2010, S.58). Quelle zur Ermittlung der Anforderungen sind Dokumente, bereits existierende Systeme und hauptsächlich die sog. Stakeholder. Diese sind „Person[en] oder Organisation[en], die (direkt oder indirekt) Einfluss auf die Anforderungen ha[ben]“ (Pohl u. Rupp, 2015, S.21). Gemeint sind damit also alle Menschen, die in irgendeiner Weise mit der Software zu tun haben oder haben werden, z.B. der Kunde, der Nutzer, die Entwickler etc.

Da der Auftraggeber oft nicht in der Lage ist genau auszudrücken, was er will und braucht, ist es die Aufgabe des Requirement Engineers „in Software-Projekten frühzeitig mit dem Kunden und den späteren Nutzern [zu] reden, um zu erfahren, was sie sich vorstellen“ (Kleuker, 2016, S.15).

Es muss also eine ständige Kommunikation und Zusammenarbeit sichergestellt werden. So eine Zusammenarbeit setzt „ein gewisses Verständnis der Arbeitsabläufe des Kunden, genauer dessen Geschäftsprozesse (vgl. Gadatsch, 2010), [voraus], die mit der zu entwi-

ckelnden Software im Zusammenhang stehen“ (Kleuker, 2016, S.15).

### 3.3 Qualitätssicherung der Anforderungen

„Grundsätzlich gilt, dass die Produkte aller Tätigkeiten bei der Softwareentwicklung [...] qualitätsgesichert [...] werden müssen“ (Winter, 1999, S.55). Deshalb „ist es notwendig, die Qualität der [...] Anforderungen zu überprüfen“ (Pohl u. Rupp, 2015, S.95), um Probleme der Anforderungsermittlung (siehe Kapitel 2.3) zu lösen.

Diese ständige Qualitätssicherung dient dazu, Fehler möglichst früh innerhalb des Softwarelebenszyklus zu erkennen und zu beheben. Der Grund dafür ist, dass die Fehlerbehebung in der Softwareentwicklung mit steigendem Projektfortschritt mehr Aufwand verursacht (vgl. Hußmann, 2001, S.2). Aufwand kann dabei Geld, Zeit oder auch andere zu erbringende Leistung oder Einsatz sein. Das Ziel ist daher, Fehler schon möglichst früh in dem Projektverlauf zu beseitigen.

Dieser Aufwand der Fehlerbehebung ist der Grund für die Qualitätssicherung der Anforderungsermittlung und dadurch ist sie so wichtig für den Erfolg eines Projektes. Diese wichtige Rolle wird in dem folgenden Balkendiagramm dargestellt:

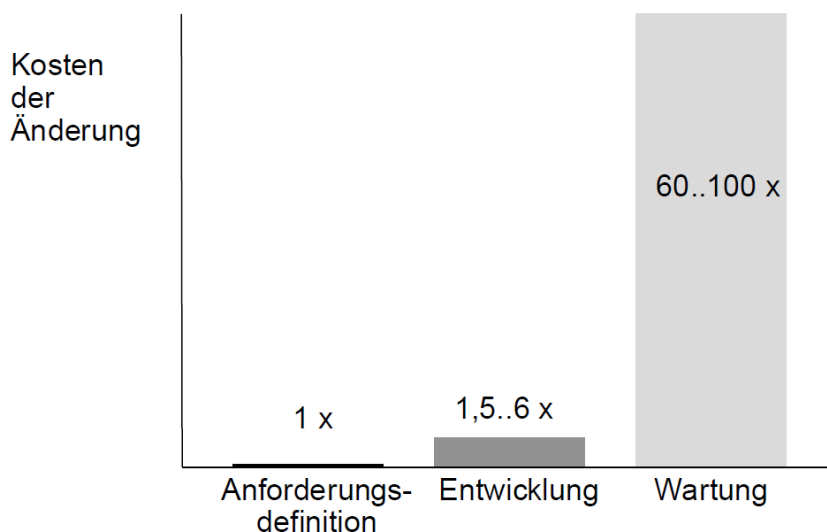


Abbildung 2: Aufwand der Fehlerbehebung in Softwareprojekten  
(Hußmann, 2001, S.2)

Das Diagramm in der Abbildung beschreibt den Aufwand der Fehlerbehebung in einem Softwareprojekt in Abhängigkeit von dem Projektfortschritt. Dieser Fortschritt wird dabei



in den drei Kategorien Anforderungsdefinition, Entwicklung und Wartung angegeben, die Abschnitte in fast jedem Software-Lebenszyklus sind.

Wird ein Fehler in der Phase der Anforderungsermittlung oder am Anfang des Projektes behoben, wird der Aufwand dafür mit dem Faktor Eins multipliziert. Sobald ein Fehler gefunden wurde, gilt es die Anforderung zu überarbeiten und ihn einfach zu korrigieren. Während der Entwicklung des Projektes ist der Aufwand meist um das 1,5 bis 6-fache höher. Grund dafür ist, dass Fehler, die sich dann noch in der Software befinden, oft mehrere Bereiche betreffen und viel verändert und berücksichtigt werden muss bevor man sie endgültig beheben kann.

Am meisten Aufwand verursacht die Fehlerbehebung, falls der Fehler erst am Ende des Software-Lebenszyklus entdeckt wird. Dabei ist die Software schon in Betrieb und befindet sich in der Wartung. Um dort einen Fehler zu beheben, müssen oft ganze Programmabschnitte verändert werden, damit die Software wieder fehlerfrei läuft. Es hat also große Folgen, wenn ein Teil des Programmes während der Wartung verändert werden muss, weshalb der Aufwand etwa 60 bis 100 mal höher ist als zu Anfang des Projekts.

Deshalb ist es so wichtig, die einzelnen Phasen eines Software-Lebenszyklus sorgfältig zu bearbeiten und zu überprüfen, sodass keine Fehler auftreten oder diese schon so früh wie möglich erkannt werden. Es sollte daher nicht zu früh und unvorbereitet mit der Entwicklung begonnen werden. Andernfalls entsteht ein "Wegwerf-Prototyp", der „kein Ersatz für die Dokumentation und auch kein brauchbarer Bestandteil des finalen Systems“ (Kuhmann, 2012) und damit nicht Ziel dieses Projektes ist.

Obwohl es Ausnahmen gibt, lohnt sich der Aufwand in der Analyse meistens, da „viele schwerwiegende Fehler in den frühen Phasen IS-Entwicklung [Informationssystem-Entwicklung; Anmerk. d. Verf.] gemacht werden“ (Alpar u. a., 2016, S.316). Trotzdem wird in vielen Fällen zu voreilig mit der Entwicklung angefangen.

"Die Analyse der in einer Anforderungsdefinition festgelegten Anforderungen zielt letztlich darauf ab, Aufschluss über die Qualität der Anforderungsbeschreibung zu erhalten“ (Parsch, 2010, S.51). Kriterien für die Qualität lassen sich hauptsächlich in Inhalt, Dokumentation und Abgestimmtheit gliedern. Es wird sich also damit befassen, ob die Anforderungen vollständig, detailliert, passend dokumentiert und mit allen Stakeholdern abgestimmt sind. Für jeden der drei Qualitätsaspekte gibt es damit unterschiedliche Prüfkriterien (vgl. Pohl u. Rupp, 2015, S.97).

### 3.3.1 Qualitätsaspekt Inhalt

„Der Qualitätsaspekt »Inhalt« bezieht sich auf die Überprüfung von Anforderungen auf inhaltliche Fehler“ (Pohl u. Rupp, 2015, S.98). Dafür gibt es acht Prüfkriterien(vgl. Pohl u. Rupp, 2015, S.98):

*Vollständigkeit:* Wurden alle relevanten Anforderungen erfasst?

*Korrektheit:* Beschreibt jede Anforderung die dafür notwendigen Informationen?

*Verfolgbarkeit:* Können die Anforderungen verfolgt, also z.B. auf die Quelle zurückgeführt werden?

*Adäquatheit:* Enthalten die Anforderungen die Bedürfnisse und Wünsche der Stakeholder angemessen?

*Konsistenz:* Gibt es Widersprüche zwischen den Anforderungen?

*Vorzeitige Entwurfsentscheidungen:* Wurden Entwurfsentscheidungen vorweggenommen, die nicht durch Randbedingungen bestimmt sind?

*Überprüfbarkeit:* Können Abnahme- und Prüfkriterien anhand der Anforderungen definiert werden?

*Notwendigkeit:* Trägt jede Anforderung zu dem definierten Ziel bei?

### 3.3.2 Qualitätsaspekt Dokumentation

Bei der Dokumentation geht es darum, die „Anforderungen auf Mängel in der Dokumentation bzw. auf Verstöße gegen geltende Dokumentationsvorschriften“(Pohl u. Rupp, 2015, S.99) zu überprüfen. Hierbei gibt es folgende vier Prüfkriterien(vgl. Pohl u. Rupp, 2015, S.99f.):

*Konformität:* Wurden die Anforderungen in dem vorgeschriebenen Dokumentationsformat strukturiert und in der richtigen Modellierungssprache dokumentiert?

*Verständlichkeit:* Können die Anforderungen in dem gegebenen Kontext ggf. mithilfe eines Glossars verstanden werden?

*Eindeutigkeit:* Ist eine eindeutige Interpretation möglich?

*Konformität mit Dokumentationsregeln:* Sind vorgegebene Dokumentationsregeln und -richtlinien eingehalten worden?

### 3.3.3 Qualitätsaspekt Abgestimmtheit

Die Abgestimmtheit stellt sicher, dass keine “Mängel in der Abstimmung der Anforderungen unter relevanten Stakeholdern“(Pohl u. Rupp, 2015, S.100) vorliegen. Auch dafür gibt es folgende drei Prüfkriterien(vgl. Pohl u. Rupp, 2015, S.100) :

*Abstimmung:* Wurde jede Anforderung mit relevanten Stakeholdern abgestimmt?

*Abstimmung nach Änderungen:* Wurde jede Änderung der Anforderungen auch abgestimmt?

*Konflikte:* Wurden alle bekannten Konflikte gelöst?

### 3.4 Anforderungskategorisierung

Anforderungen werden kategorisiert und nach Wichtigkeit eingeteilt. Das ist hilfreich, da die Anforderungen unterschiedlich zur Zufriedenheit der Stakeholder beitragen (vgl. Pohl u. Rupp, 2015, S.24). Durch Kategorisierung lassen sie sich leichter einordnen, um Aufwand und Priorität besser abschätzen zu können.

In diesem Fall werden die Anforderungen nach dem Kano-Modell kategorisiert. Demnach gibt es drei Kategorien (Pohl u. Rupp, 2015, vgl. S.24):

**Basisfaktoren** sind selbstverständliche unterbewusste Systemmerkmale, die vorausgesetzt werden.

**Leistungsfaktoren** sind bewusste, explizit geforderte Systemmerkmale.

**Begeisterungsfaktoren** sind für den Stakeholder unbekannte Systemmerkmale, die er während der Benutzung als angenehme Überraschung entdeckt.

Mit der Zeit werden aus Begeisterungsfaktoren Leistungsfaktoren und aus Leistungsfaktoren Basisfaktoren, da der Nutzer sich an die Merkmale gewöhnt und sie irgendwann voraussetzt. Das Modell lässt sich grafisch darstellen:

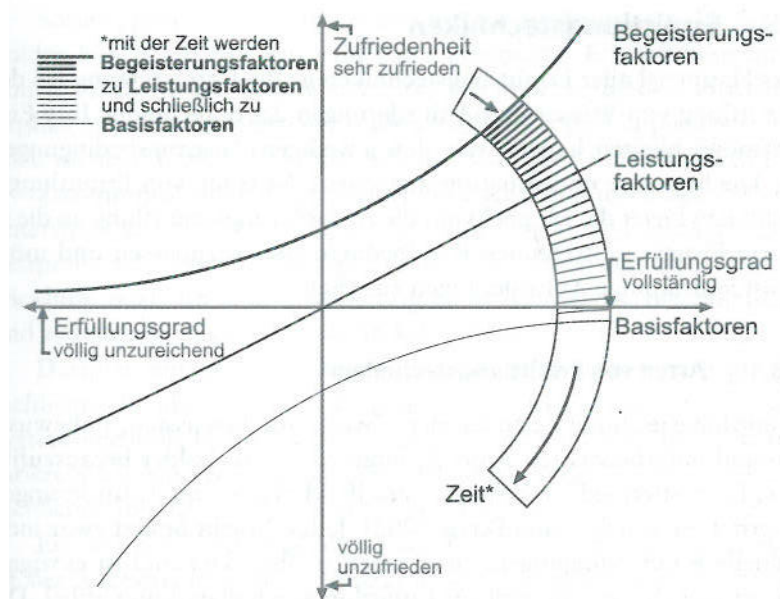


Abbildung 3: Grafische Darstellung des Kano-Modells

(Pohl u. Rupp, 2015, S.25)

Die Grafik zeigt die Zufriedenheit der Stakeholder in Abhängigkeit von dem Erfüllungsgrad der jeweiligen Faktoren. Auf einer dritten Achse wird die Zeit dargestellt, wodurch die Anforderungen die Kategorien wechseln können.

Die Basisfaktoren müssen erfüllt sein, um eine massive Unzufriedenheit der Stakeholder zu vermeiden (Zhu u. a., 2010, vgl. S.106). Sie werden vorausgesetzt und haben daher nur den Nutzen, Unzufriedenheit zu vermeiden. Daher verläuft die Funktion negativ exponentiell.

Die Funktion der Leistungsfaktoren verläuft linear und schneidet den Nullpunkt. Diese Faktoren erzeugen proportional Unzufriedenheit, wenn sie nicht erfüllt sind, und Zufriedenheit, wenn sie erfüllt sind (Zhu u. a., 2010, vgl. S.106).

Da die Begeisterungsfaktoren vorher den Stakeholdern nicht bekannt sind und damit nicht erwartet werden können, kann ihr unzureichender Erfüllungsgrad keine Unzufriedenheit verursachen (Zhu u. a., 2010, vgl. S.106). Das Vorhandensein dieser Faktoren freut jedoch die Stakeholder, wodurch es zu einem überproportionalen Nutzen bei steigender Erfüllung kommt. Deshalb ist die Funktion der Begeisterungsfaktoren exponentiell und erreicht niemals den Wert null oder weniger, aber steigt sehr schnell an.

Es ist ungeklärt, ob man diese Faktoren und ihre Eigenschaften so auf die Anforderungen bei der Erstellung eines Prototyps anwenden kann. Laut Pombergers Definition soll ein

explorativer Prototyp ermöglichen, die Vorstellungen der Anwender „anhand von Anwendungsbeispielen zu prüfen und die gewünschte Funktionalität zu ermitteln“ (Pomberger u. Pree, 2004, S.27), er soll also die Analyse und die Anforderungsspezifikation unterstützen. Das bedeutet, dass der Prototyp nicht alle Anforderungen enthalten kann, besonders die Leistungs- und Begeisterungsfaktoren nicht, und genau im Gegenteil dazu dient, Anforderungen zu ermitteln.

Kanos Begeisterungsfaktoren widersprechen zudem dem weit verbreiteten Vorgehen bei einer erfolgreichen Anforderungsanalyse. Wie von Pohl und Rupp beschrieben ist dafür „eine gute Kommunikation und [...] Qualität der Zusammenarbeit mit den Stakeholdern“ (Pohl u. Rupp, 2015, S.33) notwendig, um ihn „erfolgreich in den Ermittlungsprozess einzubinden“ (Pohl u. Rupp, 2015, S.34). Durch diese enge Zusammenarbeit werden alle Anforderungen abgesprochen, sodass es theoretisch kaum noch Begeisterungsfaktoren geben kann.

### **3.5 Abgrenzung des Systems und Systemkontextes**

Die Abgrenzung eines Systems zu seiner Umgebung und zu den Schnittstellen ist essentiell für den Entwurf eines korrekten Produktes. Dafür müssen die System- und Kontextgrenzen bestimmt werden.

Dadurch wird Aufwand vermieden, indem bereits vorhandene Funktionen evtl. integriert oder nur in etwas veränderter Weise übernommen werden. Außerdem bringt sie Klarheit über die Arten der Informationsgewinnung und -weiterverarbeitung, um Missverständnisse zu vermeiden. Es ist wichtig, „die Grenzen des Systems zum Systemkontext und die Grenzen des Systemkontexts zur irrelevanten Umgebung zu bestimmen“ (Pohl u. Rupp, 2015, S.20), da der Systemkontext „auch die Anforderungen an das zu entwickelnde System bestimmt“ (Pohl u. Rupp, 2015, S.20).

Da diese Arbeit COC als Grundlage nutzt sind auch die Systemgrenzen und -schnittstellen ähnlich. Wie bereits beschrieben muss nur die Schnittstelle zur CDB etwas verändert werden. Nach der Erarbeitung des Lösungsentwurfs ist es dann die Entscheidung des Kunden, ob CAB in COC eingliedert werden soll und die Abgrenzung damit größtenteils wegfällt oder ob System und Systemkontext für ein neues Programm neu bestimmt werden müssen.

### 3.6 Arten und Ziele des Prototypings

Prototyping beschreibt „eine Vorgehensweise bei der Softwareentwicklung, bei der nicht sofort ein endgültiges Softwaresystem, sondern zunächst ein oder mehrere Prototypen erstellt werden“ (Brich u. Hasenbalg, 2013, S.152). Dabei wird nach Kuhrmann zwischen horizontalem und vertikalem Prototyping unterschieden (vgl. Kuhrmann, 2012):

**Horizontales Prototyping** bezieht sich auf einen bestimmten Bereich der Software, z.B. die Benutzeroberfläche. Der Bezug zu der technischen Funktionalität und der tatsächlichen Implementierung ist dabei nicht gegeben. Nur die eine Ebene des Programms soll vorgestellt werden.

Beim **vertikalen Prototyping** wird ein Ausschnitt komplett in allen Ebenen vollständig implementiert. Diese Art dient zur Demonstration von komplexer Funktionalität.

Darüber hinaus unterscheidet die IEEE das Prototyping anhand von Anwendungszwecken (vgl. Lichter u. a., 1994, S.826):

**Exploratives Prototyping** wird benutzt, wenn man das Problem und die Anforderungen noch nicht genau kennt. Indem viele Ideen und Ansätze ausprobiert werden, lernt der Entwickler die Arbeit und die Anforderungen des Kunden besser kennen, um die Anforderungen zu identifizieren.

**Experimentelles Prototyping** zielt auf das Sammeln von Erfahrungen und Ideen. Der Anwender kann mit dem Prototypen experimentieren, um Ideen an das Programm weiterzuentwickeln und Anforderungen zu konkretisieren. Währenddessen enthält der Entwickler Eindrücke von der Realisierbarkeit des Systems und technischen Herausforderungen.

**Evolutionäres Prototyping** stellt die Softwareentwicklung nicht als ein temporäres Projekt, sondern als eine fortlaufende Entwicklung dar. Dabei wird das Programm nach und nach erweitert, wobei der Entwickler eng an der Seite des Anwenders das System immer weiter erweitert.

Die Definition des evolutionären Prototypings widerspricht jedoch der allgemeinen Definition nach Brich u. Hasenbalg (2013). Das Ergebnis des explorativen und des experimentellen Prototypings ist ein Prototyp im engeren Sinne, der der Demonstration dient. Dieser wird nach der Erfüllung seiner Aufgaben nicht mehr benötigt (vgl. Liggesmeyer, 2012,

S.21). Beim evolutionären Prototyping hingegen entsteht ein Pilotsystem, das der Kern des Produktes ist. Der Prototyp wird also fließend zum laufenden Produkt (vgl. Liggesmeyer, 2012, S.24) und ist damit eigentlich kein Prototyp laut Definition mehr, sondern schon ein endgültiges Softwaresystem.

## **4 Istanalyse**

### **4.1 Bestandsrechnung in COMPAS Cockpit**

Der aktuelle Algorithmus für die Bestandsrechnung in COC ermittelt zuerst den Bruttobestand der Besatzungen (vgl. DLH, 1995, S.8). Die Einheiten für alle Bestands- und Bedarfsrechnungen sind verfügbare Beschäftigungstage (BT) oder Stunden.

Der Bestand wird ermittelt, indem der aktuelle Pilotenbestand mit allen Formen von Zu- und Abgängen z.B. durch Umschulungen, Altersabgänge, Neueinstellungen oder Kündigungen sowie allen bezahlungswirksamen Fehlzeiten wie Teilzeit, Mutterschutz oder Fluguntauglichkeit errechnet wird (vgl. LSY, 2001, S.19). Insgesamt gibt es 16 Kategorien, die hier aber der Übersicht halber nicht alle genannt werden.

Von diesem Bruttobestand werden alle weiteren Fehlzeiten, die sich grob in Urlaub und Krankheit einteilen lassen, subtrahiert, was zu dem Nettobestand führt (vgl. DLH, 1995, S.8).

Das Ergebnis aller verfügbaren Piloten ergibt sich dann durch die Subtraktion von freien Tagen, die den Piloten zustehen. Auch die Arten und die Berechnung der freien Tage werden hier nicht weiter erläutert.

Dadurch ergibt sich der Bestand aller verfügbaren Piloten in BTs. Diese BTs lassen sich auf Tages-, Wochen- und Monatsbasis darstellen und zusammenfassen. Ein Screenshot der Bestandsrechnung aus COC befindet sich im Anhang A, wo weitere Details erkennbar sind.

### **4.2 Anforderungsanalyse- und kategorisierung**

Die Planerinnen formulierten folgende Anforderungen (vgl. Berg u. a., 2017):

Basisfaktoren: Es soll ein Programm erstellt werden, dass die Kapazitäts- und später auch die Schulungsplanung automatisch regelt. Dafür müssen vor allem die Gruppen und Untergruppen eingeteilt und dargestellt werden können. Der Zeitraum soll 15 Monate betragen, um die Urlaubsplanung des jeweils kommenden Jahres mit einfließen lassen zu können. Eine Bestandsrechnung wird also benötigt, die später um die Bedarfsrechnung ergänzt werden soll. Wird eine Integration in das COMPAS-Umfeld beschlossen, wird das Programm nach und nach erweitert bis es in etwa den Funktionalitäten von COC entspricht.



Der Kunde legt außerdem viel Wert auf Nachvollziehbarkeit. Es soll bei jeder Zelle der Tabellen verstanden werden können, woher die Daten kommen oder wie sie berechnet werden. Dazu soll es wie bei COC die Möglichkeit geben, sich einzelne Detailabfragen anzeigen lassen zu können.

**Leistungsfaktoren:** Da es sich um den Entwurf eines Prototyps zur Kapazitätsplanung handelt, ist die Bedarfsrechnung und die Schulungsplanung ein Leistungsfaktor. Dazu gehören auch Bewerbungen für Schulungen. Diese Leistungsfaktoren werden aber später ergänzt, sodass sie nicht Bestandteil dieses Lösungsentwurfs sind. Darüber hinaus wird eine gute Performance erwartet, um die Planung zu erleichtern und Zeit zu sparen.

**Begeisterungsfaktoren:** Modernes Design ist einer der Begeisterungsfaktoren. Dabei ist wünschenswert, das Programm so zu designen, dass sich Cockpit- und Kabinenplaner/innen ersetzen können und die Bedienung und Anordnung der Elemente gleich ist.

Da sich diese Arbeit aber nur mit der Entwicklung eines Lösungsentwurfs für den Prototypen beschäftigt, stehen die Basisfaktoren im Vordergrund. Die Leistungs- und Begeisterungsfaktoren sind in einem explorativen Prototypen noch nicht vorhanden. Sie kommen im Verlauf des Projektes dazu. Trotzdem ist eine Bestimmung davon wichtig, Kernfunktionen des Systems festzulegen und Schwerpunkte festzulegen.

### **4.3 Die Schnittstelle: Das Crew Management System**

Die Quelle, aus der COC die benötigten Daten entnimmt, ist die CDB, welche Bestandteil des Crew Management System (CMS) ist. Unter dem CMS wird das gesamte Planungssystem der Lufthansa verstanden, das aus vielen verschiedenen einzelnen Systemen, wie eine Crew Einsatzplanung oder einer Umlaufplanung, besteht. Im Mittelpunkt davon steht die CDB, in der Daten zusammen fließen und in einer Datenbank gespeichert werden. Es wird also von mehreren unterschiedlichen Systemen aus auf die CDB zugegriffen. Diese gleichzeitigen Lese- und Schreibzugriffe mehrerer Systeme müssen in kurzer Zeit von der CDB bearbeitet werden können.

Nachdem die Daten in der CDB, wie in Kapitel 2.1 beschrieben, aufbereitet wurden, werden die daraus entstandenen Views in die eigene COMPAS-Datenbank geladen, auf die COC zugreifen kann.

## **5 Konkrete Modernisierung und Anpassung von Compas Cockpit**

### **5.1 Veränderungen bei der Datenaufbereitung**

Um Laufzeit- und Performanceprobleme bei der Datenaufbereitung zu vermeiden, wird eine andere Vorgehensweise als in COC benötigt. Die einfachste Möglichkeit dazu ist, alle Rohdaten, aus denen die Views erstellt werden, unverändert von der CDB in den eigenen Teil der Datenbank von COC zu laden. Dort werden die Views erstellt von einem für COC dedizierten Server erstellt und dem Tool weitergegeben (vgl. dazu Becker u. a., 2017).

Dadurch hat die CDB nur noch die Aufgabe, die Tabellen der Rohdaten zur Verfügung zu stellen, und es wird ihr keine große Leistung abverlangt. Die Auslastung wird von der CDB auf den COMPAS-eigenen Server verschoben. Weil COC diesen Server nur nutzt, um kurzzeitige zyklische Aufgaben auszuführen, ist es kein Problem, diesen für eine längere Dauer auszulasten. Selbst bei starker Auslastung wird damit kein anderes System beeinträchtigt. Durch eine höhere Programmiersprache und einem effizienteren Algorithmus kann die Datenaufbereitung möglicherweise deutlich schneller erfolgen. Nach der Erstellung der Views werden alle daraus erhaltenen Daten wie bislang wieder in den COMPAS-eigenen Teil der CDB geladen, worauf das Programm darauf zugreift.

Bei der Datenaufbereitung wird mit personenbezogenen Daten gearbeitet, weshalb beim Erstellen der Views „die Bestimmungen des Bundesdatenschutzgesetzes“ (Maier, 2005, S.6) und weitere „spezielle Schutzvorschriften [,die] zwischen den Geschäftsleitungen und den Mitarbeitervertretungen [der Lufthansa AG; Anmerk. d. Verf.] definiert [wurden]“ (Maier, 2005, S.6), gelten. Obwohl für beide Systeme die gleichen Sicherheitsvorkehrungen gelten, musste diese Vorgehensweise diskutiert und eine Genehmigung benötigt werden.

### **5.2 Veränderungen in der Bestandsrechnung**

In der in Kapitel 4.1 beschriebenen Bestandsrechnung von COC müssen einige Zeilen angepasst werden, damit sie der bisherigen Bestandsrechnung der Kabine entspricht. Diese Anpassungen wurden mit dem Kunden detailliert besprochen (vgl. Winkel u. a., 2017).

Einige Zeilen wie „Verlängerer Teilzeit“ oder „Alterssonderurlaub“ fallen weg, da so etwas bei der Kabinenbesatzung nicht vorkommen kann. Eine Änderung wird für das Feld „Umschulungen“ benötigt, das in COC aus der bereits implementierten Schulungsplanung berechnet wurde. In CAB ist es notwendig, zwischen Schulungen, die kürzer als fünf Tage, und welchen, die mindestens fünf Tage dauern zu unterscheiden, weshalb zwei Zeilen für eine manuelle Eingabe beider Arten benötigt werden.

Bei der restlichen Bestandsrechnung kann genau so vorgegangen werden, wie es bereits in COC für die Cockpit-Mitarbeiter implementiert ist.

### **5.3 Abbilden der Mehrfachqualifikationen und Einbindung der Kleingruppen**

Nach Absprache mit dem Kunden über das Problem der Mehrfachqualifikationen wurde dafür folgender Lösungsansatz gefunden (vgl. Wagner u. a., 2017a):

Es soll die Möglichkeit geben, die KGs manuell PUs zuzuordnen. Die KGs sind auf der CDB definiert und verfügbar, weshalb sie nicht vom Nutzer bearbeitet werden können. Es soll für ihn aber möglich sein, PUs anzulegen, zu bearbeiten und zu entfernen. Jede KG wird dann vom Nutzer genau einer PU zugeordnet.

Deshalb ist eine Verbindung zwischen einer Person und einer PU nur über die KG möglich, der sowohl die Person als auch die PU zugewiesen ist. Jede Person wird also anstatt einer PU mit einer KG verknüpft. Danach ist eine neue Tabelle notwendig, in der die KGs den PUs zugeordnet werden, welche vom Anwender bearbeitet werden kann. Die Mehrfachqualifikationen fließen somit in das Konzept der PUs ein und es kann für jede PU der Bestand ermittelt werden. Eine eindeutige Bedarfsrechnung ist mit dieser Methode aber noch nicht möglich.

## **6 Zusammenfassung und Ausblick**

### **6.1 Änderungen an COMPAS Cockpit um es für eine automatisierte Kapazitätsplanung der Kabinenbesatzung zu nutzen**

Um auf Grundlage von COC die Kapazitätsplanung der Kabinenbesatzung zu automatisieren, müssten einige Dinge des Programms angepasst werden:

Die Funktionsweise der Datenaufbereitung muss aufgrund der höheren Anzahl an Quelldaten und ihrer Aufbereitung verändert werden. Eine nahe liegende Lösung dafür ist, die Datenaufbereitung von der CDB auf einen anderen Server zu verlegen, der nur von COC genutzt wird. Dazu kann der Algorithmus durch Nutzung spezieller Programmierbibliotheken verbessert werden.

Um Mehrfachqualifikationen der Kabinenbesatzung für Flotten sinnvoll in das Programm mit aufzunehmen, werden die KGs, die die Qualifikationen enthalten vom Nutzer selbst PUs zugewiesen. Dadurch ist eine eindeutige Bestandsrechnung möglich. Die Bestandsrechnung an sich ist für die Kapazitätsplanung der Cockpit- und der Kabinenbesatzung nahezu dieselbe.

Des Weiteren ist es nötig, viele Kleinigkeiten, z.B. Datenfelder oder Prämissen für eine Prognosenberechnung, anzupassen und detailliert mit dem Kunden abzustimmen. Daraus ergibt sich, welche Funktionen nicht mehr oder zusätzlich benötigt werden. Die hauptsächlichen Funktionen bleiben aber im Vergleich zu COC unverändert.

Durch die Ähnlichkeit der Kapazitätsplanung in beiden Bereichen, ist es relativ einfach, CAB in COC einzugliedern und es als Vorlage zu nutzen. Mithilfe der in dieser Arbeit gefundenen Lösungsansätzen sollte es keine großen Schwierigkeiten mehr beinhalten, die Bestandsrechnung der Kabinenbesatzung zu automatisieren.

Das ursprüngliche Ziel, einen Lösungsentwurf für die gesamte Kapazitätsplanung der Kabinenbesatzung zu erstellen, wurde nicht erreicht, der hier entworfene Lösungsansatz für die Bestandsrechnung dient aber als Grundlage dafür.

### **6.2 Probleme bei der Anpassung der Bedarfsrechnung der Kabinenbesatzung nach dem Vorbild von COMPAS Cockpit**

Nachdem die Bestandsrechnung relativ einfach angepasst werden kann, um den Bestand des Kabinenpersonals zu ermitteln, entstehen bei der Anpassung der Bedarfsrechnung

weitere Probleme. Der Bedarf wird als nächster Schritt benötigt, um aus Bestand und Bedarf die Kapazität als Differenz zu ermitteln. Bekommt die Lufthansa ein neues Flugzeug, beispielsweise einen Airbus A340, wird für dieses Flugzeug eine Cockpitbesatzung benötigt. Dieser Bedarf fällt auf eindeutige PUs zurück, die genau die Piloten beinhalten, die in den benötigten Funktionen eine A340 fliegen. Da jede PU des Cockpits eindeutig ist, kann der Bedarf einfach zugeordnet werden.

Aufgrund der Mehrfachqualifikationen entstehen bei der Bedarfszuordnung der Kabinenbesatzung aber Schwierigkeiten. Für die Kabinenbesatzung gibt es für eine Rolle z.B. eine PU1, in denen die Leute auf einer A380 und einer A340 fliegen, aber auch eine PU2, in denen sie auf A340 und Boeing 747 eingeteilt sind. Entsteht nun ein Bedarf an Kabinenpersonal für eine A340 ist es unklar, ob dieser aus PU1 oder PU2 abgedeckt wird.

Ein erster Lösungsansatz wäre es, den Bedarf prozentual auf alle möglichen PUs zu verteilen und diese zu gewichten. Wie genau damit umgegangen wird, muss aber mit dem Kunden ausführlich diskutiert werden, damit man sich auf eine Lösung einig wird.

### **6.3 COMPAS Cockpit als Vorlage - kritische Auseinandersetzung**

Kurzfristig gesehen ist die Integration in COC eine Lösung, mit der man mit wenig Aufwand das gewünschte Ziel, eine automatisierte Kapazitätsplanung der Kabinenbesatzung, erreichen kann.

Dieser Lösungsansatz ist jedoch kaum nachhaltig. COC ist technisch gesehen auf einem fast 20 Jahre alten Stand. Gerade in der IT-Branche, die sich sehr schnell weiterentwickelt, ist dies eine sehr lange Zeit. Für die Gegenwart reicht es bis jetzt noch, aber in einigen Jahren können sich die Anforderungen an ein modernes, komplexes Programm und seinen Funktionen so stark verändern, dass das alte COC dafür nicht mehr ausreicht. Deshalb wäre es vergeudeter Aufwand, den einfachen Weg zu gehen, um in einigen Jahren trotzdem ein fast komplett neues und modernes Programm zu entwickeln, wenn festgestellt wird, dass die Grenzen von CAB überschritten werden.

Es sollte lieber jetzt Aufwand in ein neues, eigenständiges und modernes Programm investiert werden; früher oder später reicht die bisherige Lösung nicht mehr aus und doppelte Arbeit bleibt dadurch erspart. Außerdem könnte CAB vielleicht auch ein Vorbild für eine Modernisierung von COC sein. Es ist im Sinne der Lufthansa, langfristig die beste Lösung in allen Bereichen zu finden, um auch in Zukunft ein einflussreiches und erfolgreiches Unternehmen zu bleiben.

## **6.4 Weitere Vorgehensweise**

Zunächst müssen Probleme und Unterschiede bei der Bedarfsrechnung diskutiert und geklärt werden. Sobald es eine Lösung gibt, diese anzupassen, kann auf Grundlage darauf die Deltarechnung erstellt werden. Das entspricht dem größten Teil der Kapazitätsplanung. Hinzu kommen noch Detailabfragen und Prämissen, die abgesprochen werden müssen, bevor die Kapazitätsplanung nach Vorbild von COC in einem Prototypen implementiert werden kann. Als nächstes wäre es sinnvoll, eine automatisierte Lösung für die Schulungsplanung der Kabinenbesatzung zu entwickeln.

Der Prototyp dient als Entscheidungshilfe, um die wahrscheinlich wichtigste Anforderung zu klären. Die Stakeholder können sich damit vor Augen führen, wie gut COC als Vorbild für die Kapazitätsplanung der Kabinenbesatzung geeignet ist. Sie müssen dann entscheiden, ob sie diese in COC integrieren oder in ein größeres Projekt investieren, um eine komplett neue Software zu entwickeln.

# Literatur

- [Alpar u. a. 2016] ALPAR, Paul ; ALT, Rainer ; BENSBERG, Frank ; GROB, Heinz L. ; WEIMANN, Peter ; WINTER, Robert: *Anwendungsorientierte Wirtschaftsinformatik : Strategische Planung, Entwicklung und Nutzung von Informationssystemen*. 8.Auflage. Wiesbaden : Springer Vieweg, 2016 ( 477152740). – ISBN 978–3–658–14146–2
- [Becker u. a. 2017] BECKER, Torsten ; BERGER, Iwan ; GARSKE, Julian: *Gespräch über Veränderungen bei der Datenaufbereitung*. Raunheim, 26. Juni 2017
- [Berg u. a. 2017] BERG, Stefanie ; WAGNER, Vanessa J. ; GARSKE, Julian: *Erstes Gespräch mit den Planerinnen über die aktuelle Vorgehensweise*. Frankfurt am Main, 24. Mai 2017
- [Berger 2016] BERGER, Iwan: *Anwendungsdokumentation gemäß CMS-BetrVbg; COMPAS - Kapazitätsplanung Cockpit*. Version 1.20. Raunheim: Lufthansa Systems GmbH & Co.KG, 4. Mai 2016
- [Brich u. Hasenbalg 2013] BRICH, Stefanie ; HASENBALG, Claudia: *Kompakt-Lexikon Wirtschaftsinformatik : 1.500 Begriffe nachschlagen, verstehen, anwenden*. Wiesbaden : Springer Gabler, 2013. – ISBN 978–3–658–03028–5
- [DLH 1995] DLH: *Funktionale Beschreibung Kapazitätsplanung Cockpit*. Version 0.1 - Entwurf. Frankfurt am Main: Deutsche Lufthansa AG, 20. März 1995
- [Gadatsch 2010] GADATSCH, Andreas: *Grundkurs Geschäftsprozess-Management*. Wiesbaden : Vieweg + Teubner, 2010. – ISBN 978–3–8348–9346–8
- [Hentschel u. a. 2017] HENTSCHEL, Norman ; RAPP, Dominik ; GARSKE, Julian: *Gespräch über die Datenbank-Schnittstelle*. Raunheim, 21. Juni 2017
- [Hußmann 2001] HUSSMANN, Heinrich: *Anforderungsermittlung; Vorlesungsskript in „Softwaretechnologie 2“*. Dresden: Technische Universität Dresden, 2001. – URL: <http://st.inf.tu-dresden.de/Lehre/WS00-01/st2/vorlesung/st2k2a-extra.pdf>; abgerufen am 05.07.2017
- [Kleuker 2016] KLEUKER, Stephan: *Grundkurs Datenbankentwicklung - Von der Anforderungsanalyse zur komplexen Datenbankabfrage*. 4. Auflage. Wiesbaden : Springer Vieweg, 2016. – ISBN 978–3–658–12338–3

- [Kuhrmann 2012] KUHRMANN, Marco: Prototyping. In: *Enzyklopädie der Wirtschaftsinformatik, Online-Lexikon*. München : Hrsg.: Norbet Gronau, Jörg Becker, Elmar J. Sinz, Leena Suhl und Jan Marco Leimeister, September 2012. – URL: <http://www.encyklopaedie-der-wirtschaftsinformatik.de/lexikon/is-management/Systementwicklung/Vorgehensmodell/Prototyping/index.html>; abgerufen am 26.06.2017
- [Lichter u. a. 1994] LICHTER, Horst ; SCHNEIDER-HUFSCHEIDT, Matthias ; ZÜLLIGHOVEN, Heinz: Prototyping in Industrial Software Projects-Bridging the Gap Between Theory and Practice. In: *IEEE Transactions on Software Engineering* Vol. 20 (1994), S. 825–832
- [Liggesmeyer 2012] LIGGESMEYER, Peter: *Software Entwicklung 2 - Prozessmodelle; Vorlesungsskript*. Kaiserslautern: Technische Universität Kaiserslautern, 2012. – URL: [http://agde.cs.uni-kl.de/teaching/se2/ss2012/material/vorlesung/se2\\_ss2012\\_03\\_Prozessmodelle.pdf](http://agde.cs.uni-kl.de/teaching/se2/ss2012/material/vorlesung/se2_ss2012_03_Prozessmodelle.pdf); abgerufen am 27.06.2017
- [LSY 2001] LSY: *Anhang zum Benutzerhandbuch - Berechnungen*. Version 1.2. Frankfurt am Main: Lufthansa Systems GmbH & Co.KG, 26. September 2001
- [M2P 2016] M2P CONSULTING (Hrsg.): *Projektergebnisse der Prüfung von M2P*. Frankfurt am Main: M2P Consulting, Februar 2016
- [Maier 2005] MAIER, P.: *Security Handbuch für das dezentrale CMS (CMS/D)*. Version 9.0. Frankfurt am Main: Deutsche Lufthansa Aktiengesellschaft, 2. Mai 2005
- [Partsch 2010] PARTSCH, Helmuth A.: *Requirements-Engineering systematisch : Modellbildung für softwaregestützte Systeme*. 2. überarbeitete und erweiterte Auflage. Berlin, Heidelberg : Springer-Verlag, 2010. – ISBN 978–3–642–05358–0
- [Pohl u. Rupp 2015] POHL, Klaus ; RUPP, Chris: *Basiswissen Requirements Engineering*. 4. Edition. Heidelberg : dpunkt.verlag, 2015. – ISBN 978–3–89864–550–8
- [Pomberger u. Pree 2004] POMBERGER, Gustav ; PREE, Wolfgang: *Software Engineering: Architektur-Design und Prozessorientierung*. 3., völlig überarbeitete Auflage. Carl Hanser Verlag GmbH & Co.KG, 2004. – ISBN 978–3–446–22788–0



- [Schatten u. a. 2010] SCHATTEN, Alexander ; DEMOLSKY, Markus ; WINKLER, Dietmar ; BIFFL, Stefan ; GOSTISCHA-FRANTA, Erik ; ÖSTREICHER, Thomas: *Best Practice Software-Engineering*. Heidelberg : Spektrum Akademischer Verlag, 2010. – ISBN 978-3-8274-2487-7
- [Wagner u. a. 2017a] WAGNER, Vanessa J. ; WINKEL, Alfred ; GARSKE, Julian: *Gespräch über Mehrfachqualifikationen und die Datenerhebung*. Lufthansa-Basis Frankfurt am Main, 26. Juni 2017
- [Wagner u. a. 2017b] WAGNER, Wolfgang ; BECKER, Torsten ; RAPP, Dominik ; GARSKE, Julian: *Erstes Gespräch über Compas Kabine*. Raunheim, 22. Mai 2017
- [Winkel 2017] WINKEL, Alfred: *High Level Items - Personalbestandsermittlung Kabine*. Version 1.3. Frankfurt am Main: Deutsche Lufthansa AG, 20. Februar 2017
- [Winkel u. a. 2017] WINKEL, Alfred ; WAGNER, Vanessa J. ; GARSKE, Julian: *Gespräch über die detaillierte Bestandsrechnung und die Datenbank-Strukturen*. 4. Juli 2017. – Aufzeichnung: siehe Anhang B
- [Winter 1999] WINTER, Mario: *Qualitätssicherung für objektorientierte Software: Anforderungsermittlung und Test gegen die Anforderungsspezifikation*, Fachbereich Informatik der FernUniversität - Gesamthochschule - in Hagen, Doktorarbeit, 1999
- [Zhu u. a. 2010] ZHU, Dauw-Song ; LIN, Chih-Te ; TSAI, Chung-Hung ; WU, Ji-Fu: A Study on the Evaluation of Customers Satisfaction - the Perspective of Quality. In: *International Journal for Quality Research* (2010), August, S. 105–115

# Glossar

Begriff	Erläuterung
Bestandsrechnung	Berechnung des verfügbaren Bestands in Beschäftigungstagen; dieser wird aus dem Bruttobestand und verschiedenen Ab- und Zugängen errechnet
COMPAS Cabin	Gefordertes Programm; soll in COMPAS Cockpit integriert werden und die Kapazitäts- und Schulungsplanung der Kabinencrew automatisieren
COMPAS Cockpit	Crew Operation Manpower Planning Advanced System; Programm zur Kapazitäts- und Schulungsplanung der Cockpit-Mitarbeiter; für die Lufthansa Passage von Lufthansa Systems 1999 entwickelt und seitdem ständig erweitert
Crew Database	Zentrale Datenbank der Lufthansa mit Schnittstellen zu vielen unterschiedlichen Systemen; stellt die erforderlichen Daten für COMPAS zusammen
Crew Management System	Überbegriff für alle Schnittstellen-Systeme der Lufthansa mit der Crew Database im Mittelpunkt
Deltarechnung	Der Bestand wird ins Verhältnis zum Bedarf gesetzt und daraus das Delta ermittelt
Funktion	auch Function; Rolle und Aufgaben, die Piloten oder die Kabinenbesatzung auf einem Flug übernehmen
Homebase	auch Standort; Flughafen an dem eine Person oder Gruppe stationiert ist
IEEE	Institute of Electrical and Electronics Engineers; Berufsverband von Ingenieuren aus Elektro- und Informationstechnik
Kapazitätsplanung	Mithilfe der Deltarechnung sollen dabei die richtige Anzahl an Personen mit den richtigen Qualifikationen zum richtigen Zeitpunkt verfügbar sein

Kleingruppe	Gruppen für die Kap.- und Schulungsplanung der Kabinenebesatzung; definiert durch Funktion, Homepage, Unterfunktion, Fluggesellschaft und Flugzeugtyp(en)
Planungseinheit	auch Planningunit; Gruppen für die Kap.- und Schulungsplanung des Cockpits; definiert durch Flugzeugtyp, Funktion, Homepage, Fluggesellschaft, Unterfunktion
Unterfunktion	auch Subfunction; optionale Erweiterung der Funktion

# Anhang

## A Detaillierte Bestandsrechnung in COC - Screenshot

## B Gesprächsprotokoll vom 4. Juli 2017

### Dokumentation der Meeting-Ergebnisse vom 04.07.2017

Teilnehmer: Alfred Winkel, Vanessa Wagner, Julian Garske

Thema der Besprechung: Veränderungen in den Datenbank-Strukturen und in der Bestandsrechnung im Vergleich zu Compas Cockpit

Veränderungen der Views COMPAS.COMPAS und COMPAS.COMPAS\_CRM auf der CDB:

COMPAS.COMPAS: Kumulierte tagesgenaue Werte

- Ggf. um Aircraft 2 und 3 als optionale Felder ergänzen
- Wie wird mit unterschiedlichen Teilzeitmodellen umgegangen?
- Teilzeit-Verlängerer und Alterssonderurlaub fallen weg
- Die Schulung BASE\_CHECK fällt weg

COMPAS.COMPAS\_CRM: Personifizierte tagesgenaue Werte

- An Stelle der PU muss auf die KG verwiesen werden
- TO\_DO, Teilzeit-Verlängerer, Alterssonderurlaub und BASE\_CHECK fallen weg
- Teilzeitmodelle?

Es muss eine dritte Tabelle erstellt werden mit allen KGs, die den PUs zugeordnet werden können.

KG	PU
1	A380/A340-P2-FRA-DLH-ALL
2	A380/A340/A320-P1-FRA-DLH-ALL
3	(keine Zuordnung)
4	A380/A340/A320-QFB_ANÜ-FRAU-DLH-SMK

## Veränderungen bei der Bestandsrechnung:

ET-Bestand (detailliert)		Mrz 2017	Apr 2017	Mai 2017	Jun 2017	Jul 2017
Piloten Gesamt(BK)		638,0	645,0	650,0	645,0	639,0
davon Piloten (BK)		611,0	618,0	625,0	619,0	612,0
Verlaengerer (BK)		27,0	27,0	25,0	26,0	27,0
Verlaengerer Pot. (BK)		0,0	0,0	0,0	0,0	0,0
= Brutto (BK)	▲	638,0	645,0	650,0	645,0	639,0
+ Ausgeliehene (BK)		0,0	0,0	0,0	0,0	0,0
+ Neueinstellungen (BK)		0,0	0,0	0,0	0,0	0,0
+ Umschulungen-Zugang (BK)		3,0	0,0	0,0	0,0	0,0
+ Homebasewechsler (BK)		7,0	7,0	0,0	0,0	0,0
- Umschulungsabgang (BK)		0,2	0,0	2,3	2,9	1,0
- Homebasewechsler (BK)		0,0	0,0	0,0	0,0	0,0
- Kuendigungen/Sterbefaele (BK)		0,0	0,0	0,0	0,0	0,0
- Altersabgaenge (BK)		0,1	0,1	0,0	0,0	0,0
- Fluguntauglichkeit (BK)		3,0	2,0	2,0	2,0	2,0
Schnittstelle/Hochrechnung (BK)		1,0	1,0	1,0	1,0	1,0
+ geplant (BK)		0,0	0,0	0,0	0,0	0,0
- Abgeordnete gesamt (BK)	▲	1,0	1,0	1,0	1,0	1,0
- Mutterschutz+Erziehungsurlaub (BK)		1,2	1,2	1,3	1,2	1,0
Elternzeit (Block)		0,6	1,5	1,0	1,1	1,0
+ Elternzeit (monatsreduziert)		1,0	1,0	1,0	1,0	1,0
- Elternzeit (gesamt)	▲	1,6	2,5	2,0	2,0	2,0
Elternteilzeit (Block)		2,0	12,0	3,0	7,0	4,0
+ Elternteilzeit (monatsreduziert)		1,8	1,8	1,8	1,8	1,0
- Elternteilzeit (gesamt)	▲	3,8	13,8	4,8	8,8	5,0
Teilzeit (Block)		14,5	12,1	10,4	9,7	9,0
+ Teilzeit (monatsreduziert)		12,4	12,1	12,4	12,4	12,0
- Teilzeit (gesamt)	▲	27,0	24,2	22,9	22,2	22,0
Verlaengerer Teilzeit (Block)		0,5	0,0	0,0	0,0	0,0
+ Verlaengerer Teilzeit (monatsreduziert)		0,2	0,2	0,2	0,2	0,0
- Verlaengerer Teilzeit (gesamt)	▲	0,7	0,2	0,2	0,2	0,0
- Verlaengerer Teilzeit Pot.(BK)		0,0	0,0	0,0	0,0	0,0
- Langzeitkranke (BK)		15,0	13,0	13,0	13,0	13,0
- unbezahlter Urlaub (BK)		0,0	0,0	0,0	0,0	0,0
- Umschueler (BK)		4,0	3,2	3,0	3,0	2,0
- Checker in Ausbildung (BK)		0,0	0,0	0,0	0,0	0,0
+/- anderer Bestand (BK)		-3,0	-3,0	-3,0	-3,0	-3,0
= Aktiv (BK)		587,5	587,8	594,5	585,6	585,0
= Aktiv (BT)	▲	18.213,2	17.634,2	18.430,9	17.567,7	18.154,0
- Erholungsurlaub (BT)		1.565,0	2.060,0	1.882,0	1.978,0	2.468,0
- Alterssonderurlaub (BT)		336,0	0,0	0,0	0,0	0,0
- Bildungsurlaub (BT)		29,0	28,0	30,0	28,0	29,0
+/- Urlaubsboerse (BT)		0,0	0,0	0,0	0,0	0,0
- Krank (REP)		549,3	385,6	283,0	262,3	288,0
- Kur (BT)		59,9	56,4	60,6	56,2	59,0
= Netto (BT)	▲	15.673,9	15.104,1	16.175,2	15.243,2	15.309,0
- bezahlter Sonderurlaub (BT)		22,0	15,0	4,0	0,0	2,0
- geplante freie Tage (BT)		6.485,0	6.397,0	6.576,0	6.398,0	6.275,0
- FZ-Tage (BT)		530,0	0,0	0,0	0,0	0,0
- Funktionaere / DT Tage		158,2	154,8	161,2	154,8	158,0
- PV-Tage		79,1	77,4	80,6	77,4	79,0
+ FO-Leistung gesamt(BT)	▶	0,0	0,0	0,0	0,0	0,0
= Verfuegbar (BT)	▲	8.399,6	8.459,9	9.353,4	8.613,0	8.794,0

Bisherige Bestandsrechnung in Compas Cockpit. Folgende Felder werden nicht mehr benötigt:

- Verlaengerer Pot, Homebasewechsler, Verlängerer Teilzeit, Checker in Ausbildung (für die Ausbildung der Crew muss noch eine Lösung gefunden werden), Alterssonderurlaub

Zu ändernde Felder:

- Bei Schulungen muss aufgrund unterschiedlicher Regelungen für freie Tage zwischen Schulungen mit einer Dauer von <5 und >4 Tagen unterschieden werden

## C Beispiele für Kleingruppen und Planungseinheiten

Kleingruppen- Kennungen (Stand ab 01.08.2016)									
<b>Purser II</b>	Request-restriktion	747/340	ANÜ CLH			380/340	ANÜ CLH	DUS	MUC
Linienpersonal	keine	W1				P1			M1
Linienpersonal (ANÜ CLH)	keine		A1				C1		
Linienpersonal (SiP FRW)	keine	A2					C2		
max. 1 Langstrecken-Muster	keine	W2				P2			
Lehrer	keine	W3	A3			P3	C3		M3
CCMP (Cabin Crew Manager Purser)	siehe EK- Qualitätsanforderung	W4	A4			P4	C4		M4
MAOD (Mitarbeiter am OPS-Desk)	siehe EK- Qualitätsanforderung	W5	A5			P5	C5		M5
<b>Purser II Gem</b>	Request-restriktion	747/340 320	ANÜ CLH	747/340 737	ANÜ CLH	380/340 320	ANÜ CLH	340/320	340/320
Linienpersonal Gemischt PII	keine		D1		E1		G1		
max. 1 Langstrecken-Muster	keine		D2		E2		G2		
only LR	keine		D6		E6		G6		
only SR	keine		D7		E7		G7		
Lehrer	keine		D3		E3		G3		
CCMP (Cabin Crew Manager Purser)	siehe EK- Qualitätsanforderung		D4		E4		G4		
MAOD (Mitarbeiter am OPS-Desk)	siehe EK- Qualitätsanforderung		D5		E5		G5		
<b>Purser I</b>	Request-restriktion	747/340 320	ANÜ CLH	747/340 737	ANÜ CLH	380/340 320	ANÜ CLH	340/320	340/320
Linienpersonal	keine	VA	DA	WA	EA	PA	GA	KA	MA
max. 1 Langstrecken-Muster	keine	VB	DB	WB	EB	PB	GB		
only LR	keine	VR	DR	WR	ER	PR	GR	KR	MR
only SR	keine	VS		WS		PS		KS	MS
Lehrer	keine	VE	DE	WE	EE	PE	GE		ME
CCMP (Cabin Crew Manager Purser)	siehe EK- Qualitätsanforderung	VK	DK	WK	EK	PK	GK	KK	MK
MAOD (Mitarbeiter am OPS-Desk)	siehe EK- Qualitätsanforderung	VG	DG	WG	EG	PG	GG	KG	MG
Kontwechsler virtuell	keine	VI				PI			J1
<b>Firstclass-Flugbegleiter</b>	Request-restriktion	747/340 320	ANÜ CLH	747/340 737	ANÜ CLH	380/340 320	ANÜ CLH	340/320	340/320
Linienpersonal	keine	8Y	8D	8Z	8E	8X	8G	8L	8N
max. 1 Langstrecken-Muster	keine	8A	8H	8B	8I	8C	8K		
only LR	keine	8R	8T	8R	8T	8R	8T		
Linienpersonal SMK	keine	8U				8W			8M
max. 1 LR-Muster SMK	keine	8O				8P			
only LR SMK	keine	8S				8S			
<b>Flugbegleiter</b>	Request-restriktion	747/340 320	ANÜ CLH	747/340 737	ANÜ CLH	380/340 320	ANÜ CLH	340/320	340/320
Linienpersonal	keine	YA	SA	ZA	TA	QA	UA	LA	NA
max. 1 Langstrecken-Muster	keine	YB	SB	ZB	TB	QB	UB		
only LR	keine	YR	SR	ZR	TR	QR	UR	LR	NR
only SR	keine	YS		ZS		QS		LS	NS
FAZ-B	keine	YF		ZF		QF		LF	NF
Linienpersonal JAZ	keine	YJ				QJ			NJ
Linienpersonal SMK	keine	BA				CA			OA
max. 1 LR-Muster SMK	keine	BB				CB			
only LR SMK + JAZ	keine	BR				CR			OR
only SR SMK+ JAZ	keine	BS				CS			OS
Lehrer	keine	YO	SO	ZO	TO	QO	UO		NO
TLFB	siehe EK- Qualitätsanforderung	YK		ZK		QK		LK	NK
MAOD (Mitarbeiter am OPS-Desk)	siehe EK- Qualitätsanforderung	YG	SG	ZG	TG	QG	UG	LG	NG
Kontwechsler virtuell	keine	YI				QI			JA
<b>RTK</b>	Request-restriktion	747/340 320	ANÜ CLH	747/340 737	ANÜ CLH	380/340 320	ANÜ CLH	340/320	340/320
NFBL und Ready Entries mit Abw. >12 Monate	nur OFF- Request 6 Monate	1Y		1Z		1Q		1L	1N
JAZ	nur OFF- Request 6 Monate	5Y				5Q			5N
SMK	nur OFF- Request 6 Monate	1B				1C			1O
WE, und Ready Entries mit Abw. <12 Monate	nur OFF- Request 2 Monate	2Y	2S	2Z	2T	2Q	2U	2L	2N
WE FC	nur OFF- Request 2 Monate	3Y	3S	3Z	3T	3Q	3U		3O
WE SMK	nur OFF- Request 2 Monate	2B				2C			2O
FAZ-B mit Flugeinsatz Vorjahr ≤ 2 Monate	nur OFF- Request 2 Monate	4Y		4Z		4Q		4L	4N
<b>Regionale Flugbegleiter</b>	Request-restriktion	Thailand ex BKK 380/747	Indien ex DEL 747/340	Korea ex FRA 340	China ex FRA 380/340 747	Japan ex FRA 380/340 747		China ex MUC 340	Japan ex NRT 340
Linienpersonal	keine	RP	RG	RA	RD	RJ		RS	RM
NFBL	nur OFF- Request 6 Monate	RQ	RH	RB	RE	RK		RT	RN
Wiedereingliederung	nur OFF- Request 2 Monate	RR	RI	RC	RF	RL		RU	RO



**Planungseinheiten:****Join:****FRA**

P2-FRA-380/340
P2-ANÜ-FRA-380/340
P2-FRA-747/340
P2-ANÜ-FRA-747/340
gem_P2-FRA-380/340/320
gem_P2-ANÜ-FRA-380/340/320
gem_P2-FRA-747/340/320
gem_P2-ANÜ-FRA-747/340/320
P1-FRA-380/340/320
P1-ANÜ-FRA-380/340/320
P1-FRA-747/340/320
P1-ANÜ-FRA-747/340/320
P1-DUS
QFFB-FRA-380/340/320
QFFB-ANÜ-FRA-380/340/320
QFFB_SMK-FRA-380/340/320
QFFB_SMK-ANÜ-FRA-380/340/320
QFFB-FRA-747/340/320
QFFB-ANÜ-FRA-747/340/320
QFFB_SMK-FRA-747/340/320
QFFB_SMK-ANÜ-FRA-747/340/320
QFFB-DUS
FB-FRA-380/340/320
FB-ANÜ-FRA-380/340/320
FB_SMK-FRA-380/340/320
FB_SMK-ANÜ-FRA-380/340/320
FB-FRA-747/340/320
FB-ANÜ-FRA-747/340/320
FB_SMK-FRA-747/340/320
FB_SMK-ANÜ-FRA-747/340/320
FB-JAZ-747/340/320
FB-JAZ-380/340/320
FB-DUS
FB-BKK
FB-DEL
FB-JAF
FB-CIF
FB-KOF
Lost & Found

P2-FRA-all
------------

gem_P2-FRA-all
----------------

P1-all
--------

QFFB-all
----------

FB-all
--------

FB-QFFB-all
-------------

FB-Regio-all
--------------