



# Konzeption und Implementierung einer touchgesteuerten Oberfläche für einen konfiguratorbasierten Produktkatalog

**Bachelorthesis**

für die Prüfung zum  
**Bachelor of Science**

des Studienganges Angewandte Informatik  
an der Dualen Hochschule Baden-Württemberg Karlsruhe

von

**Dane Leube**

23.08.2013

**Bearbeitungszeitraum**  
**Matrikelnummer, Kurs**  
**Ausbildungsfirma**  
**Betreuer**  
**Gutachter**

12 Wochen  
1313394, TAI10B2  
CAS Software AG, Karlsruhe  
Dr. Michael Klein  
Dipl. Inform. Thorsten Schlachter

## **Erklärung**

gemäß § 5 (2) der „Studien- und Prüfungsordnung DHBW Technik“ vom 18. Mai 2009.

Ich habe die vorliegende Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet.

Karlsruhe, 23.08.2013

Dane Leube

## **abstract**

Die sogenannte Mass Customization ist ein Ansatz für die Herstellung eines individuellen Produktes. Voraussetzung für diese Art der Fertigung ist eine große Anzahl verschiedener Einzelbauteilen. Die Herausforderung besteht in der Vermittlung des komplexen Aufbaus eines solchen Produktes, so dass der Kunde eine individuelle Auswahl der Produktkomponenten durchführen kann. Gleichzeitig müssen komplexe Abhängigkeiten zwischen den Bauteilen beachtet werden, um ein Produkt technisch realisieren zu können. In dieser Arbeit wird eine Lösung aufgezeigt, die einen Touchscreen auf einem mobilen Endgerät für eine übersichtliche Darstellung der Produktstruktur in einem Produktkatalog verwendet. Gleichzeitig findet durch den Einsatz eines Produktkonfigurators eine Überprüfung der aktuellen Zusammenstellung statt. Das Zusammenführen dieser beiden Komponenten führt dabei zu einer Vereinfachung des gesamten Konfigurationsprozesses. Zusätzlich wird eine Beschleunigung des Prozesses durch den Einsatz eines mobilen Endgerätes und einer damit verbundenen Verfügbarkeit der Software beim Kunden erreicht.

# Inhaltsverzeichnis

<b>1. Einleitung</b>	<b>1</b>
1.1. Motivation . . . . .	1
1.2. Ziel der Arbeit . . . . .	1
1.3. Vorgehensweise . . . . .	2
<b>2. Grundlagen</b>	<b>3</b>
2.1. Definition Produkt . . . . .	3
2.1.1. Produktkatalog . . . . .	4
2.1.2. Boolesche Algebra in der Produktmodellierung . . . . .	5
2.2. Produktkonfiguratoren . . . . .	6
2.2.1. CAS Configurator Merlin . . . . .	7
2.2.2. Anwendungsbeispiel der Arbeit . . . . .	8
2.3. Mobile Anwendungen . . . . .	9
2.3.1. Native Anwendungen . . . . .	10
2.3.2. Web Anwendungen . . . . .	11
2.3.3. Hybride Anwendung . . . . .	12
<b>3. Analyse der Anwendung</b>	<b>13</b>
3.1. Aktueller Konfigurationsprozess . . . . .	13
3.1.1. Übertragung auf die Anwendungsdomäne Luftfahrt . . . . .	15
3.2. Workflow Modellierung . . . . .	17
3.2.1. Umstellung auf einen mobilen Konfigurationsprozess . . . . .	17
3.2.2. Soll-Prozess des Workflows im Anwendungsbeispiel . . . . .	18
3.2.3. Prozessbeteiligte der Anwendung . . . . .	19
3.3. Anforderungsanalyse . . . . .	20
3.3.1. Nicht-Funktionale Anforderungen . . . . .	21
3.3.2. Funktionale Anforderungen . . . . .	23
<b>4. Entwurf der Benutzerschnittstelle</b>	<b>26</b>
4.1. Auswahl der Anwendungsplattform . . . . .	26
4.1.1. Anwendungsform . . . . .	26
4.1.2. Anwendungstechnologie . . . . .	30

4.2.	Untersuchung der Plattform Windows 8 . . . . .	32
4.2.1.	Design Richtlinien . . . . .	32
4.2.2.	Bedienkonzepte . . . . .	33
4.2.3.	Touchoptimierte Bedienelemente . . . . .	34
4.3.	Entwurf der Ansichten . . . . .	35
4.3.1.	Hauptfunktion der Anwendung . . . . .	36
4.3.2.	Produktkatalog . . . . .	37
4.3.3.	Flugzeugauswahl . . . . .	39
4.3.4.	Konfigurationsergebnisse . . . . .	40
4.3.5.	Weitere Ansichten . . . . .	41
4.4.	Navigation und Bedienung . . . . .	43
4.4.1.	Navigationsverlauf . . . . .	44
4.4.2.	Bedienelemente . . . . .	45
4.5.	Expertenmodus . . . . .	46
4.5.1.	Einsatz der Flip View . . . . .	46
4.5.2.	Erweiterung des Semantischen Zooms . . . . .	47
<b>5.</b>	<b>Implementierung</b>	<b>48</b>
5.1.	Anwendungsarchitektur . . . . .	48
5.1.1.	Model-View-ViewModel . . . . .	49
5.1.2.	Anwenden des Entwurfsmusters . . . . .	50
5.2.	Navigation . . . . .	52
5.3.	Ergebnisse der Implementierung . . . . .	54
5.3.1.	Startseite und Flugzeugprogrammauswahl . . . . .	54
5.3.2.	Upgradeauswahl . . . . .	56
5.3.3.	Flugzeugauswahl . . . . .	57
5.3.4.	Konfigurationsansicht . . . . .	58
<b>6.</b>	<b>Evaluation der Anwendung</b>	<b>61</b>
6.1.	Durchführung der Evaluation . . . . .	61
6.1.1.	Funktionale Anforderungen . . . . .	61
6.1.2.	Nicht-Funktionale Anforderungen . . . . .	62
6.1.3.	Auswahl der Testmenge . . . . .	62
6.2.	Ergebnisse . . . . .	63
6.2.1.	Auswertung der Benutzerergebnisse . . . . .	63
6.2.2.	Auswertung der Expertenergebnisse . . . . .	66
6.3.	Konsequenzen aus der Evaluation . . . . .	68
<b>7.</b>	<b>Fazit und Ausblick</b>	<b>70</b>

<b>A. Anhang</b>	<b>72</b>
A.1. Quellcodeauszug . . . . .	72
A.2. Evaluationsbögen . . . . .	73
A.3. Zusätzliche Evaluationsergebnisse . . . . .	77
<b>Abbildungsverzeichnis</b>	<b>ii</b>
<b>Tabellenverzeichnis</b>	<b>iii</b>
<b>Literaturverzeichnis</b>	<b>iv</b>

# **1. Einleitung**

## **1.1. Motivation**

Die Entwicklung von der Massen-Produktion zur Massen-Individualisierung (engl. mass-customization) bei Produkten schreitet immer weiter voran.[Pin93]. Mit der höheren Produktvielfalt können auch individuelle Kundenwünsche bedient werden. Bedingt durch die hohe Komplexität, die durch diesen Trend notwendig ist, wird die Zusammenstellung des Produktes aufwändiger. Bisweilen sind die für die Durchführung einer Produkt-Individualisierung viel Zeit, Geld und personelle Ressourcen notwendig. Nach einer erfolgten Produktauswahl aus einem Produktkatalog in Papierform, wird die Zusammenstellung manuell geprüft, sodass der Kunde ein Feedback über die technische Realisierung erhält.

Für die Lösung dieses Problems werden zur Qualitätssteigerung und aus ökonomischen Gesichtspunkten heraus immer mehr computergestützte Systeme verwendet. Diese können innerhalb von Sekunden die Abhängigkeiten der Produkte berechnen und ein schnelles Feedback liefern. Für eine weitere Verbesserung des Prozesses werden bereits mobile Anwendungen auf sogenannten Tablet-PCs konzipiert, welche den Vorteil haben, dass die Lösung beim Kunden vor Ort eingesetzt werden kann. Diese Geräte und deren Anwendungen, Apps genannt, sind im Geschäftsumfeld zunehmend in verschiedenen Bereichen verbreitet[VL11]. Die Vorteile dieser Systeme sind neben der mobilen Verfügbarkeit eine Verwendung von Touchscreens. Diese bieten eine intuitive Bedienung, womit komplexe Sachverhalte vereinfacht durchgeführt werden können.

## **1.2. Ziel der Arbeit**

Im aufgezeigten Rahmen soll die vorliegende Bachelorarbeit eine Möglichkeit aufzeigen, wie eine komplexe Produktlandschaft für einen Kunden übersichtlich dargestellt

werden kann. Hierzu wird mithilfe eines Produktkataloges das Produkt aufbereitet und mit einer Konfigurationslösung die Zusammenstellung überprüft. Der Kunde soll hierdurch in der Lage sein, eine Konfiguration selbstständig durchzuführen.

Für die Unterstützung des Gesamtprozesses wird ein mobiles Endgerät verwendet. Hier soll der Einsatz einer touchgesteuerten Bedienung die Vereinfachung des Prozesses unterstützen. Damit eine bessere Vermittlung des Produktes stattfindet, muss die Anwendung unter Einsatz der technischen Möglichkeiten eine ansprechende Darstellung bieten.

Bei einem mobilen Einsatz der Anwendung sind Anpassungen an den Konfigurationsprozess nötig. Diese sollen die Effizienz beim Durchführen einer Konfiguration erhöhen. Die Herausforderung besteht in der Umsetzung der angepassten Prozesse als mobile Anwendung. Hier sollen Heuristiken beachtet werden, die eine Integration in die Zielumgebung ermöglicht.

#### Ziel:

Vereinfachung des Konfigurationsprozesses eines komplexen Produktes durch den Einsatz einer touchgesteuerten Benutzerschnittstelle.

## 1.3. Vorgehensweise

Ausgangspunkt der Arbeit ist eine ausgiebige Analyse des Ist-Zustandes eines Konfigurationsprozesses. Dieser Prozess wird für einen neuen mobilen Workflow angepasst. Für die resultierende Arbeit wird ein Anwendungsbeispiel verwendet. An diesem werden die Veränderungen des Prozesses gezeigt. Aus den einzelnen Prozessschritten werden die Anforderungen spezifiziert. Basierend auf dieser Spezifikation wird ein passender Anwendungstyp und Technologie gewählt. Ein Entwurf der benötigten Ansichten erfolgt nach den Vorgaben der Plattform sowie den Anforderungen. Nach der Implementierung erfolgt eine Evaluation der Zielvorgaben.

**Abriss:** Kapitel 2 beschreibt die Grundlagen der Arbeit. In Kapitel 3 wird der Prozess und die Anforderungen analysiert. Das Kapitel 4 behandelt das Entwerfen der einzelnen Ansichten, bevor in Abschnitt 5 und 6 die Implementierung und Evaluation der Anwendung beschrieben wird. Zuletzt wird es einen Ausblick und ein Fazit über die gesamte Arbeit geben.

## 2. Grundlagen

Für ein besseres Verständnis des Gegenstand der vorliegenden Arbeit ist zunächst eine genaue Definition des Produktes notwendig. Damit ein Produktkonfigurator später verwendet werden kann, müssen die Vorgänge bei der Konfiguration verstanden werden. Weiterhin werden für eine spätere Entscheidung des korrekten Anwendungstypen die unterschiedlichen mobilen Anwendungsformen vorgestellt.

### 2.1. Definition Produkt

Im Marketing wird ein Produkt als Ergebnis im Produktionsprozess definiert. Innerhalb des Prozesses entsteht ein Produkt, welches am Ende eine Summe mehrerer materieller oder immaterieller Eigenschaften besitzt [Wi13]. Aus Sicht des Kunden ist ein solches Produkt ein Einzelstück, das für die Befriedigung eines Nutzens eingesetzt werden kann. Ein konkretes Produkt ist bspw. ein Auto, da es ein Resultat eines Produktionsprozesses ist. Ein Kunde nimmt das Produkt als einzelnes Objekt wahr. Bei der Produktion hingegen ist das Auto eine Zusammenstellung aus mehreren Einzelteilen. Hier besteht ein Auto aus den vier Hauptbereichen Karosserie, Motor, Innenausstattung und Getriebe. Die Innenausstattung besteht wiederum aus Sitzen und Armaturen. Diese Verfeinerung ist die Basis für die Individualität eines bestimmten Produktes. Je mehr Verfeinerungen existieren, desto komplexer ist ein einzelnes Produkt. Sobald der Hersteller mehr als eine Variante einer Einzelkomponente für den Kunden zur Verfügung stellt, lässt sich ein Produkt individualisieren. Für die Durchführung einer Individualisierung gibt es verschiedene Möglichkeiten. Die Einzelfertigung fertigt immer nur eine Einheit eines Produktes, wodurch jedes Produkt individuell ist. Das Gegenstück hierzu ist die Massenproduktion, bei der große Mengen des gleichen Produktes, mit unterschiedlichen Bauteilen hergestellt werden. Zwischen diesen beiden Extremen liegt die sogenannte mass-customization. Bei diesem Ansatz werden Produkte meist in

Bausteine unterteilt, die vom Kunden individuelle zusammengestellt werden können. Am Ende entsteht das Produkt.

Voraussetzung für die Individualisierung ist eine veränderte Wahrnehmung des Kunden. Ein Produkt darf nicht mehr als einzelnes Objekt gesehen werden. Für die individuellen Anpassungen muss der Kunde ein Produkt als eine Zusammenstellung mehrerer Komponenten verstehen. Diese veränderte Wahrnehmung muss dem Kunden vermittelt werden, um ihm dadurch eine Individualisierung seines Produktes zu ermöglichen.

Die zweite große Herausforderung entsteht bei baulichen Abhängigkeiten der einzelnen Produktteile. Bei einem komplexen Produkt mit vielen Einzelteilen können viele Abhängigkeiten entstehen. Wenn bei einem Auto bspw. ein bestimmter Motor ausgewählt wurde, so lassen sich nur für den Motor passende Getriebe einbauen. Durch die Verwendung mehrerer Möglichkeiten für eine bestimmte Einzelkomponente steigt ebenfalls die Anzahl der Abhängigkeiten. Die Prüfung dieser Abhängigkeiten muss ein Experte durchführen, der sich bestens mit der Produktzusammensetzung auskennt. Damit die einzelnen Vorgänge nicht zu komplex werden, sind geeignete Formen der Darstellung nötig.

### **2.1.1. Produktkatalog**

Um dem Kunden einen Einblick in ein Produkt zu verschaffen werden sogenannte Produktkataloge verwendet. Diese Kataloge sind meist in Papierform vorhanden und enthalten für den Kunden relevante Informationen über ein Produkt. Hierbei wird oben genanntes Ziel, beim Kunden eine andere Sicht des Produktes zu erzeugen, verfolgt. Für das Erreichen dieses Ziels bestehen Produktkataloge aus anschaulichen Bildern und besitzen eine übersichtliche Struktur für ein schnelles Finden des gewünschten Produktes. Die Herausforderung bei einem Katalog besteht bei der Abwägung, wie viele technische Informationen enthalten sein müssen, damit ein Produkt für den Kunden konfigurierbar wird. Je weniger der Kunde von der technischen Seite wissen muss, desto einfacher gestaltet sich der gesamte Konfigurationsprozess.

## 2.1.2. Boolesche Algebra in der Produktmodellierung

Die Zweite bereits genannte Herausforderung bei Produkten ist das Auswerten bzw. Modellieren der komplexen Abhängigkeiten von Einzelbauteilen. Ein Ansatz zur Lösung dieses Problems ist die boolesche Algebra. Bei der booleschen Algebra werden zwei Werte: wahr und falsch definiert [HLW06]. In der Aussagenlogik wird dies so verwendet, dass eine Aussage, wie "Heute regnet es" entweder wahr oder falsch sein kann [TT09]. Mithilfe von verschiedenen Operatoren lassen sich die Aussagen miteinander Verknüpfen, so dass auch komplexere Zusammenhänge möglich sind. Grundlegend zu nennen sind hier die Disjunktion (Symbol:  $\vee$ ), bei der einer von zwei Aussagen wahr sein muss, um den kompletten Ausdruck wahr werden zu lassen. Bei der Konjunktion (Symbol:  $\wedge$ ) müssen beide Aussagen zutreffend sein. Um Schlussfolgerungen durchführen zu können ist die sogenannte Wenn-Dann Verknüpfung (Symbol:  $\Rightarrow$ ) wichtig. Der Ausdruck "Wenn Aussage A, dann Aussage B" ist falsch, wenn Aussage A richtig und B falsch ist. In allen weiteren Konstellationen ist der gesamte Ausdruck wahr.

Übertragen auf das Modellieren eines Produktes können die Abhängigkeiten der Einzelbauteile mithilfe der booleschen Algebra aufgezeigt werden. Eine Auswertung dieser Modellierung erzeugt eine klare Aussage über die technische Umsetzung der aktuellen Auswahl. Hierbei können komplexe Zusammenhänge innerhalb eines Produktes korrekt abgebildet werden. Für das Auto Beispiel wäre eine Regel für die Beziehung von Motor und Getriebe in folgender Form möglich:

$$Verwendung\text{ von }MotorA \Rightarrow Einbau\text{ von }GetriebeA$$

Bedeutung: Wenn der Motor A verwendet werden soll, dann muss das Getriebe A eingebaut werden, damit die Regel zutrifft. Wenn der Motor A nicht eingebaut wird, ist für die Erfüllung dieser Regel der Einbau des Getriebes keine Voraussetzung.

Ein Problem, welches bei der Modellierung mit booleschen Regeln auftritt sind sogenannte Alternativen. Diese treten bei einer Zusammenstellung auf, bei der es mehrere Möglichkeiten gibt, wie eine Aussage wahr werden kann. Ein Beispiel wäre hier, dass die Auswahl des Armaturenbrett oder des Sitzes gefordert wird, wenn ein Motor und ein Getriebe ausgewählt wurde. Die Modellierung des Beispiels würde folgendermaßen aussehen:

$$MotorA \wedge GetriebeA \wedge (SitzA \vee ArmaturenbrettB)$$

Damit diese Bedingung wahr werden kann, müssen entweder Sitz A oder Armaturenbrett B ausgewählt werden. Eine weitere Möglichkeit in diesem Fall wäre die Auswahl beider Komponenten. Dieser konkrete Fall würde beim Einbau von Motor A und Getriebe A somit drei Alternativen bieten. Dieses Problem bei der Produktkonfiguration mit booleschen Regeln gilt es zu beachten sowie Möglichkeiten zu finden, wie diese ausgewertet werden können.

## 2.2. Produktkonfiguratoren

Nachdem die Probleme, welche bei der Konfiguration komplexer Produkte auftreten können, im vorherigen Abschnitt aufgezeigt wurden, bedarf es nun eines Lösungsansatzes zum Umgang mit diesen Schwierigkeiten. Eine Lösungsmöglichkeit bieten sogenannte Produktkonfiguratoren. Das Ziel des Konfigurators ist es, produkt spezifisches Wissen für die Anwender bereit zu stellen, welches zuvor von Experten in das System eingepflegt wurde. Dieses hilft bei der individuellen Zusammenstellung des Produktes. Ein solches System wird in die Kategorie der Expertensysteme[Pup88] oder wissensbasierte Systeme eingeordnet. Der Aufbau eines solches System ist in Abbildung 2.1 zu sehen.

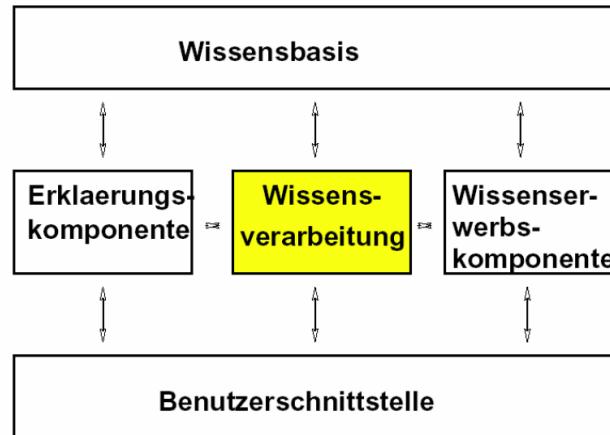


Abbildung 2.1.: Aufbau eines Expertensystems (Quelle: [Kel07, s.6])

Die zentrale Komponente ist die *Wissensverarbeitung*. Diese hat auf alle weiteren Komponenten Zugriff und interagiert mit diesen. Es werden die erhaltenen Fakten mithilfe

der vorhandenen Regeln verknüpft. Aus der Verknüpfung werden neue Fakten gewonnen, die auf der *Benutzerschnittstelle* angezeigt werden. Die Wissensbasis ist für das Speichern des Expertenwissens in Fakten und Regeln zuständig. Die Speicherung der Daten kann auf folgende zwei Arten geschehen[Rei13]:

- **generisch:** unabhängig vom aktuellen Anwendungsfall, meist in einfachen Wenn-Dann-Regeln oder auf einem Modell beruhend oder
- **fallspezifisch:** stellt Lösungen für einen konkreten Anwendungsfall bereit.

Die Pflege dieser Basis erfolgt durch die *Wissenserwerbskomponente*. Mit deren Hilfe lässt sich das vorhandene Expertenwissen in das System einpflegen. Die *Erklärungskomponente* unterstützt das Nachvollziehen des Ergebnisses durch Erläuterungen zu den getätigten Entscheidungen.

### 2.2.1. CAS Configurator Merlin

Eine konkrete Umsetzung eines Produktkonfigurators mit den entsprechenden Expertensystemkomponenten ist das Produkt CAS Configurator Merlin. Es ist die Konfigurationslösung der CAS Software AG für große und mittelständische Unternehmen. Das Produkt besteht aus Standardkomponenten, die auf die einzelnen Bedürfnisse der Großkunden angepasst werden. In Abbildung 2.2 ist der Aufbau und das Zusammenspiel der verschiedenen Komponenten des Konfigurators zu sehen:

Die Wissensverarbeitungs-Komponente aus Abschnitt 2.2 ist hier der sogenannte *Konfigurationskern*. Der Kern wertet die zuvor zusammengestellten Produktkomponenten aus. Für die Auswertung verwendet er sogenannte Regeldateien, die mit dem *Produktpflegeeditor* modelliert wurden. Diese Regeln sind auf booleschen Algebra aufgebaut um komplexe Abhängigkeiten der Einzelteile eines Produktes modellieren zu können. Die Speicherung dieser Dateien erfolgt in der *Datenbank*.

Der Konfigurationskern berechnet ebenfalls sogenannte Alternativen. Diese treten auf, sobald die derzeitige Selektion alleine, ohne Hinzunahme von weiteren Bauteilen, nicht umsetzbar ist. Der Konfigurator kann in diesem Fall neue Möglichkeiten (Alternativen) vorschlagen, damit die Konfiguration durchgeführt werden kann. Die Auswahl der Konfigurationselemente erfolgt im sogenannten *Konfigurator-Client*. Der Client ist mit der Benutzerschnittstelle im Expertensystem zu vergleichen.

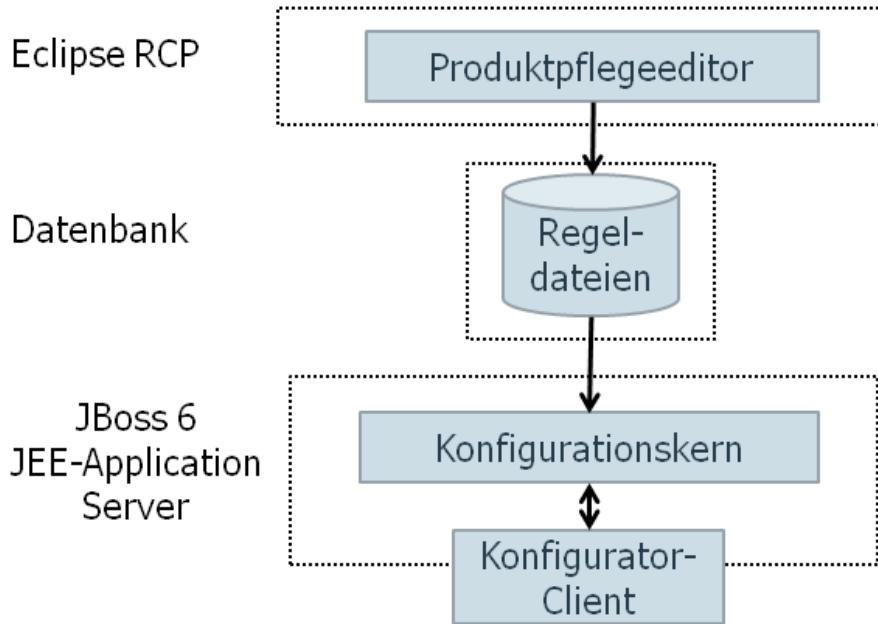


Abbildung 2.2.: Architektur und Datenfluss der Merlin Komponenten

Der Konfigurationskern sowie der Client befinden sich auf einem Java-Application-Server. Der Produktpflegeeditor ist eine eigenständige Rich-Client Anwendung, welche auf dem Eclipse Rich-Client-Plattform Framework [Ebe11] basiert.

## 2.2.2. Anwendungsbeispiel der Arbeit

Damit die Arbeit an einem geeigneten Beispiel durchgeführt werden kann, wird der vorhandene Produktkonfigurator eines Flugzeugherstellers verwendet. Hier soll anhand des vorhandenen Prozesses gezeigt werden, wie eine Vereinfachung des gesamten Workflows bei einer Produktkonfiguration mit der Unterstützung einer mobilen Anwendung aussehen kann.

Die aktuelle Konfigurationslösung wird für den Upgradeprozess eines vorhandenen Flugzeugs eingesetzt. Die Upgrades werden in die Kategorien Kabine und System eingegordnet. Alles was ein Flugzeugpassagier sehen kann, gehört in den Kabinenbereich. Alle weiteren Elemente sind Systemupgrades. Bei der Herstellung eines Flugzeugs wird derzeit hauptsächlich Einzelfertigung betrieben. Dies hat zur Folge, dass jedes Flugzeug sehr individuell ist. Aus diesem Grund ist die Konfiguration für ein Upgrade

sehr aufwändig und komplex. Damit ein Upgrade möglich ist, werden Status definiert, die ein Flugzeug erreichen kann. Auf Basis dieser Status werden die möglichen Upgrades definiert. Der Konfigurationsraum ist von der jeweiligen Maschine abhängig und liegt bei 40 - 50 Möglichkeiten im Systembereich. Im Kabinenbereich ist ein komplettes Upgrade der Innenausstattung möglich. Deshalb sind hier ähnliche Möglichkeiten, wie bei der initialen Erstellung des Flugzeuges vorhanden.

Dieses Anwendungsfeld ist besonders herausfordern, da somit zu der Auswahl der Produktkomponenten zusätzlich einzelne oder mehrere Flugzeuge ausgewählt werden müssen. Dies hat zur Folge, dass eine übersichtliche Darstellung der Upgrades alleine nicht ausreicht. Es muss ebenfalls eine Lösung für die Auswahl der einzelnen Flugzeuge gefunden werden. Dadurch, dass jedes Flugzeug individuell zusammengestellt wurde, muss jedes auch eigenständig überprüft werden.

## 2.3. Mobile Anwendungen

Die Frage, wie der Kunde besser an dem Entstehungsprozess eines Produktes teilhaben kann und dessen komplexen Aufbau verstehen kann, ist nur über ein ausreichendes Maß an Kommunikation und Wissensvermittlung zu bewerkstelligen. Eine Möglichkeit zur Unterstützung dieses Prozesses bieten mobile Anwendungen.

Diese sind definiert als eine Software, die auf einem Smartphone oder Tablet verwendet werden. Mit dieser Form der Anwendung können neue Anwendungsgebiete im Softwareinsatz erschlossen werden. Ein neues Einsatzgebiet ist der mobile Einsatz der App bei einem Kunden vor Ort [BDB08]. Hier können insbesondere Tablet-PCs die Kommunikation mit dem Kunden fördern [Hau13]. Die Vorteile durch den großen Bildschirm und die Möglichkeit wie bei einem Blatt Papier den Kunden ins Verkaufsgespräch mit einzubeziehen sind hier überzeugend. Gleichzeitig ermöglicht eine Bedienung der Geräte durch Touch-Eingaben eine bessere Interaktion mit der Software. Die Möglichkeiten beim Einsatz dieser Geräte im Geschäftsumfeld ist noch nicht ausgeschöpft und birgt auch weiterhin Potenziale [PAC11, Fazit]. In diesem Markt gilt es sich durch gute Anwendungskonzepte zu etablieren. Diese Aussichten werden in der vorliegenden Arbeit versucht zu nutzen, um eine innovative Lösung für den Businessbereich zu bieten.

Für Entwickler einer mobilen Anwendung ist der Umgang mit den begrenzten Ressourcen auf dem Endgerät wichtig. Aufgrund der geringen Speicherkapazität der mobilen

Geräte sowie die Vermeidung von hohem Synchronisierungsaufwand ist eine Ablegung der Daten auf einem leistungsstärkeren Gerät essentiell. Deshalb ist eine Anbindung an einen Server eine wichtige Grundvoraussetzung bei einer mobilen Anwendung, wenn viele Daten zu verarbeiten sind. Durch die immer bessere mobile Anbindung an das Internet wird die Verbindung mit einem entfernten System einfacher. Bei der Entwicklung einer mobilen App ist damit die Entscheidung über die Online und Offline Komponenten wichtig.

Ebenfalls von großer Bedeutung ist die Auswahl der richtigen Art der Anwendung. Hier stehen drei Möglichkeiten zur Verfügung, die aufgrund von unterschiedlichen Einschränkungen verschiedene Ziele verfolgen. Im Folgenden werden diese drei Arten vorgestellt und in Bezug zu ihrem Nutzen für die Arbeit bewertet.

### **2.3.1. Native Anwendungen**

Native Anwendungen sind auf die jeweilige Zielplattform beschränkt [Clo13]. Die Anwendung kann nur auf dem gewählten Betriebssystem ausgeführt werden. Die Programme für native Apps werden dafür mithilfe des vorgegebenen Frameworks entwickelt. Die Verbreitung dieser Anwendungen auf die Endgeräte der einzelnen Benutzer erfolgt über spezielles Software-Stores. Diese Stores werden von den Herstellern der jeweiligen Betriebssysteme bereitgestellt. Die zentrale Verwaltung der Apps sorgt für ein schnelles Verbreiten der Anwendung.

Die Vorteile einer nativen Anwendung liegen im Bereich der Performance und der Funktionen. Da die Programmierung der gegebenen Hardware angepasst wird, sind native Anwendungen für das Zielsystem optimiert. Durch die Optimierung können die Hardwareressourcen besser ausgenutzt werden. Die zweite Stärke von diesem Anwendungstyp ist der erweiterte Funktionsumfang [Sma13]. Es können lokale Datenbanken, 3D-Renderer, lokale Sensoren und Ressourcenmanager der Plattform ohne Anpassungen verwendet werden. Dies erhöht die Anzahl der Möglichkeiten, die mit einer nativen Anwendung realisiert werden können. Bei der Verwendung der App in einem Offline Modus(ohne Internetverbindung), kann die native Anwendung ohne Einschränkungen arbeiten. Es ist keine Verbindung mit einem Server notwendig, um Funktionen bereitzustellen. Durch den direkten Speicherzugriff ist eine Speicherung von größeren Onlinedaten ohne größere Probleme möglich.

Der Nachteil dieses Anwendungstyp ist, dass die entwickelten Komponenten nicht für andere Plattformen eingesetzt werden können. Bei einer Implementierung für ein anderes Betriebssystem muss die Anwendung neu entwickelt werden. Dies führt zu einem sehr hohen Aufwand bei der Entwicklung. Durch die Vorgabe der Verteilung über die vorhandenen Stores ist man gezwungen die Anwendungen auf diesen bereitzustellen. Hier fallen Kosten für die Qualitätssicherungsmaßnahmen sowie Gebühren für die Bereitstellung an.

### **2.3.2. Web Anwendungen**

Web Anwendungen sind mobile Webseiten, welche mit zusätzlichen Funktionen, die ein mobiles Gerät zur Verfügung stellt, erweitert sind. Sie agieren gleich einer App. Der Start der Anwendung erfolgt durch das Aufrufen einer Url im Browser . Es werden dabei die Webtechnologien HTML, CSS und Javascript verwendet. Für die Entwicklung solcher Anwendungen stehen verschiedene Frameworks zur Verfügung. Die Funktionen der web Anwendung werden vom Browser bereitgestellt und hängen von dessen Funktionsumfang ab (Unterstützte HTML-5 Funktionen beliebter Browser: Google Chrome ca. 100, Firefox 93 und Safari 90 Stand: 4.4.2013, Quelle: [Sch13]).

Der Hauptvorteil dieses Anwendungstypus liegt bei der plattformübergreifenden Verwendung dieser Apps. Es muss kein bestimmtes Betriebssystem festgelegt werden, für das die Anwendung entwickelt werden soll. Die einzige Voraussetzung zur Verwendung ist ein funktionierender Webbrowser, der bei jedem mobilen Gerät der Standard ist. Der Updateprozess, das Aktualisieren der App, ist ebenfalls sehr einfach, da ein zentraler Server für die Darstellung zuständig ist. Es fallen keine Lizenzgebühren bei der Bereitstellung an, da man den lokalen Store nicht benötigt.

Der Nachteil liegt am beschränkten Funktionsumfang im Vergleich zu einer nativen App. Da die Webapp für eine möglichst große Zahl an mobilen Geräten verfügbar ist, können nicht alle Funktionen unterstützt werden. Gleichzeitig wird eine weitere Einschränkung durch den Browser hinzugefügt. Von diesem hängt ab, wie viel die Anwendung auf dem Betriebssystem ausführen darf. Dateioperationen beispielsweise sind aus Sicherheitsgründen aus dem Browser heraus nur begrenzt möglich. Die Offline Nutzung der Anwendung ist nur bedingt gegeben. Es kann zwar offline gestartet werden, jedoch stößt man bei einer großen Menge an Daten an Grenzen, da Speicherbeschränkungen für den Browser bestehen.

### **2.3.3. Hybride Anwendung**

Das Problem der Bereitstellung von mobilen Anwendungen für immer neuere Technologien und Plattformen wird durch den Ansatz einer hybriden App gelöst. Bei dieser Form der Anwendung werden die Vorteile einer Native und einer Web App vereint. Der Kern dieses Anwendungstyps ist eine Web Anwendung. Diese wird in einen nativen Container eingebunden. Der Container ist eine native Anwendung, die den Funktionsumfang der Zielplattform verfügbar macht. Hierbei werden Adapter auf dem Betriebssystem bereitgestellt, die von der Web Anwendung verwendet werden können. Mithilfe des Containers lassen sich mehr native Funktionen verwenden. Dies bietet sich bspw. bei besonders rechenintensiven Operationen an.

Die Stärken der hybriden Anwendung liegen in dem erweiterten Funktionsumfang im Gegensatz zu einer Web Anwendung. Durch das Container-Prinzip kann jede Funktion, die bei einer nativen Anwendung verfügbar ist auch verwendet werden. Ebenfalls erfolgt die Verteilung der Anwendung über die jeweiligen Stores, was bei einem schnellen finden der Anwendung hilft. Die Vorteile der hohen Wiederverwendbarkeit der entwickelten Komponenten kommen ebenfalls hinzu. Es ist somit einfacher für mehrere Plattformen eine App zur Verfügung zu stellen.

Für die Verwendung auf mehreren Plattformen müssen im Gegensatz zu einer Web Anwendung für jedes Betriebssystem ein neuer Container bereitgestellt werden. Dies erfordert zusätzlichen Aufwand bei der Entwicklung. Hybride Anwendungen kommen durch die zusätzliche Verwendung der Web Technologie nicht an die Performance und Effizienz einer nativen Anwendung heran.

# 3. Analyse der Anwendung

Damit der Konfigurationsprozess für den Kunden einfacher wird, muss im ersten Schritt der aktuelle Prozess analysiert werden. Darauf aufbauend werden Konzepte zur Vereinfachung dieses Prozesses überlegt und ein neuer Workflow erstellt. Die Anforderungen an die neue Anwendung können mithilfe des neuen Prozesses anschließend spezifiziert werden.

## 3.1. Aktueller Konfigurationsprozess

Die Anpassungen und Vereinfachungen eines Konfigurationsprozesses bei der Verwendung von mobilen Anwendungen sollen am konkreten Anwendungsbeispiel gezeigt werden. Die Analyse beginnt bei grundlegenden Fragen, wie ein solcher Prozess aufgebaut ist und wird im Folgenden anhand eines konkreten Kundenfalls weiter ausgebaut.

Ein Auszug aus einem Konfigurationsprozess ist in Abbildung 3.1 zu sehen. Die Darstellung zeigt eine Vereinfachung, da in der Praxis auch Schleifen entstehen können. Zu Beginn möchte der Kunde ein konkretes Produkt auswählen. Dieses wird aus dem Katalog herausgesucht. Anschließend kann er mithilfe von eindeutigen Produktnummern die Bestellung über ein Formular weitergeben. Damit die Zusammenstellung des

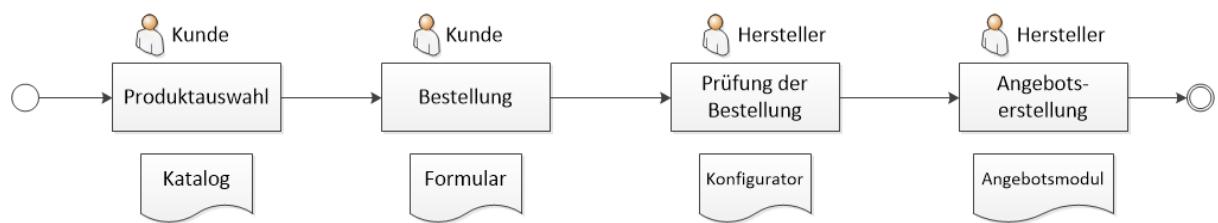


Abbildung 3.1.: Auszug eines Konfigurationsprozesses

Kunden geprüft werden kann, werden die Daten beim Hersteller in das Konfigurationssystem eingepflegt. Dieses berechnet die komplexen Abhängigkeiten und prüft die Gültigkeit der Konfiguration. Der letzte Schritt ist die Angebotserstellung. Hierbei wird das konkrete Angebot für den Kunden erstellt.

In der Umsetzung des beschriebenen Prozesses entsteht ein Kommunikationsproblem. Dies tritt aufgrund der Trennung von Produktauswahl und der Prüfung der Bestellung durch den Produktkonfigurator auf. Der Kunde ist nur bei der Auswahl beteiligt. Das Feedback für die Umsetzung der Zusammenstellung erfolgt erst nach der Weitergabe der Bestellung. Probleme zeigen sich, wenn bei der Konfiguration Alternativen auftreten, die vom Kunden zu entscheiden sind, oder eine Konfiguration komplett nicht umsetzbar ist. Hier muss ggf. der Prozess wiederholt werden. Für die Lösung ist eine Hinzunahme des Kunden bei der Konfigurationsüberprüfung notwendig. Das Ziel, welches durch die Erstellung einer mobilen Anwendung erreicht werden soll, ist ein schnelles Feedback bei der aktuellen Auswahl.

Die zweite Stelle, an der eine Verbesserung möglich ist, befindet sich bei der Eingabe der Daten für den Produktkonfigurator und der damit verbundenen Prüfung. Der Kunde hat im vorigen Schritt bereits die Daten für eine Konfiguration erfasst. Eine zweite Erfassung kann zu Fehlern oder Kommunikationsproblemen führen. Diese Probleme entstehen bei der falschen Eingabe des bestellten Produktes. Hierdurch kann der Konfigurator ein falsches Ergebnis liefern, welches zu einem inkorrekten Angebot führt. Auch in einem solchen Fall muss der Prozess wiederholt werden. Für die Verbesserung dieser Situation wird die Auswahl des Kunden direkt an den Konfigurator übermittelt. Diese Maßnahme minimiert die Fehler bei der Erfassung.

Ein weiteres Problem ist die Verwendung von Produktnummern. Diese Nummern dienen einer eindeutigen Identifizierung des Produktes bei der Bestellung. In diesem Schritt kann es passieren, dass der Kunde die falsche Produktnummer auswählt, weshalb nicht das richtige Produkt bei der Angebotserstellung verwendet wird. Bei der Produktauswahl wird dieses Problem umgangen, da die Informationen nicht notwendig sind, der Kunde interessiert sich hierbei nur für das konkrete Produkt. Für die Vereinfachung des Prozesses und eine Fehlerminimierung ist eine automatische Zuordnung der korrekten Nummer im Hintergrund die Lösung. Somit wird der direkte Umgang des Kunden mit einzelnen Produktnummern vermieden und die alleinige Auswahl des Produktes steht für den Kunden im Vordergrund. Hierdurch wird der Kunde nicht mit zu vielen Produktdetails überfordert und es entsteht eine Vereinfachung des Prozesses.

Zusammenfassend sind drei grundlegende Probleme beim dargestellten Konfigurationsprozess festgestellt:

- zusätzlicher Kommunikationsaufwand durch zu spätes Feedback,
- Daten werden doppelt erfasst,
- Produktnummern sind für den Kunden schwierig zu handhaben.

Diese drei Punkte müssen im neuen Workflow verbessert werden.

### **3.1.1. Übertragung auf die Anwendungsdomäne Luftfahrt**

Beim Anwendungsbeispiel in der Luftfahrt erfolgt die Auswahl der Upgrades ebenfalls über einen Produktkatalog. In diesem Katalog sind die einzelnen Upgrades aufgeführt. Die Auswahl der Produkte erfolgt über eine vorhandene Weboberfläche. Diese Oberfläche ist über die Homepage des Kunden verfügbar. Der Endkunde, im Anwendungsbeispiel eine Fluggesellschaft, wählt das gewünschte Upgrade aus dem Katalog aus. Bei der Bestellung werden die Produktcodes der Auswahl verwendet. Zusätzlich müssen die Flugzeuge angegeben werden, die das Upgrade erhalten sollen. Die Identifizierung erfolgt anhand von eindeutigen Flugzeugcodes. Beide Nummern werden im nächsten Schritt von einem Produktkonfigurator erfasst, der eine Überprüfung der Konfiguration durchführt.

Beim Kunden wird die klare Trennung der Auswahl und der Überprüfung durch die Verwendung von zwei unterschiedlichen Systemen deutlich. Die Kommunikation der beiden Systeme erfolgt über die Produkt-, bzw. Flugzeugcodes. Bei der Überprüfung der Konfiguration ist der Kunde nicht involviert. Der Experte bearbeitet die Bestellung und pflegt diese in das System ein. Die Ergebnisse werden anschließend für den Kunden aufbereitet, so dass dieser die auftretenden Alternativen verstehen und entsprechend handeln kann. Diese Besonderheit im Anwendungsbeispiel muss bei der Umstellung des Prozesses beachtet werden.

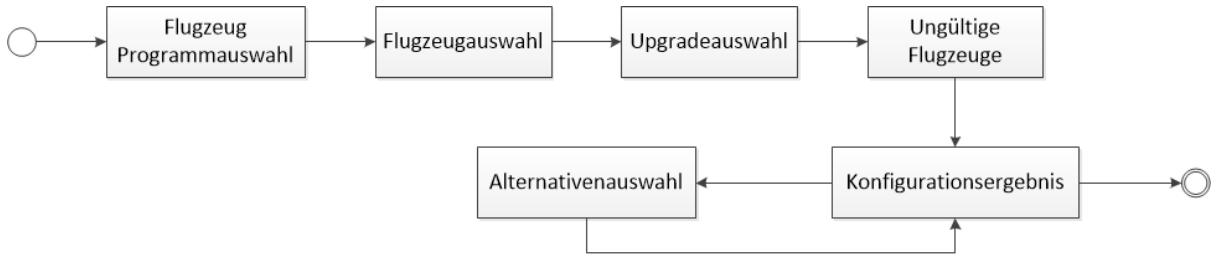


Abbildung 3.2.: Programmablauf des Anwendungsbeispiels

Abbildung 3.2 zeigt den Ablauf einer Konfiguration mit dem aktuellen System. Im ersten Schritt wird das passende Flugzeugprogramm ausgewählt. Ein Programm ist eine grobe Einteilung für Flugzeuge nach deren Größe und Art. Die Auswahl ist eine erste Filterung der Datensätze. Des Weiteren werden mit Hilfe des Programms die Regeln ausgewählt, die auf dem Konfigurationsserver für eine Konfiguration verwendet werden. Anschließend folgt die Auswahl der Flugzeuge, die ein Upgrade erhalten sollen. Bei der Identifizierung für eine Bestellung wird die Flugzeugnummer angegeben. Sind die Flugzeuge ausgewählt, werden Upgrades aus einer Liste selektiert. Die Auswahl erfolgt mit der eindeutigen Nummer, welche bei der Bestellung angegeben wird.

Nach dem vorigen Schritt sind alle für die Konfiguration benötigten Elemente ausgewählt. Es folgt eine Validierung der Flugzeuge. Bei dieser Überprüfung werden die einzelnen Flugzeuge auf Konfigurationen untersucht, die in einem Widerspruch mit dem ausgewählten Upgrade stehen. Wenn keine Widersprüche vorhanden sind, werden sogenannte Konfigurationsgruppen gebildet. Eine Konfigurationsgruppe enthält Flugzeuge, die in die gleichen Zielzustände kommen, wenn das Upgrade eingebaut wird. Wenn es mehrere Möglichkeiten gibt, um in einen bestimmten Zustand des Flugzeuges zu kommen, sind sogenannte Alternativen in einer Konfigurationsgruppe enthalten. Damit die Konfiguration vollständig ist, muss der Anwender für jede Gruppe eine Alternative auswählen.

Nachdem eine vollständige Konfiguration erzeugt wurde, wird daraus ein Excel-Dokument generiert. In diesem sind die Upgrades enthalten, die in den einzelnen Flugzeugen eingebaut werden müssen. Aus dem Dokument wird ein Upgrade-Angebot erstellt, das anschließend dem Kunden vorgelegt wird.

Das Hauptproblem des aktuellen Systems ist es, dass nur Experten die Anwendung bedienen können. Eine effektive Nutzung kann nur mithilfe der Produktcodes erfolgen. Dies führt dazu, dass man ein großes Wissen über die Produktstruktur besitzen muss.

Dadurch kann der Kunde die Konfiguration mit der aktuellen Anwendung nicht selbstständig durchführen. Dieses Problem wird im Folgenden bei der Modellierung des neuen Workflows gelöst.

## **3.2. Workflow Modellierung**

Der neue Workflow muss die im vorigen Abschnitt erwähnten Probleme des Konfigurationsprozesses lösen. Anschließend müssen die Lösungen auf den konkreten Anwendungsfall übertragen werden und ein neuer Programmablauf gefunden werden.

### **3.2.1. Umstellung auf einen mobilen Konfigurationsprozess**

Die Probleme mit der Verwendung von Produktnummern und die doppelte Erfassung der Daten lassen sich durch die Zusammenlegung von Auswahl und Konfigurationsprüfung mit einem System lösen. Diese Maßnahme ermöglicht es dem Kunden die gewünschte Auswahl zu tätigen und gleichzeitig ein Feedback der Konfiguration zu erhalten. Durch die bessere Rückmeldung verringert sich der Kommunikationsaufwand, da der Kunde im Idealfall sofort das Resultat sieht.

Für eine noch bessere Unterstützung sowie Hilfestellung beim Aufkommen von Alternativen ist die Durchführung des Prozesses mit einem Mitarbeiter des Herstellers von Vorteil. Dieser kann den Kunden durch die Konfiguration führen oder dabei unterstützen. Gleichzeitig wird hier ein besseres Verständnis für das Produkt ermöglicht. Der Hersteller hat den Vorteil einer Beschleunigung des Prozesses von der Produktauswahl bis zur Angebotserstellung. Er kann durch die direkte Auswahl der Produkte sowie ein sofortiges Prüfen und ggf. eine Selektion der Alternativen die Konfiguration abschließen und ein Angebot erstellen.

Diese Umstellungen des Prozesses setzt die Verwendung eines mobilen Endgerätes, in diesem Fall eines Tablet-PCs voraus. Durch die Mobilität des Gerätes kann die Konfiguration direkt beim Kunden vor Ort durchgeführt werden. Die verbesserte Kommunikationsmöglichkeit mithilfe des Tablets sorgt für ein besseres Verständnis des Kunden und dessen Wünsche.

Zusammenfassend sind folgende zwei Maßnahmen beim neuen Workflow durchzuführen:

- Zusammenlegen von Produktauswahl und Produktkonfiguration.
- Unterstützung durch einen Mitarbeiter des Herstellers.

Die konkrete Umsetzung dieser beiden Änderungen wird im Folgenden am Anwendungsbeispiel durchgeführt.

### **3.2.2. Soll-Prozess des Workflows im Anwendungsbeispiel**

Für die Umsetzung der aus Abschnitt 3.2.1 erstellten Maßnahmen müssen die beiden vorhandenen Systeme für die Auswahl und Konfiguration auf ein gemeinsames Gerät portiert werden. Das Konfigurationssystem muss hierfür vereinfacht werden, um den Zugang für den Kunden zu erleichtern. Die Herausforderung besteht hier in einer übersichtlichen Darstellung der Konfigurationsergebnisse. Diese müssen für den Kunden nachvollziehbar aufbereitet werden, weil er nicht die Produktkenntnisse des Experten besitzt. Da der Kunde die Auswahl der einzelnen Produkte "live" vornimmt, muss die Anwendung ein schnelleres Feedback erzeugen. Bei der vorherigen Lösung hat der Experte die komplette Zusammenstellung des Kunden erhalten und musste diese in das System übertragen. Beim neuen Workflow dagegen möchte der Kunde die Reihenfolge bei der Auswahl selbst bestimmen. Dies muss im neuen Anwendungsverlauf berücksichtigt werden.

In Abbildung 3.3 ist der Anwendungsverlauf der App zu sehen, welcher die benannten Problemstellen im Ist-Prozess löst. Analog zu der Weboberfläche wird bei einer neuen Konfiguration zuerst ein Programm (**Programm-auswahl**) ausgewählt. Dies ist aufgrund einer Filterung der Daten weiterhin notwendig. Nach der Auswahl gelangt der Benutzer in eine **Zusammenfassung** der aktuellen Konfiguration. Von dieser Ansicht aus können Flugzeuge (**Flugzeugauswahl**) oder Upgrades (**Upgradeauswahl**) selektiert werden. Hier wird den unterschiedlichen Bedürfnissen des Kunden entsprochen und kein strikter Konfigurationsablauf vorgegeben, wie es im vorherigen System war. Mit dieser Umstellung wird dem Kunden die Möglichkeit gegeben die einzelnen Upgrades nacheinander zu wählen und jederzeit eine schnelle Änderung zu ermöglichen. Sobald mindestens ein Flugzeug oder ein Upgrade ausgewählt ist, wird die Konfiguration überprüft. Nach der Überprüfung werden die Konfigurationsgruppen, die der Konfigurator erstellt hat, angezeigt. Diese Gruppen kann der Benutzer einsehen

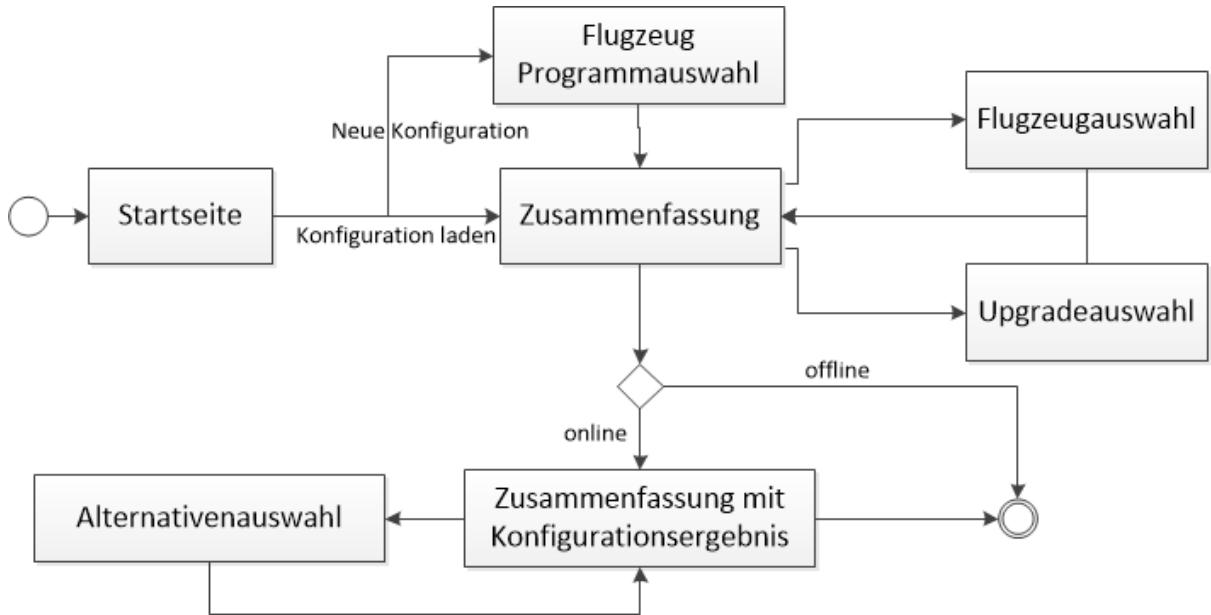


Abbildung 3.3.: Workflow der Konfigurator-App

und bei mehreren Alternativen in einer separaten Ansicht (**Alternativenauswahl**) die richtige Lösung auswählen. Ist die Konfiguration vollständig, so hat der Nutzer die Möglichkeit, die aktuelle Zusammenstellung zu bestellen und den Vorgang mit der Bestellung zu beenden.

Da die Verwendung des Konfigurationsservers, wie in Kapitel 2.2.1 beschrieben nach dem Client-Server Modell aufgebaut ist, wird eine andere Möglichkeit bei der Konfiguration nötig. Beim mobilen Einsatz der Anwendung kann es passieren, dass eine Verbindung mit dem Konfigurationsserver nicht möglich ist. Aus diesem Grund muss es einen alternativen Weg des Workflows geben. Wenn der Konfigurationsserver nicht verfügbar ist, wird die Konfiguration gespeichert, damit eine Prüfung später durchgeführt werden kann. In diesem Fall ist der Prozesses mit der Speicherung der Konfiguration beendet.

### 3.2.3. Prozessbeteiligte der Anwendung

Nach der Umstellung des Konfigurationsprozesses müssen die Benutzer der Anwendung identifiziert werden. Dies ist eine Voraussetzung, um die Anforderungen für die Zielgruppe passend zu definieren. Der neue Prozess ist darauf ausgelegt, dass ein Vertreter des Flugzeugherstellers zusammen mit einem Endkunden die Konfiguration durchführt. Der Kunde im Anwendungsbeispiel ist kein unerfahrener Laie. Stattdessen

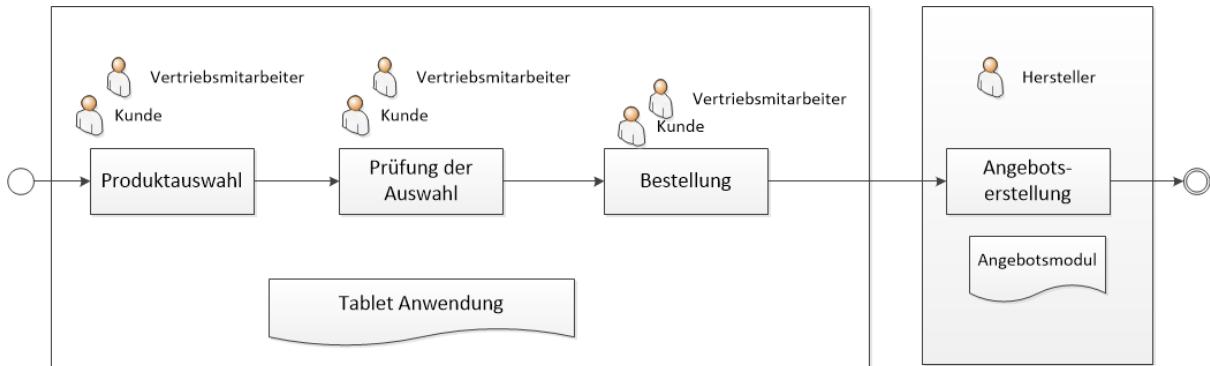


Abbildung 3.4.: Anwendungsbereiche der Tablet Anwendung

ist er für Upgrades der vorhandenen Flugzeugflotte einer Fluggesellschaft zuständig. Somit ist bereits ein Wissen über das Produkt vorhanden. Dies hat zur Folge, dass der Kunde die Struktur und den Aufbau des Flugzeugs kennt. Dadurch unterscheidet sich die Konfiguration bspw. zu einem Online-Konfigurator eines Autoherstellers. Hier sind unerfahrene Anwender die Zielgruppe, denen der Aufbau des Produktes auf eine andere Weise erklärt werden müsste.

Betrachtet man die Zielgruppe im Workflow des Anwendungsbeispiels, so stellt man fest, dass zuvor die Produktauswahl sowie die Bestellung dem Kunden überlassen wurde. Die neue Tablet Anwendung vereint die einzelnen Prozessschritte, so dass am Ende eine direkte Bestellung folgen kann, die zu einem Angebot führt. Die Durchführung des Konfigurationsprozesses wird im Zusammenwirken von Kunde und Anbieter durchgeführt (siehe Abbildung 3.4). Dies hat zur Folge, dass die App für zwei unterschiedliche Zielgruppen konzipiert werden muss.

### 3.3. Anforderungsanalyse

Nachdem der grundlegende Prozess auf die Bedürfnisse des Kunden sowie an die mobile Umgebung angepasst ist, können daraus die Anforderungen der Anwendung für das Anwendungsbeispiel festgelegt werden. Bei der Festlegung wird zuerst die allgemeine Anforderung erläutert sowie anschließend auf das Anwendungsbeispiel spezifiziert. Zur übersichtlichen Darstellung wird im Folgenden zwischen funktionalen und nicht-funktionalen Anforderungen unterschieden.

### 3.3.1. Nicht-Funktionale Anforderungen

Die nicht-funktionalen Anforderungen betreffen alle Maßnahmen zur Vereinfachung, bzw. Unterstützung des Prozesses sowie Anforderungen aufgrund des vorhandenen Systems. Diese speziellen Voraussetzungen sind für den Benutzer meistens nicht sichtbar und laufen im Hintergrund, bzw. unbewusst ab. Aus diesem Grund müssen hier auch spezielle Gütekriterien festgelegt werden, um am Ende eine Evaluation dieser nicht-funktionalen Anforderungen zu ermöglichen. Als Basis werden die 10 Heuristiken für Benutzerschnittstellen von Nielsen [Nie95] verwendet. Analog lassen sich aus dem neuen Workflow folgende Anforderungen spezifizieren:

**Einfache Bedienung:** Der Kunde ist kein Experte und verwendet die Anwendung nicht jeden Tag somit ist eine einfache Bedienung eine Voraussetzung. Je weniger bei der Verwendung der Software erklärt werden muss, desto besser ist diese Anforderung erfüllt. Für die Erfüllung dieses Ziels sollen folgende Schwerpunkte der Softwareergonomie [Rud06] beachtet werden:

- Selbstbeschreibungsfähigkeit,
- Lernförderlichkeit,
- Erwartungskonformität.

Die Kriterien nach Nielsen sind Konsistenz, Erkennung vor Erinnerung und Sichtbarkeit des Systemstatus.

**Schnelle Bedienung:** Die zweite Zielgruppe der Anwendung ist der Experte. Für diesen muss es die Möglichkeit einer schnellen Bedienung der Anwendung geben, sodass er die Konfiguration möglichst effizient gestalten kann. Somit wird eine flexible Bedienung der Anwendung nötig. Für eine zusätzliche Steigerung der Effizienz muss die Navigation zwischen den einzelnen Seiten schnell sein. Dies bedeutet, dass beim Übergang von einer zur nächsten Seite der Benutzer nicht lange warten sollte. Ebenfalls muss es genügend Feedback geben, wenn ein Seitenwechsel länger dauern sollte. Hier werden die beiden Heuristiken Sichtbarkeit des aktuellen Status und Flexibilität sowie Effizienz bei der Benutzung beachtet.

Bez.	Anforderung	Beschreibung	Heuristiken nach Nielson
N1	Einfache Bedienung	Die Anwendung soll von unerfahrenen Benutzern bedient werden können.	<ul style="list-style-type: none"> <li>• Sichtbarkeit des aktuellen Status</li> <li>• Erkennung vor Erinnerung</li> <li>• Sichtbarkeit des aktuellen Status</li> </ul>
N2	Schnelle Bedienung	Experten müssen die Anwendung schnell und effizient bedienen können.	<ul style="list-style-type: none"> <li>• Flexibilität und Effizienz bei Benutzung</li> <li>• Sichtbarkeit des aktuellen Status</li> </ul>
N3	Optimierung auf Umgebung	Die Anwendung muss auf die Zielplattform optimiert werden.	<ul style="list-style-type: none"> <li>• Ästhetisches und minimalistisches Design</li> <li>• Standards und Konsistenz</li> </ul>

Tabelle 3.1.: Nicht-Funktionale Anforderungen mit Bezeichner

**Optimierung auf die Umgebung:** Da ein mobiles Gerät nur begrenzte oder eingeschränkte Ressourcen zur Verfügung hat, müssen die einzelnen Bedienelemente auf die Umgebung angepasst sein. Ebenfalls muss die Bedienung für eine Touch-Eingabe optimiert sein. Für ein einheitliches Aussehen der Anwendung müssen die Richtlinien der jeweiligen Technologie beachtet werden. Diese Anforderung enthält die beiden Heuristiken ästhetisches und minimalistisches Design sowie Standards und Konsistenz.

Für eine bessere Übersicht der nicht-funktionalen Anforderungen sind diese in der Tabelle 3.1 dargestellt. Zu jeder Anforderung wird hier ein zusätzlicher Bezeichner definiert.

### 3.3.2. Funktionale Anforderungen

Die funktionalen Anforderungen sind von der Prozessbeschreibung im Abschnitt 3.2.1 abhängig. Im Folgenden werden diese speziell auf das Anwendungsprojekt spezifiziert. Zusätzlich zu der Spezifikation wird der Bezug zum neuen Workflow hergestellt:

**Filterung der Anwendungsdaten:** Da sehr viele Daten vorhanden sind, müssen diese für eine übersichtliche Darstellung im Vorfeld gefiltert werden. Die Filterung erfolgt in zwei Schritten:

1. Auswahl der Kundendaten: Die Anwendung soll speziell bei einem Kunden eingesetzt werden. Aus diesem Grund kommen nur die kundenspezifischen Daten zum Einsatz.
2. Programmauswahl: Durch die Wahl des Programmes wird eine Filterung der einzelnen Flugzeuge und der möglichen Updates erreicht.

Diese Anforderung wird ebenfalls durch die Verwendung einer mobilen Umgebung notwendig. Damit der zweite modellierte Workflow mit einer nicht vorhandenen Internetverbindung funktioniert, müssen die Daten offline verfügbar sein. Da nur begrenzte Speichermöglichkeiten auf dem Gerät vorhanden sind, können nicht alle Kundendaten gespeichert werden. Dies bedingt eine Filterung durch die Auswahl des Kunden.

**Upgradeauswahl:** Bei der Auswahl von Upgrades im Anwendungsbeispiel werden die Funktionen des Produktkataloges benötigt. Es muss möglich sein, ein bestimmtes Produkt an- oder abzuwählen. Damit der Kunde den Aufbau des Produktes versteht, müssen die einzelnen Produkte nach einer bestimmten Struktur geordnet sein.

**Flugzeugauswahl:** Flugzeuge, die ein ausgewähltes Upgrade erhalten sollen, müssen ebenfalls in der Anwendung auswählbar sein. Die Anforderungen an diese Auswahl sind nicht die Gleichen wie bei der Produktauswahl. Der Schwerpunkt muss hier auf ein schnelles Finden der einzelnen Flugzeuge gelegt werden. Dies wird ebenfalls durch eine dem Kunden bekannte Struktur erreicht.

**Anzeige der Konfigurationsergebnisse:** Damit der Kunde ein schnelles Feedback der aktuellen Auswahl erhält, müssen die Konfigurationsergebnisse angezeigt werden. Dies impliziert eine Anbindung an den Konfigurationsserver, der die entsprechenden Ergebnisse berechnet. Diese Anzeige muss eine vereinfachte Darstellung beinhalten, damit der Kunde diese Ergebnisse verstehen kann.

**Alternativenauswahl:** Damit eine vollständige Konfiguration erzeugt werden kann, muss die Anwendung eine Auswahl für Alternativen bereitstellen. Hierbei muss das Verständnis des Kunden für das Produkt berücksichtigt werden. Die Alternativen müssen auf eine verständliche Art dargestellt sein.

**Speichern und Laden der Konfiguration:** Die zweite Voraussetzung des alternativen Offline Workflows ist eine Speicherung der Konfiguration. Für eine spätere Bearbeitung ist das erneute Laden der Konfiguration notwendig. Diese beide Anforderungen werden aufgrund der mobilen Zielumgebung benötigt.

Die Zuordnung der einzelnen Anforderungen zu den jeweiligen Prozessschritten ist in Tabelle 3.2 zu sehen. Hierbei werden die drei Prozessanforderungen Produktkatalog, Konfiguration und mobile Zielumgebung unterschieden.

NR.	Anforderung	Beschreibung	Prozesszuordnung
F1	Upgrade-auswahl	Es sollen Upgrades für Flugzeuge auswählbar sein	Produktkatalog
F2	Flugzeug-auswahl	Es müssen Flugzeuge eines bestimmten Kunden auswählbar sein.	Produktkatalog
F3	Konfigurationsergebnisse anzeigen	Übersichtliche Darstellung der aktuellen Zusammenstellung	Konfiguration
F4	Alternativenauswahl	Bei mehreren Möglichkeiten einer Auswahl sollen Alternativen ausgewählt werden können	Konfiguration
F5	Speichern und Laden	Die getätigte Auswahl soll gespeichert und geladen werden können	mobile Zielumgebung
F6	Filterung der Anwendungsdaten	Die Daten (Flugzeuge und Upgrades) müssen gefiltert werden können	mobile Zielumgebung

Tabelle 3.2.: Funktionale Anforderungen mit Bezeichner

# **4. Entwurf der Benutzerschnittstelle**

Die Anforderungen der Anwendung für die Unterstützung des neuen Workflows wurden im vorigen Kapitel spezifiziert. Diese Kriterien werden im nächsten Schritt in einem Entwurf der Benutzerschnittstelle umgesetzt. Bevor die Schnittstelle entworfen werden kann, muss die geeignete Anwendungsform und eine darauf basierende Technologie ausgewählt werden. Anschließend folgt eine kurze Analyse der Zielplattform und deren Konzepte, damit diese beim Entwurf berücksichtigt werden können.

## **4.1. Auswahl der Anwendungsplattform**

Aufgrund der vielen Möglichkeiten bei der Entwicklung von mobilen Anwendungen muss die Anwendungsart und Technologie richtig gewählt werden. Hierbei soll zuerst die passende Anwendungsform (Nativ, Web oder Hybrid siehe 2.3) gewählt werden. Durch diese Vorauswahl, wird die Anzahl der möglichen Technologien begrenzt, was im folgenden Schritt die Auswahl vereinfacht.

### **4.1.1. Anwendungsform**

Damit die richtige Anwendungsform der App ausgewählt werden kann, müssen die drei Möglichkeiten Nativ, Web und Hybrid auf ihre Eignung bei der Umsetzung der Anforderungen untersucht werden. Für eine Entscheidungsgrundlage werden konkrete Kriterien festgelegt. Die Umsetzung der rein fachlichen Funktionen Produktauswahl und Konfiguration kann mit jeder Anwendungsform durchgeführt werden. Hier sind die Unterschiede für eine Auswahl nicht ausreichend.

Bei den funktionalen Anforderungen sind die zwei Kriterien, die aufgrund der mobilen Zielumgebung wichtig sind für die Auswahl bedeutend. Diese sind das Speichern und Laden (F5), sowie die Filterung der Anwendungsdaten (F6). Für die Umsetzung dieser

Funktionen ist eine Verwendung des Dateisystems auf der mobilen Zielumgebung notwendig. Hier muss eine Form der Speicherung, ob Datenbank oder einfaches Speichern in einer Datei möglich sein. Diese Voraussetzung ist für die Entscheidung der Anwendungsform essentiell.

Das zweite Kriterium leitet sich von den nicht-funktionalen Anforderungen einer schnellen Bedienung (N2) ab. Damit die Anwendung schnell bedient werden kann, müssen die vorhandenen Hardwareressourcen optimal verwendet werden können. Dies ist notwendig, um bspw. das Laden von Bildern zu beschleunigen. Ebenfalls müssen Seitenübergänge ohne große Wartezeiten möglich sein, um die Anforderung erfüllen zu können. Die Anwendungsform muss für die Umsetzung Schnittstellen bereitstellen, die auf die vorhandene Hardware optimiert sind.

Für eine Optimierung der Anwendung auf die Zielumgebung (N3) ist ein ästhetisches und minimalistisches Design die entscheidende Heuristik für die Auswahl der Anwendungsform. Für die Implementierung muss eine gute optische Integration in das System vorhanden sein. Dies ist für eine übersichtliche Aufbereitung des Produktkataloges eine Voraussetzung. Der App-Typ muss Oberflächenelemente zur Verfügung stellen, die zu der Gesamtumgebung passen. Durch eine gute Integration in die Zielumgebung wird dadurch die Anwendung ästhetischer. Die gesamte Wahrnehmung der App wird hierdurch verbessert. Das dritte Zielkriterium für die Auswahl ist somit die Verwendung von betriebssystemspezifischen Oberflächenelementen.

Im Folgenden werden die drei festgestellten Kriterien lokale Speicherung der Daten, Hardwarenahe Schnittstellen und Verwendung von betriebssystemspezifischen Oberflächenelementen für jede Anwendungsform bewertet, sodass am Ende eine Gegenüberstellung stattfinden kann.

**Native Anwendung:** Bei Nativen Anwendungen ist der komplette Funktionsumfang der mobilen Zielplattform verfügbar. Dies ermöglicht ein breites Anwendungsfeld für native Anwendungen.

- **Lokale Speicherung der Daten:** Die native Anwendungsform stellt Schnittstellen für den Zugriff auf eine lokale Datenbank oder ein lokales Dateisystem bereit. Diese können ohne Anpassungen verwendet werden.
- **Hardwarenahe Schnittstellen:** Dadurch, dass die einzelnen Schnittstellen direkt vom Hersteller der jeweiligen Plattform bereitgestellt werden, sind die Opera-

tionen für das jeweilige System optimiert. Mit einer nativen Anwendung wird damit die beste Performanz erreicht.

- **Betriebssystemspezifische Oberflächenelemente:** Native Anwendungen verwenden für die Benutzerschnittstelle die nativen Oberflächenelemente. Weiterhin werden spezielle Elemente bereitgestellt, die für eine Verwendung mit Touchscreen optimiert sind.

Auf die einzelnen Kriterien bezogen erfüllt die native Anwendung alle Anforderungen ohne Einschränkungen. Der volle Funktionsumfang des Systems ist gegeben, wodurch die App ideal an die gewählte Zielplattform angepasst werden kann.

**Web Anwendung:** Web Anwendungen werden im Browser ausgeführt. Dies führt dazu, dass nur Funktionen verwendet werden können, die der jeweilige Browser auf der Zielplattform zur Verfügung stellt.

- **Lokale Speicherung der Daten:** Ein Speichern und Laden der Daten vom Betriebssystem ist bei einer Web Anwendung nicht ohne weiteres möglich. Der Browser verbietet durch ein Sandbox-Modell einen direkten Zugriff auf das System. Ein Ansatz zur Lösung des Problems ist die Installation des Servers direkt auf dem Zielsystem, so dass dieser lokal zur Verfügung steht. Eine Erfüllung der Anforderungen ist damit möglich, jedoch mit erheblichem Mehraufwand.
- **Hardwarenahe Schnittstellen:** Die Kommunikation mit dem Betriebssystem erfolgt durch den Browser. Diese zusätzliche Zwischenschicht sorgt dafür, dass die Hardware nicht direkt verwendet wird, wie es bei einer nativen Anwendung der Fall ist. Der zusätzliche Overhead sorgt dafür, dass die Performance bei einer Web Anwendung nicht ideal im Vergleich zur nativen Lösung ist.
- **Betriebssystemspezifische Oberflächenelemente:** Die grundlegenden Oberflächenelemente sind HTML Elemente. Diese können mit Javascript und CSS einen passenden Stil für das jeweilige Betriebssystem erhalten. Die Verwendung von speziellen Touch Gesten ist mit Web Anwendungen aufgrund der Interoperabilität mit mehreren Betriebssystemen schwieriger. Hier sind ebenfalls nur begrenzte Interaktionen möglich.

Der Hauptvorteil der Web Anwendung die Verwendung auf mehreren Systemen ohne eine komplett neue Implementierung ist aufgrund der Kriterien des Anwendungsbeispiels nicht relevant. Die wichtigen Eigenschaften der Anwendung Performanz und

lokale Speicherung können mit dieser Anwendungsform gelöst werden, jedoch mit einem deutlichen Mehraufwand und Einbußen bei der Performanz.

**Hybride Anwendung:** Durch die Verwendung der lokalen Schnittstellen im nativen Anwendungscontainer (siehe 2.3.3) können die vorhanden Ressourcen wie bei einer Nativen Anwendung verwendet werden. Dies führt zu einigen Verbesserungen gegenüber einer reinen Web Anwendung.

- **Lokale Speicherung der Daten:** Für die Umsetzung dieser Anforderung muss eine geeignete Schnittstelle im Anwendungscontainer zur Verfügung gestellt werden. Hierdurch wird eine Verwendung des Dateisystems möglich. Das Kriterium kann ohne Einschränkungen erfüllt werden.
- **Hardwarenahe Schnittstellen:** Durch den Anwendungscontainer wird, wie bei einer Webanwendung der Browser, eine Zwischenschicht nötig. Dadurch entsteht ein zusätzlicher Kommunikationsaufwand. Dieser zusätzliche Aufwand verringert die Performanz der Anwendung.
- **Betriebssystemspezifische Oberflächenelemente:** Die Oberflächenelemente des Betriebssystems lassen sich im Container bereitstellen. Somit können die nativen Elemente in der Hybriden Anwendung verwendet werden. Die Verwendung der Touch Bedienelemente einer Plattform wird durch dieses Konzept ermöglicht.

Die Probleme, die bei der Verwendung einer Web Anwendung auftreten können durch den hybriden Ansatz gelöst werden. Ein Problem was weiterhin besteht, ist die Performanz. Diese wird mit der hybriden Anwendungsform besser, kommt jedoch nicht an die Leistung einer nativen Anwendung heran.

**Abwägung:** Nachdem alle Anwendungsformen auf die Erfüllung der aufgestellten Kriterien untersucht sind, kann eine Gegenüberstellung der einzelnen Komponenten erfolgen. Die Tabelle 4.1 zeigt eine Bewertung der Kriterien. Hierbei wurden die einzelnen Punkte anhand einer Skala von 0 (kann nicht erfüllt werden) bis 5 (ohne Einschränkungen möglich) bewertet. Die Implementierung als Web Anwendung scheidet klar aus. Jede Anforderung kann zwar umgesetzt werden, jedoch ist ein erheblicher Mehraufwand bei der Implementierung notwendig. Aus diesem Grund wird jeweils nur ein Punkt vergeben. Weiterhin hat die Anwendung im Vorhinein bereits Einschränkungen bzgl. der Performanz, sowie der Bereitstellung von Oberflächenelementen.

Kriterium	Web Anwendung	Hybride Anwendung	Native Anwendung
Lokale Datenspeicherung	1	4	5
Hardwarenahe Schnittstellen	1	2	5
Betriebssystemspezifische Oberflächenelemente	1	4	5
Summe	3	10	15

Tabelle 4.1.: Gegenüberstellung und Bewertung der Zielkriterien für die Anwendungsform

Die Vorteile einer Verfügbarkeit der Anwendung für mehrere Zielsysteme mit einer Implementierung ist bei der Umsetzung kein Kriterium.

Schwieriger ist die Entscheidung zwischen dem hybriden und nativen Ansatz. Die beiden Kriterien lokale Speicherung der Daten und betriebssystemspezifische Oberflächenelemente sind mit beiden Ansätzen ohne größere Einschränkungen möglich. Beim hybriden Ansatz ist ein geringer Mehraufwand bei der Implementierung nötig, deshalb wird hier nicht die volle Punktzahl gegeben. Dies ist jedoch nicht relevant für die Entscheidung. Der wichtigste Grund für die Entscheidung für eine nativen Anwendung ist die Performanz bei der Bedienung. Nur durch die Verwendung von nativen Anwendungen können aufwändige Animationen oder das Rendern von vielen Bildern flüssig ablaufen. Eine hybride Lösung besitzt hier Einschränkungen, da die Hardware mit einer Zwischenschicht verwendet wird, deshalb die Bewertung mit zwei Punkten. Weiterhin können die Touch Bedienelemente der Zielplattform bei einer nativen Anwendung besser verwendet werden, da keine Kompromisse notwendig sind.

#### 4.1.2. Anwendungstechnologie

Mit der Entscheidung für eine mobile Anwendungsform kommen drei mögliche Technologien für die Implementierung in Frage. Die Plattformen Android<sup>1</sup>, iOS<sup>2</sup> und Windows 8<sup>3</sup> sind aufgrund ihrer Marktanteile (Android: 43,4% iOS: 48,2% Windows:

<sup>1</sup> <http://www.android.com/>

<sup>2</sup> <http://www.apple.com/de/iphone/ios/>

<sup>3</sup> <http://windows.microsoft.com/de-de/windows-8/>

Kriterium	Android	iOS	Windows 8
Lokale Datenspeicherung	5	5	5
Hardwarenahe Schnittstellen	4	5	5
Betriebssystemspezifische Oberflächenelemente	5	5	5
Summe	14	15	15

Tabelle 4.2.: Gegenüberstellung und Bewertung der Zielkriterien für Plattform

7,4% Quelle: [Stu13] ) die wichtigsten Technologien im Tablet Bereich. Die drei Plattformen haben unterschiedliche Ziele. Das iOS Betriebssystem hat den Vorteil einer hohen Verbreitung bei Business Anwendungen (siehe [PAC11, S.5]). Android hat die meisten Nutzer bei Privatanwendungen (siehe [Stu13-2]). Bei Windows 8 ist der Vorteil eines neuen Konzeptes, was speziell für Tablet-PCs optimiert ist. Für die Umsetzung des Workflows kann jede dieser Technologien verwendet werden, da die Zielkriterien sich bei den unterschiedlichen Plattformen nicht unterscheiden (siehe Tabelle 4.2). Einziger Abzug erhält Android bei der Performanz, da eine Java-Virtual-Machine verwendet wird, die im Vergleich zu den C, bzw. C++ Schnittstellen der anderen beiden langsamer ist. Ist die Implementierung mit einem Framework umgesetzt, stellt die Übertragung auf einer anderen Zielplattform keine Herausforderung dar.

Bei der vorliegenden Arbeit wird auf Windows 8 gesetzt. Im Gegensatz zu iOS und Android wird dieses Betriebssystem nicht auf Smartphones ausgeführt. Dies führt zu einer besseren Optimierung für größere Bildschirme. Die Technologie ist sehr neu auf dem Markt (26. Oktober 2012), dadurch gibt es noch nicht viele Apps. Dies bietet die Möglichkeit den Anwender mit neuen Möglichkeiten in der Anwendung zu überraschen, da einige Konzepte noch nicht bekannt sind. Die Möglichkeit zu experimentieren und neue Ideen umzusetzen ist aufgrund der neuen Plattform vorhanden. Die Verwendung von Windows 8 vereinfacht ebenfalls die Integration in ein Unternehmensumfeld, da hier die Microsoft Produkte weit verbreitet sind. Dies führt zu einer schnelleren Akzeptanz im Unternehmen. Somit wird im Folgenden die Anwendung für das Windows 8 Betriebssystem implementiert.

## 4.2. Untersuchung der Plattform Windows 8

Damit die Anforderung eines ästhetischen und minimalistischen Designs (N3) erfüllt ist, muss vor der Gestaltung der einzelnen Ansichten die Zielplattform untersucht werden. Die Designgrundlagen müssen verstanden werden, um ein passendes Aussehen realisieren zu können. Die Folgende Untersuchung wird in drei Teile aufgeteilt. Im ersten Abschnitt werden allgemeine Design Prinzipien behandelt, darauf aufbauend die Bedienkonzepte und touchoptimierten Bedienelementen der Plattform.

### 4.2.1. Design Richtlinien

Damit einheitliche Apps entwickelt werden, hat Microsoft Richtlinien (Entnommen aus: [Win8], [Win8-1], [Win8-2], [Win8-3]) aufgestellt.

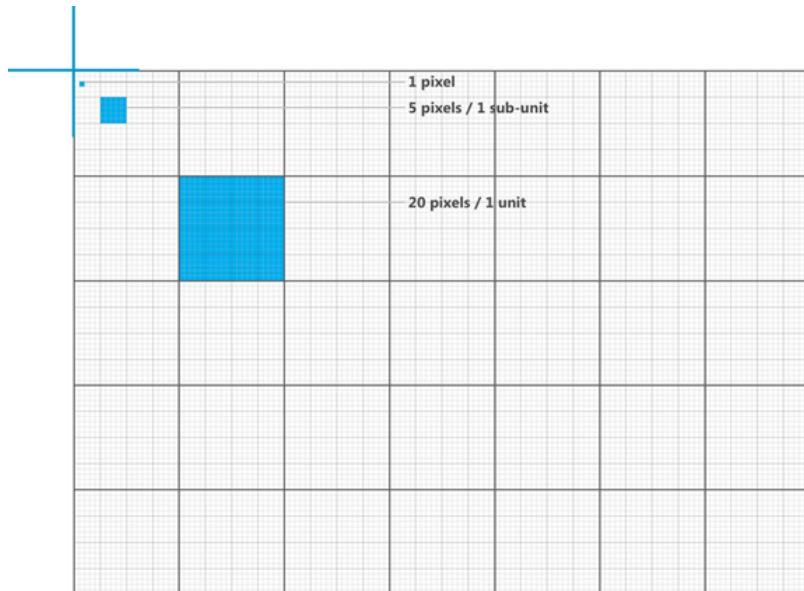


Abbildung 4.1.: Grid mit eingezeichneten Kacheln (Quelle: [Win8-5], Microsoft)

Das wichtigste Design Element bei einer Windows 8 Anwendung sind sogenannte Kacheln (siehe blaues Quadrat in Abbildung 4.1). Diese Kacheln sind quadratisch und in jeder App vorhanden. Jede Funktion wird über eine Kachel erreicht. Sie soll mehr Informationen enthalten, als ein einfaches Logo oder Icon auf einem Button. Durch einen dynamischen Inhalt und unterschiedliche Größen wird dem Benutzer ein neues Benutzererlebnis gegeben. Die Organisation der Kacheln erfolgt in einem sogenannten

Grid (engl. für Gitter). Dieses besteht aus mehreren Quadranten, wie in Abbildung 4.1 zu sehen. Das kleinste Quadrat hat eine Größe von einem Pixel. Das Grid besitzt verschiedene Bereiche für die Überschrift und den Inhalt. Dieser ist vorgegeben und sollte eingehalten werden.

Die Anwendung muss sich auf die Anzeige des Wesentlichen konzentrieren. Microsoft nennt das Prinzip “Content over Chrome”. Für die Umsetzung dieser Richtlinie soll die Anwendung nur die wichtigsten Funktionen in der Ansicht darstellen. Es sollen überladene Ansichten vermieden werden und stattdessen bewusst größere Elemente mit mehr Platz verwendet werden.

Ein weiteres wichtiges Design Element ist die Typographie. Hier geht es um die bewusste Gestaltung von Schriften. Die Kalligrafie wird als Vorbild verwendet. Die Idee ist keine rein statische Verwendung der Texte. Der Nutzer soll die Möglichkeit haben diese auszuwählen, damit eine Interaktion ermöglicht wird.

Die erläuterten Richtlinien werden beim Entwurf der einzelnen Ansichten berücksichtigt.

#### **4.2.2. Bedienkonzepte**

Bei den Konzepten für die Bedienung steht bei Windows 8 eine besondere Optimierung für Tablet-PCs im Vordergrund. Es werden deshalb sehr viele Gesten verwendet. Eine wichtige Geste ist das sogenannte “wischen”. Diese Aktion ist von jeder Seite des Bildschirms erlaubt. Beim Wischen von oben oder unten wird die sogenannte AppBar eingeblendet. Diese Bar wird an der oberen Seite für die Navigation durch die Anwendung verwendet. Dies ermöglicht einen schnellen Wechsel zwischen den Ansichten. Die untere AppBar wird für Aktionen verwendet, die nicht Vordergrund stehen. Ein Beispiel wäre die Filterung der Eingabedaten nach bestimmten Kriterien. Ein Wischen von der rechten Seite lässt die sogenannte CharmBar erscheinen. Der Inhalt dieser Bar ist die Verwendung von sogenannten Contracts (engl. Verträge). Diese Funktionen werden für alle Anwendungen durch das Betriebssystem bereitgestellt. An dieser Stelle können Einstellungen oder Suchen durchgeführt werden.

Das zweite wichtige Bedienkonzept ist das horizontale Scrollen. Aufgrund der größeren Bedienelemente sind nicht immer alle Elemente sichtbar. Die Lösung ist ein horizontales Ausbreiten des Inhalts. Damit die Inhalte verwendet werden können, wird ein horizontales Scrollen mit einer Wisch-Geste durchgeführt. Hier muss darauf geachtet

werden, dass ein Ausschnitt des nächsten Elementes sichtbar ist, damit dem Benutzer die Erweiterung der Ansicht signalisiert wird.

Beim Aufbau der Anwendung kann entweder eine hierarchische (Baumstruktur) oder flache (linear) Struktur verwendet werden. Für den in Abschnitt 3.2.2 erstellen Workflow ist eine hierarchische Architektur passender, da kein linearer Verlauf bei der Verwendung entsteht. Der Ursprung geht hierbei immer von der Startseite, der sogenannten Hub-Page aus. Diese Seite ist der zentrale Startpunkt, von der alle weiteren Aktionen ausgehen. Auf der zweiten Ebene befinden sich sogenannte Section Pages. Diese stellen den Inhalt einer Kategorie dar. Die unterste Ebene sind die Detail Pages. Diese enthalten die jeweiligen Details eines Elementes in einer Kategorie. Bei einer Zeitungs App wären beispielsweise auf der Startseite die einzelnen Kategorien wie Politik, Wirtschaft oder Sport zu sehen. In der Kategorie Ansicht die jeweiligen Überschriften der Artikel. Die Detail Seite würde den Artikel anzeigen. Dieser baumartige Aufbau wird durch das Verwenden eines Zurück Buttons unterstützt, der in jeder Ansicht, außer der Startseite vorhanden ist. Durch diesen Button wird dem Benutzer eine weitere Navigationsmöglichkeit gegeben.

Beim Design der Ansichten ist ein durchdachtes Bedienkonzept aufgrund der beiden Anforderungen N1 und N2 (einfache und schnelle Bedienung) wichtig. Die Möglichkeiten, die zur Verfügung gestellt werden, sollten beim ersten Entwurf der App enthalten sein.

#### **4.2.3. Touchoptimierte Bedienelemente**

Für die Unterstützung der Bedienung durch Gesten enthält das Framework besondere Oberflächenelemente. Diese sind für die Verwendung mittels Touch optimiert. Die in der Arbeit verwendeten Elemente werden im Folgenden vorgestellt. Die Design Richtlinien von Microsoft erzeugen Probleme bei der Darstellung von vielen Daten. Eine Umsetzung der Richtlinie "Content over Chrome", sowie die Darstellung auf einem Gerät mit kleinerem Bildschirm verursachen einen großen Aufwand beim Scrollen. Hierdurch kann es sein, dass eine lange Zeit für das Auswählen eines bestimmten Datums benötigt wird.

Für die Lösung dieses Problems bietet Windows 8 den sogenannten semantischen Zoom an. Diese Funktion wird mit einer Kneif-Geste, welche sich als allgemeine Geste für das Zoomen etabliert hat, auf dem aktuellen Datensatz durchgeführt. Hierdurch wird

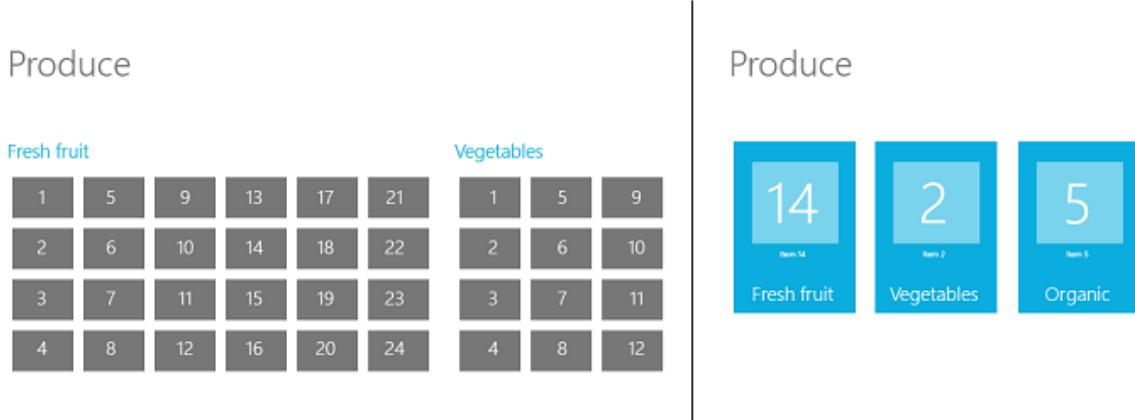


Abbildung 4.2.: Semantischer Zoom in zwei Ansichten (Quelle: [Win8-6], Microsoft)

die Ansicht nicht optisch verkleinert. Es erfolgt ein Wechsel von der Detailansicht zu einer Kategorieansicht. Die Datensätze werden somit semantisch verkleinert, wodurch ein schnelles Navigieren zum gewünschten Datum möglich ist.

Ein Beispiel ist in Abbildung 4.2 zu sehen. Hier sind in der linken Ansicht alle Elemente (Produkte) in der jeweiligen Kategorie vorhanden. Nach dem Anwenden der Kneif-Geste gelangt man zu einer zusammengefassten Ansicht (rechtes Bild), in der nur die Kategorien mit der Anzahl der vorhandenen Elemente sichtbar sind.

Der Semantische Zoom darf nicht geschachtelt verwendet werden und ist damit auf eine Ebene beschränkt. Voraussetzung für die Verwendung des Zooms ist die Einteilung der Daten in Kategorien. Ohne diese Kategorien kann keine übergeordnete Ansicht erstellt werden.

Das zweite neue Bedienelement, welches für eine Touch-Optimierung dient, ist die sogenannte Flip-View. Mit ihr kann durch ein Wischen von der linken oder rechten Seite die Ansicht gewechselt werden. Dieses Element kann den Benutzer bei einem schnellen navigieren helfen. Es sollten jedoch nicht zu viele Elemente für einen Wechsel enthalten.

### 4.3. Entwurf der Ansichten

Die vorgestellten Konzepte der Anwendungsplattform, sowie die einzelnen Bedienelemente werden für den Entwurf der Ansichten verwendet. Der Workflow der App, wie

in Abschnitt 3.3 beschrieben, wird hier als Grundlage verwendet. Damit alle Prozessschritte durchgeführt werden können, muss für jeden Zustand eine Ansicht vorhanden sein. Außerdem werden für die Umsetzung der Anforderungen aus Kapitel 3.2 evtl. zusätzliche Benutzerschnittstellen benötigt.

#### 4.3.1. Hauptfunktion der Anwendung

Um den Benutzer bei der Verwendung der App zu unterstützen, müssen zu Beginn die Hauptfunktionen der Anwendung erkennbar gemacht werden. Diese Funktionen sollen gut platziert werden, um eine schnelle Bedienung zu ermöglichen. Die Platzierung erfolgt in der Hauptansicht, von der alle weiteren Aktionen ausgehen.

Bei der vorliegenden Arbeit sind die Hauptfunktionen der Produktkatalog und die Konfiguration. Im Anwendungsbeispiel ist die Auswahl der Flugzeuge eine Voraussetzung für eine vollständige Konfiguration. Aus diesem Grund gehört die Flugzeugauswahl ebenfalls zu den Hauptfunktionen. Wie im neuen Workflow modelliert, sind diese drei Funktionen von der Zusammenfassungsseite aus zugänglich. Somit ist diese Seite die Hauptfunktionsseite

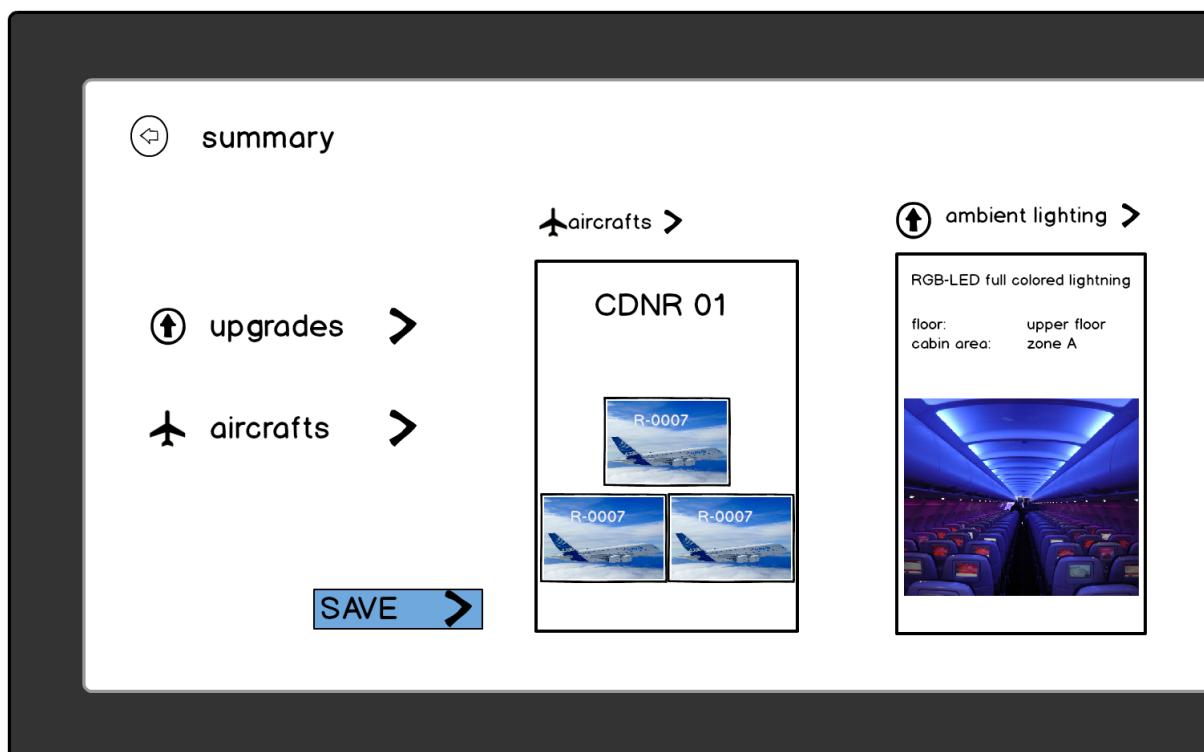


Abbildung 4.3.: Entwurf der Zusammenfassung Seite

Der erste Entwurf der Zusammenfassung ist in Abbildung 4.3 zu sehen. Die Hauptfunktionen Upgradeauswahl und Flugzeugauswahl sind als Buttons mit Text und Icon auf der linken Seite zu sehen. Bei der Selektion eines Buttons wird die jeweilige Ansicht der Auswahl geöffnet. Die weiteren Elemente der Zusammenfassung bestehen aus der aktuellen Auswahl. Die Darstellung erfolgt immer mit einem Bild und dem passenden Text in der Kachel Optik. Auf der ganz rechten Seite wird zuerst das ausgewählte Flugzeugprogramm angezeigt. Daneben werden die bisher ausgewählten Upgrades und Flugzeuge dargestellt. Die Reihenfolge der Darstellung hängt von der Auswahl ab. Diese werden in der gleichen Abfolge wie diese zuvor ausgewählt wurden angezeigt. Damit wird das zuletzt Gewählte immer auf der linken Seite angezeigt. Bei der Zusammenfassungsseite werden die Microsoft Richtlinien für das horizontale Scrollen umgesetzt. Die einzelnen Elemente sind im Grid angeordnet.

### 4.3.2. Produktkatalog

Für das Verwenden der ersten Hauptfunktion, der Produktauswahl wird der entsprechende Button in der Zusammenfassung verwendet. Beim Entwurf dieser Ansicht ist die Herausforderung das Produkt übersichtlich darzustellen. Da beim Anwendungsbeispiel ein großes und komplexes Produkt vorhanden ist, sind entsprechend viele Upgrademöglichkeiten vorhanden. Die Idee bei der Konzeption ist das Einteilen der einzelnen Upgrades in verschiedene Bereiche im Flugzeug. Beim derzeitigen Katalog sind die Upgrades bereits in Kategorien sortiert. Diese werden im Entwurf für die Einteilung auf die einzelnen Bereiche verwendet.

Abbildung 4.4 zeigt die Zuteilung der Kategorien auf die einzelnen Bereiche im Flugzeug. Wichtig hierbei ist das Verwenden von ansprechenden Bildern und großen Auswahlboxen. Der Kunde muss die benötigten Upgrades den einzelnen Bereichen zuordnen können und kann mit der Anwendung schnell zu seinem benötigten Produkt navigieren. Mit der Darstellung wird das Ziel erreicht, den Kunden für die Komplexität des Produktes zu sensibilisieren und ihm gleichzeitig eine bessere Suchmöglichkeit als über Produkt- oder Kategoriennummern zu ermöglichen.

Nach der Auswahl einer Kategorie wird nach dem in Abschnitt 4.2.2 vorgestellten Prinzip der hierarchischen Navigation eine Detail Seite angezeigt. Diese enthält die Details über die Upgrademöglichkeiten. Die Struktur beim Katalog des Anwendungsbeispiels enthält eine weitere Unterkategorie, welche im Entwurf (siehe Abbildung 4.5) auf der linken Seite als Liste dargestellt ist. Nach der Auswahl eines dieser Elemente

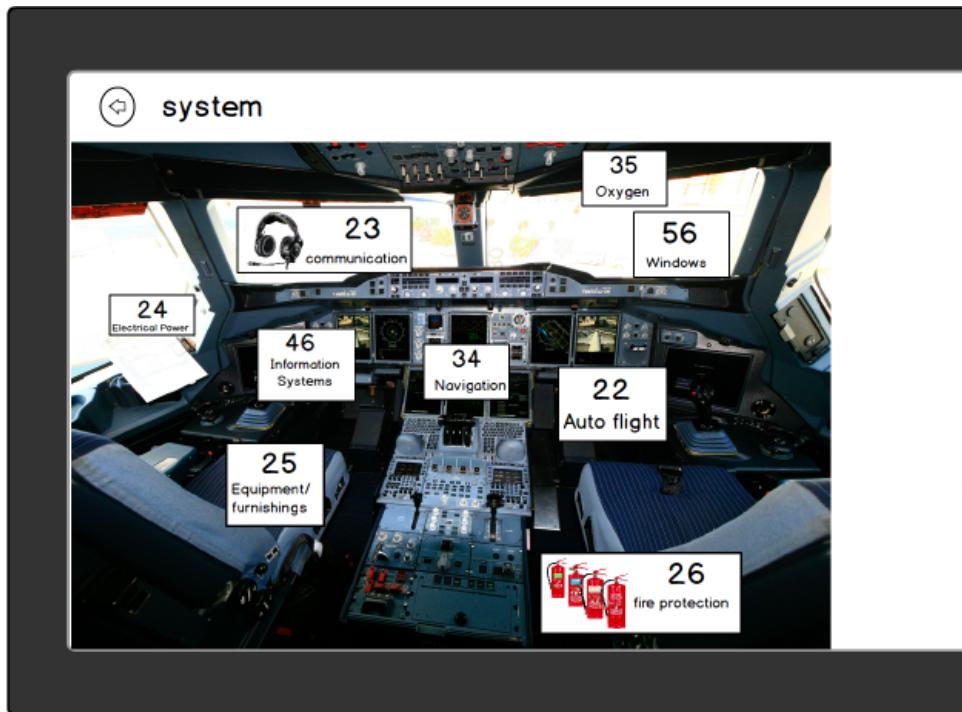


Abbildung 4.4.: Entwurf der Produktkategorie Auswahl

werden auf der rechten Seite die möglichen Upgrades der gewählten Unterkategorie angezeigt. Zusätzlich werden Informationen über die vorhandenen Möglichkeiten und optional Bilder für die Erklärung dargestellt. Für jedes vorhandene Upgrade in einer Unterkategorie werden einzelne Kacheln verwendet. Diese können per Klick selektiert werden. Die Informationen über das jeweilige Upgrade sind in der Kachel enthalten, ebenfalls ein Produktbild und ggf. der Name des Herstellers. Diese Elemente sind immer auf dem ersten Bildschirm ohne Scrollen sichtbar. Der Benutzer sollte nur bei Bedarf von mehr Informationen scrollen müssen. Hierdurch wird eine schnelle Auswahl der gewünschten Upgrades ermöglicht.

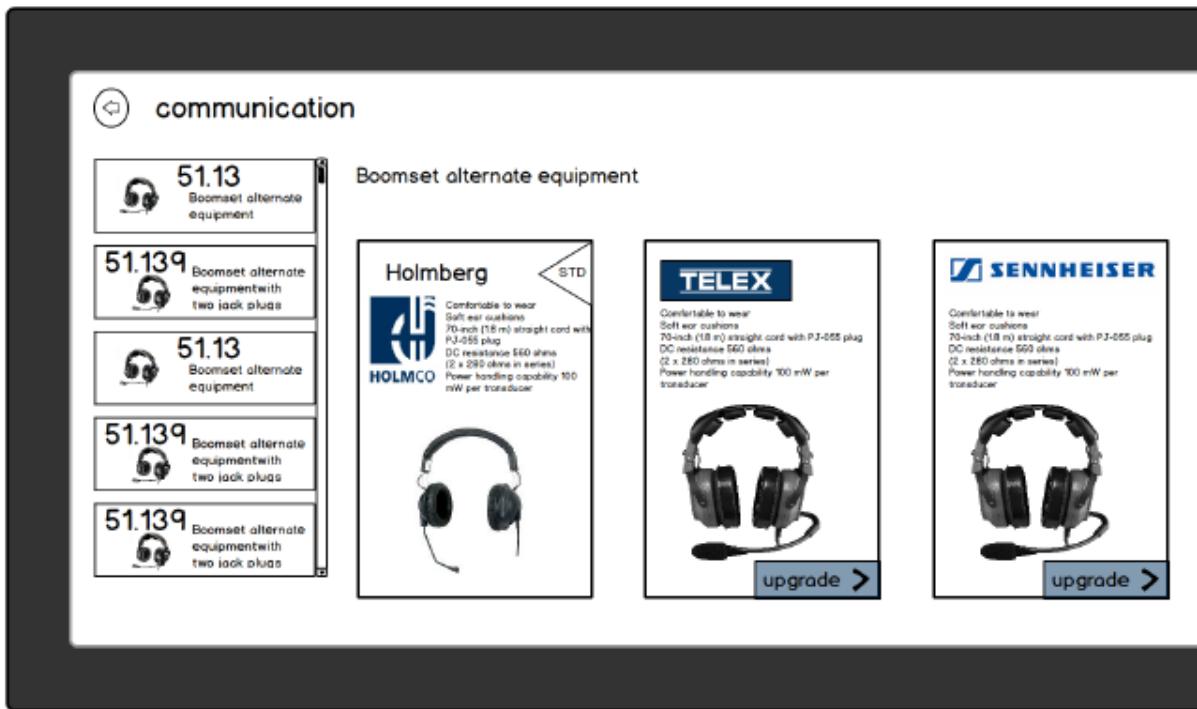


Abbildung 4.5.: Entwurf der Produktdetail Ansicht

### 4.3.3. Flugzeugauswahl

Die Auswahl der Flugzeuge ist aufgrund der Voraussetzung für die Konfiguration eine Hauptaufgabe. Für eine schnelle Selektion müssen die Daten zuvor gefiltert werden. Dadurch, dass die Anwendung beim Kunden direkt verwendet wird, sind auch nur dessen Daten relevant. Der Kunde verwendet für die Auswahl die sogenannte Flugzeugversion. Bei der Bestellung von neuen Flugzeugen werden diese in eine neue Version eingeordnet. Jede Flugzeuggesellschaft erhält ein Kürzel, worauf eine Nummer für die Version folgt. Bei der Auswahl der Flugzeuge wird diese Kategorisierung beibehalten. Die vorhandenen Datenelemente werden anhand der Flugzeugversion sortiert.

Die Darstellung der einzelnen Flugzeuge erfolgt, wie in Abbildung 4.6 zu sehen, im Grid. Die einzelnen Flugzeuge werden in einer Kachel angezeigt, wobei jede auswählbar ist. Die Sortierung der Daten erfolgt mit den Versionen als Überschrift der jeweiligen Gruppe. Damit die Daten schneller gefunden werden können, wird ein zusätzlicher Filter verwendet. Dieser soll die Flugzeuge nach dem jeweiligen Typ filtern und so für

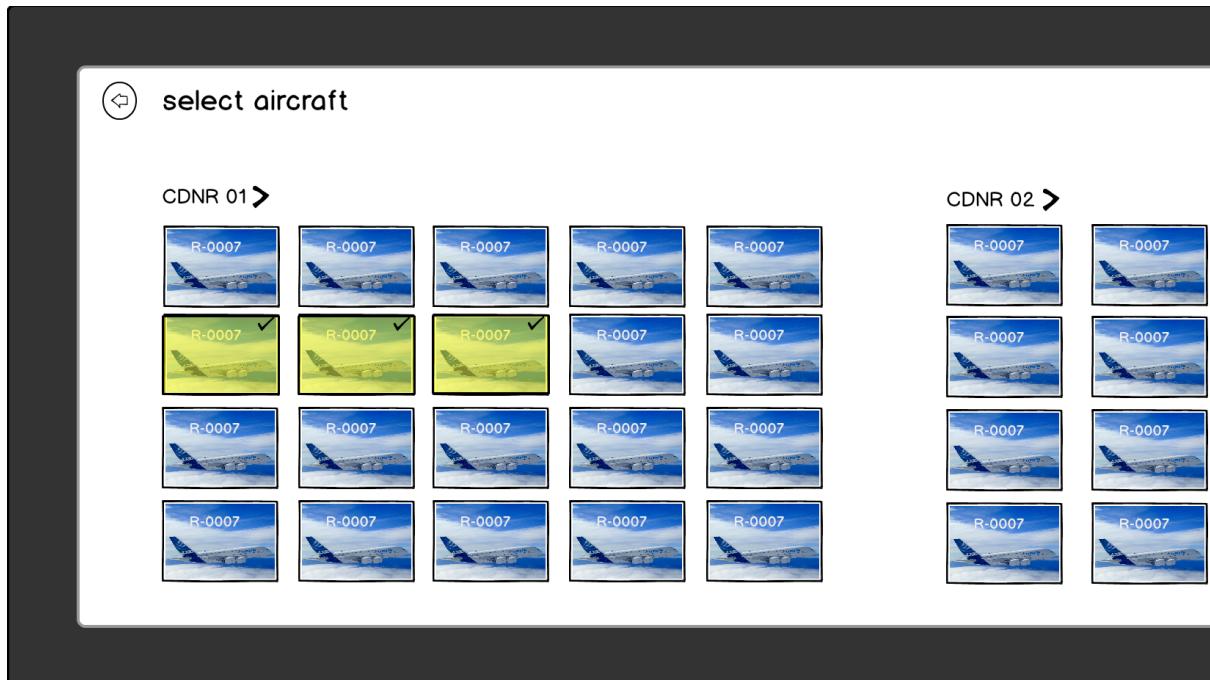


Abbildung 4.6.: Entwurf der Flugzeugauswahl

eine besser Übersicht sorgen. Der Filter wird über den Daten platziert und soll vom Scrollen ausgeschlossen werden.

Damit ein Experte beim Bedienen der Anwendung die Daten schneller auswählen kann (siehe N2), wird der semantische Zoom für diesen Datensatz verwendet. Die herausgezoomte Ansicht enthält die Versionen als Kategorie in einem Grid dargestellt.

#### 4.3.4. Konfigurationsergebnisse

Die Ergebnisse der Konfiguration werden im modellierten Workflow nach erfolgter Auswahl von Ugrades und Flugzeugen angezeigt. Das Ergebnis besteht aus den gebildeten Konfigurationsgruppen. Beim Entwurf sollen die Gruppen in der Zusammenfassungsseite zu sehen sein, sobald mindestens ein Flugzeug und ein Upgrade ausgewählt wurde. Die Kommunikation mit dem Konfigurationsserver läuft hier automatisch im Hintergrund, so dass jederzeit die Ergebnisse live angezeigt werden.

Für die Unterscheidung der verschiedenen Konfigurationsstatus werden verschiedene Farben verwendet (siehe Abbildung 4.7). Die Darstellung erfolgt nach der Ampel Semantik. Rot bedeutet in diesem Zusammenhang, dass die Konfiguration für bestimmte Flugzeuge nicht durchführbar ist. Bei gelb müssen Alternativen ausgewählt werden,

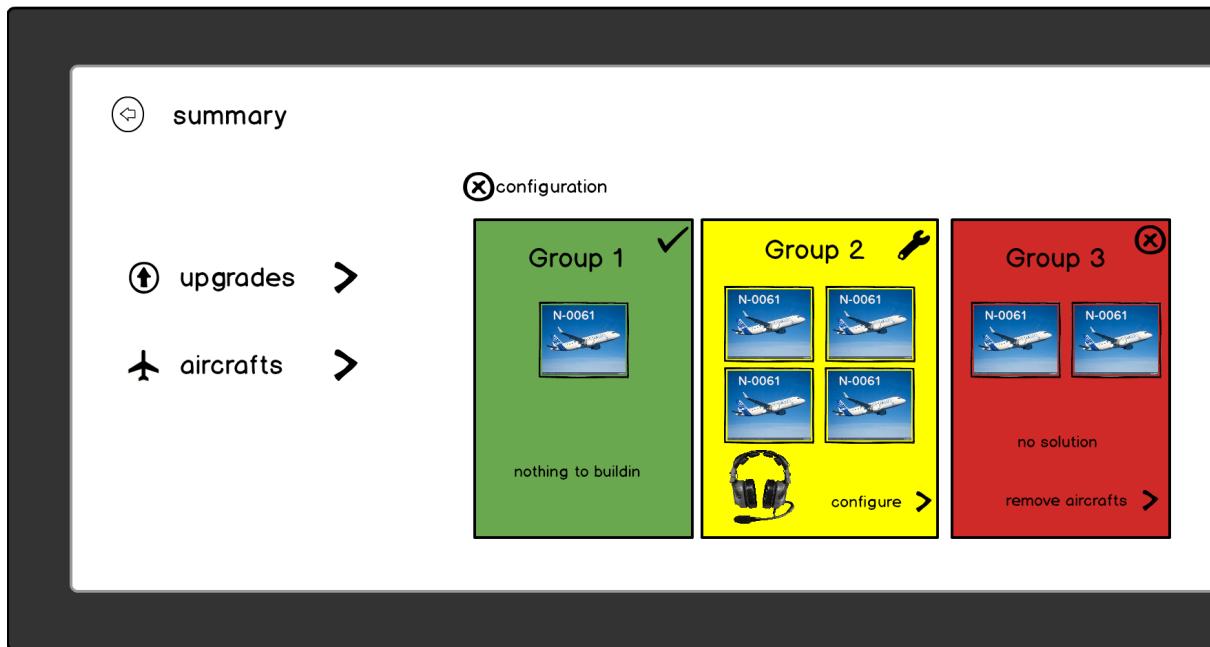


Abbildung 4.7.: Darstellung der Konfigurationsergebnisse

damit der grüne Status einer vollständigen Konfiguration für diese Gruppe erreicht wird. Die einzelnen Kacheln der Konfigurationsgruppen erhalten zusätzliche Icons, anhand derer ebenfalls eine Identifikation des Status abgelesen werden kann. Bei der Auswahl einer Konfigurationsgruppe gelangt man in die Alternativenauswahl.

Die Seite für die Auswahl der Alternativen enthält eine kleine Zusammenfassung der Upgrades und Flugzeuge. Für jede Alternative wird eine selektierbare Kachel angezeigt. Nach der Auswahl wird der Status der Konfigurationsgruppe geändert und die Konfiguration kann abgeschlossen werden, wenn alle Gruppen grün sind.

#### 4.3.5. Weitere Ansichten

Der Entwurf für die Hautfunktionen Upgradeauswahl, Flugzeugauswahl und Konfigurationsergebnisdarstellung ist abgeschlossen. Für die Umsetzung der funktionalen Anforderungen müssen weitere Ansichten hinzugefügt werden. Damit ein Speichern und Laden (F5) ermöglicht wird, soll eine Startseite verwendet werden.

Diese ist bereits im modellierten Workflow vorgesehen. Von der Startseite aus gelangt man in eine neue Konfiguration oder kann eine bereits vorhandene laden. Sie wird

beim Start der Anwendung angezeigt. Somit ist diese Ansicht bei jeder Verwendung der App zu sehen. Aus diesem Grund ist das Design der Startseite wichtig.

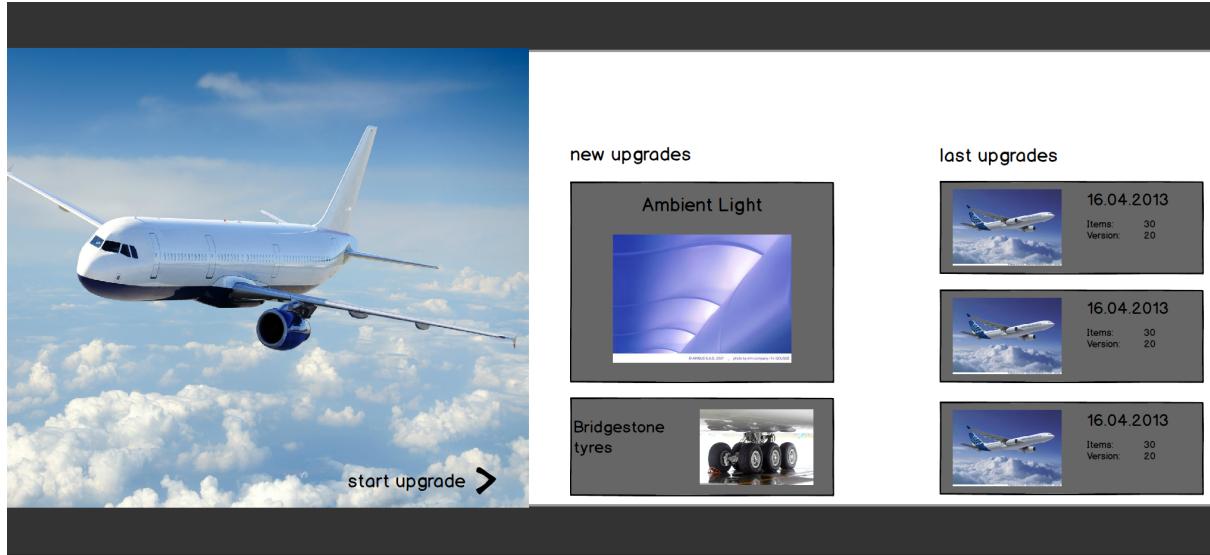


Abbildung 4.8.: Entwurf der Startseite

Die Idee ist es, eine kundenspezifische Startseite zu gestalten. Dies sorgt für eine Dynamik in der Seitenanzeige. Der Inhalt ist vom jeweiligen Kunden abhängig. Dies wird zuerst durch ein kundenspezifisches Bild zu Beginn erreicht (siehe Abbildung 4.8). Bei der Auswahl des Bildes wird eine neue Konfiguration gestartet. Die Größe wird so gewählt, dass die letzten Konfigurationen erst durch ein scrollen komplett angezeigt werden. Hierdurch wird die Hauptfunktion schneller zugänglich und die Microsoft Richtlinien der Konzentration auf das Wesentliche werden berücksichtigt. Als zweites Merkmal des Kunden wird das Logo der Fluggesellschaft am oberen linken Rand angezeigt. Für einen schnelleren Einstieg in die Anwendung werden neue oder für die Fluggesellschaft interessante Upgrades zusätzlich auf der Startseite angezeigt. Bei einem Klick auf diese Kachel wird die Detail Seite geöffnet, so dass mehr Informationen angezeigt werden und eine Auswahl möglich ist.

Für die Umsetzung der funktionalen Anforderung F6, die das Filtern der Anwendungsdaten spezifiziert, wird eine weitere Ansicht benötigt. Die Daten sind in der Flugzeugauswahl und in der neuen Startseite auf einen Kunden beschränkt. Andere Datensätze werden nicht benötigt. Das Problem mit der Filterung kann aus diesem Grund mit einer Kundenauswahl gelöst werden. Diese Auswahl stellt im Anschluss

alle kundenspezifischen Daten der Anwendung bereit und erfüllt die Anforderung eines entsprechenden Filters.

Die Ansicht für die Auswahl der Kunden wird analog zu der Flugzeugauswahl gestaltet. Es wird hier ebenfalls für jeden Kunden eine eigene Kachel angezeigt. Diese sind im Grid angeordnet. Eine Einteilung in Kategorien erfolgt Alphabetisch. Im Anwendungsbeispiel sind sehr viele Kunden vorhanden. Damit eine schnelle Auswahl erfolgen kann, wird der Semantische Zoom in der Ansicht verwendet. Die Kategorisierung erfolgt über das Alphabet. Es kann jedem Buchstaben ein Kunde zugeordnet werden. Nach der Auswahl eines Datums durch den Experten werden die kundenspezifischen Anwendungsdaten auf das mobile Endgerät geladen und können anschließend beim Kunden "offline" verwendet werden.

Die zweite Filterung der Daten wird über die Programmauswahl realisiert. Diese Auswahl wird beim Starten einer neuen Konfiguration von der Startseite aus aufgerufen. Der Programmfilter wählt den richtigen Produktkatalog aus und verringert die Menge der vorhandenen Flugzeuge. Bei der visuellen Darstellung der Ansicht werden die vier möglichen Programme angezeigt. Damit der Kunde die Programmauswahl versteht, wird für jedes Programm ein passendes Flugzeugbild gewählt. Für die Umsetzung der "Content over Chrome" Richtlinie erhalten die vier Kacheln eine Bildschirmfüllende Größe, so dass die Auswahl mit Touch vereinfacht wird.

## 4.4. Navigation und Bedienung

Alle benötigten Ansichten sind den Anforderungen entsprechend konzipiert. Damit die Anforderung einer einfachen Bedienung (N1) erfüllt ist, muss ein passendes Navigationskonzept erstellt werden. Die Navigation unterstützt den Benutzer, indem es Fehler bei der Anwendung verzeiht. Hierzu gehört beispielsweise ein schnelles Zurückspringen nach einer falschen Auswahl. Weiterhin muss ein schneller Ablauf des normalen Anwendungsfalls unterstützt werden. Der modellierte Workflow ist die Ausgangsbasis der Navigation. Dieser wird mit den zusätzlichen Ansichten und neuen Navigationsmöglichkeiten erweitert.

#### 4.4.1. Navigationsverlauf

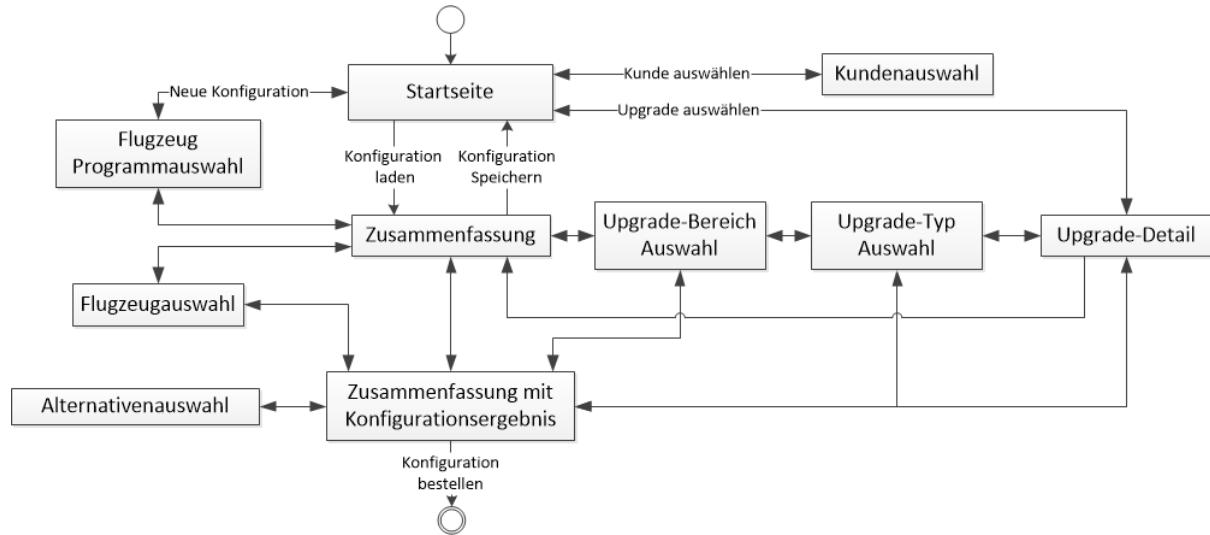


Abbildung 4.9.: Navigation in der App

Abbildung 4.9 zeigt die Möglichkeiten bei der Navigation. Es wird dabei eine hierarchische Struktur verwendet. Der Ursprungspunkt der Anwendung ist die Startseite. Von ihr können die vier Ansichten Programmauswahl, Zusammenfassung, Upgradedetail und Kundenauswahl erreicht werden. Nach der Auswahl eines Kunden werden die Kundendaten geladen und es erfolgt ein Rücksprung zur Startseite. Nach Auswahl der drei anderen Möglichkeiten ist das Ziel die Zusammenfassung zu erreichen. Dies wird bei einer neuen Konfiguration erst nach der Auswahl eines Programms möglich. Bei der Selektion eines Upgrades wird von der Detailseite aus zurückgesprungen. Aus der Zusammenfassung kann in jede Hauptfunktion navigiert werden. Damit ist immer eine Ausgangsbasis vorhanden, anhand der Benutzer sich orientieren kann. Wenn sowohl Flugzeuge, als auch Upgrades ausgewählt sind, werden die Konfigurationsergebnisse in der Zusammenfassung angezeigt. Nachdem die benötigten Alternativen ausgewählt wurden, kann der Konfigurationsprozess durch die Bestellung abgeschlossen werden.

#### 4.4.2. Bedienelemente

Damit die oben genannten Navigationsmöglichkeiten umgesetzt werden können, müssen passende Bedienelemente der Zielplattform ausgewählt werden. Hierzu werden folgende vier Möglichkeiten der Navigation eingesetzt:

**Auswahl durch Kacheln:** Zu den Hauptfunktionen der Anwendung wird mit großen Auswahlflächen navigiert. Diese besitzen ein aussagekräftiges Icon und werden direkt auf der Oberfläche angezeigt (siehe Entwürfe).

**Obere AppBar:** Für die Navigation zu allen wichtigen Ansichten wird zusätzlich eine Möglichkeit in der oberen AppBar gegeben. Über dieses Bedienelement kann zu der Startansicht, Flugzeugauswahl, Upgradeauswahl und Zusammenfassung navigiert werden (Abbildung 4.10). Diese Navigationsmöglichkeit ist in jeder Ansicht möglich und hilft dem erfahrenen Benutzer bei einer schnellen Bedienung (N2).



Abbildung 4.10.: Obere AppBar für die Navigation

**Untere AppBar:** Die Kundenauswahl ist nur für den Vertriebsexperten oder den Hersteller von Interesse. Aus diesem Grund wird die Navigation von der Startseite aus über die untere AppBar (siehe 4.11) ermöglicht. Weiterhin wird diese Leiste für ein schnelles Navigieren von der Detailansicht zur Zusammenfassung verwendet. Nach der Auswahl eines Upgrades wird durch einen Klick auf den entsprechenden Menüeintrag (rechter Button im Bild) in der unteren AppBar der Auswahlvorgang abgeschlossen und es erfolgt eine Navigation zur Zusammenfassung.



Abbildung 4.11.: Untere AppBar für die Navigation

**Zurück Button:** In jeder Ansicht, außer der Startseite, wird dem Benutzer das Zurückkehren zur vorigen Ansicht ermöglicht. Hierdurch kann eine unbeabsichtigte Navigation schnell rückgängig gemacht werden.

## 4.5. Expertenmodus

Die Anforderung einer schnellen Bedienung ist mit einem komplexen Navigationskonzept erfüllt. Damit nach der Heuristik einer flexiblen Nutzung weitere Möglichkeiten bei der Anwendung existieren, werden weitere Konzepte für die Erfüllung der Anforderung benötigt. Bisher wurden nur die Bedürfnisse des Endkunden berücksichtigt. Das primäre Ziel beim Entwurf war es, dass komplexe Produkt zu vereinfachen und die Auswahl auf einem angenehmen Weg durchzuführen. Die zweite Zielgruppe der Anwendung sind weiterhin die Experten. Diese wollen die App auf eine effizientere Weise nutzen können. Da Produktkenntnisse vorhanden sind, können andere Ansätze bei der Navigation eingesetzt werden. Das Ziel des Entwurfs ist es dabei, einen sogenannten Expertenmodus bereitzustellen.

### 4.5.1. Einsatz der Flip View

Im neuen Navigationsworkflow (siehe 4.9) müssen für die Auswahl eines Upgrades mehrere Vorselektionen durchlaufen werden. Dieser Mechanismus ist für den Kunden ideal, da er so den Aufbau des Produktes sowie die einzelnen vorhandenen Möglichkeiten versteht. Für den Experten, der den Aufbau kennt und genau weiß, was er will, ist dies unnötig. Hier muss eine weitere Möglichkeit geschaffen werden, wie die Auswahl schneller erfolgen kann.

Der Lösungsansatz besteht im Verwenden der Flip View. Dieses Oberflächenelement wird vom Framework bereitgestellt und ermöglicht ein schnelles Wechseln der aktuellen Seite durch eine Wisch Geste. Die Idee ist das Verwenden dieser Komponente in der Upgrade-Bereich Auswahl. Abbildung 4.12 zeigt den Entwurf dieser Ansicht. Der normale Benutzer sieht nur das Bild in der Mitte und kann den gewünschten Flugzeugbereich auswählen, um danach den Upgrade-Typ zu wählen. Für eine schnellere Nutzung kann durch einen Wisch nach rechts oder links eine neue Seite angezeigt werden, die ein Inhaltsverzeichnis mit allen Produktkategorien enthält. Alle Typen lassen sich in die beiden Bereiche System und Cabin einordnen, weshalb nur zwei solcher Verzeichnisse benötigt werden. Der Experte wählt eine Kategorie aus und wird direkt zur Upgradeauswahl navigiert. Damit wird die Typ Auswahl übersprungen und Upgrades können direkt ausgewählt werden.

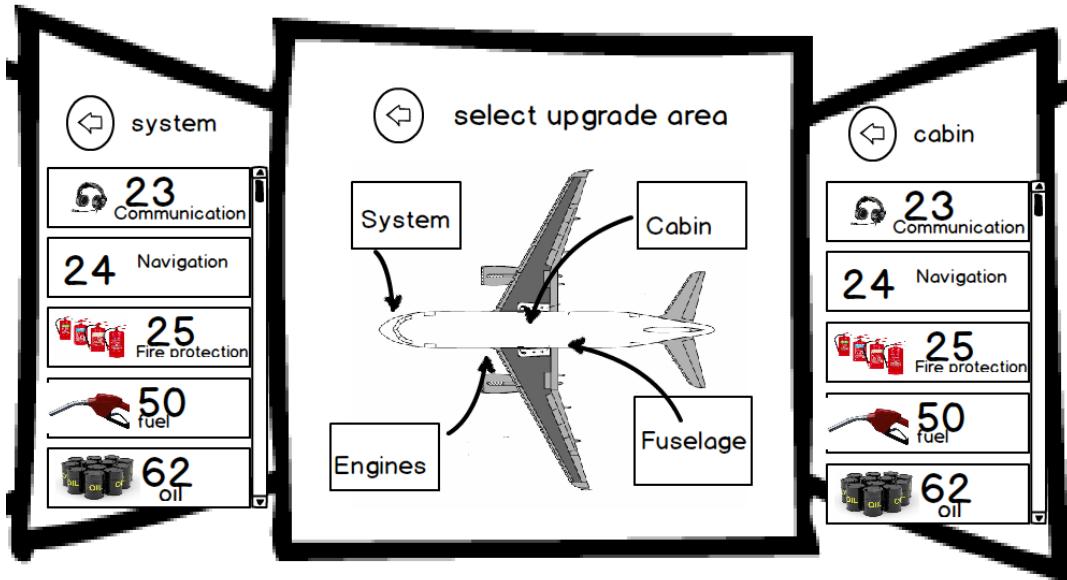


Abbildung 4.12.: Flip View in der Upgrade Bereich Auswahl

#### 4.5.2. Erweiterung des Semantischen Zooms

Die Entwürfe für die Kunden- und Flugzeugauswahl sehen die Verwendung des semantischen Zooms vor. Dieser hilft bei einer schnelleren Auswahl der Flugzeuge. Beim ursprünglichen Entwurf werden die einzelnen Flugzeuge nach den Versionen eines Flugzeuges gruppiert. Damit der Experte auch an dieser Stelle Flugzeuge schneller auswählen kann, wird der semantische Zoom erweitert. Des Weiteren wird eine flexible Nutzung dadurch ermöglicht, dass der Benutzer eine Kategorie selbst festlegen kann. Dies wird mit einer Dropdown Liste im Kopfbereich der Flugzeugauswahl realisiert (siehe 4.13). Nach der Auswahl einer Kategorie erfolgt eine Neugruppierung der Datensätze und der semantische Zoom wird somit erweitert. Durch diese Maßnahme erhält der erfahrene Benutzer eine weitere Möglichkeit, die einzelnen Flugzeuge schnell zu finden und somit eine effektive Auswahl durchzuführen.

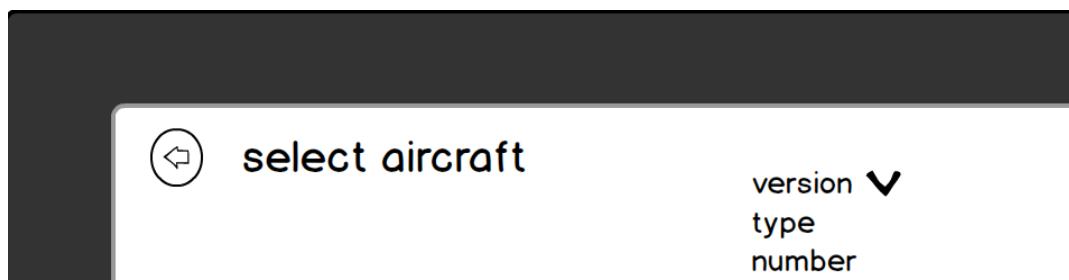


Abbildung 4.13.: Dropdown List bei der Flugzeugauswahl

# **5. Implementierung**

Die Implementierung hat zum Ziel die zuvor entworfenen Ansichten umzusetzen und die Realisierung der Anwendung auf der Zielplattform zu realisieren. Im Folgenden werden die Architektur und der Aufbau der Implementierung beschrieben. Anschließend werden die wichtigsten Entscheidungen, die bei der Implementierung getroffen wurden, erklärt. Die erzielten Ergebnisse werden im Anschluss präsentiert.

## **5.1. Anwendungsarchitektur**

Damit die Entwicklung der Anwendung beschleunigt werden kann, indem wieder verwendbare Muster in der Anwendung verwendet werden, wird zuerst eine geeignete Anwendungsarchitektur benötigt. Als erste Voraussetzung muss diese von der Zielumgebung, bzw. von der Technologie unterstützt werden.

Aufgrund des Offline-Modus in der Anwendung sowie der Verwendung des Konfigurationsservers müssen zwei unterschiedliche Formen der Datenanbindung unterstützt werden. Dies hat zur Folge, dass ein einfacher Austausch der Datenanbindung in der Anwendung möglich sein muss, ohne eine Neuimplementierung der Schnittstellen auszuführen. Die Anforderung an ein ästhetisches Design kann durch eine klare Trennung der Ansicht mit den Logikkomponenten erfüllt werden. Damit ist es möglich, die Gestaltung frei von der notwendigen Logik umzusetzen und sich auf die Gestaltung der Benutzerschnittstelle zu konzentrieren. Die Architektur muss ebenfalls für Erweiterungen offen sein, damit zusätzliche Anforderungen, die beim Einsatz der Anwendung entstehen, umgesetzt werden können.

### 5.1.1. Model-View-ViewModel

Eine Lösung für die oben genannten Anforderungen an die Architektur bietet das von Microsoft entwickelte Model-View-ViewModel (MVVM) Entwurfsmuster. Hier wird eine strikte Trennung zwischen der Ansicht (View), der Logik (ViewModel) und den Daten (Model) vorgenommen.

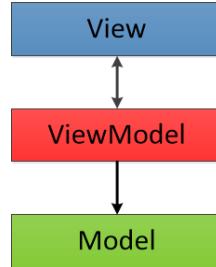


Abbildung 5.1.: Komponenten und Kontrollfluss im MVVM Entwurfsmuster

Die Kommunikation der einzelnen Komponenten ist in Abbildung 5.1 dargestellt. Die unterste Ebene ist die Model-Schicht. Diese ist für das Bereitstellen und Persistieren der Daten zuständig. Es wird hierzu entweder eine Datenbank oder Webserviceschnittstelle verwendet. Wichtig bei dieser Schicht, wie in der Abbildung zu sehen, ist der unidirektionale Kontrollfluss mit dem ViewModel. Damit ist eine Manipulation der Daten nur von einer Stelle aus möglich. Dies vereinfacht das Finden von Fehlern.

Auf der anderen Seite ist die View. Diese Komponente ist für das Darstellen der Daten zuständig. Es werden alle Oberflächenelemente einer Benutzerschnittstelle auf dieser Ebene verwendet. Die Interaktionen des Benutzers werden auf dieser Anwendungsschicht durchgeführt. Die Auswertung der Eingaben folgt im "Modell der Ansicht" [Smi10, S.9], dem ViewModel. Diese Schicht ist der Vermittler zwischen den Daten und der Benutzerschnittstelle. Aus diesem Grund werden die Daten, die vom Model erhalten werden für die Ansicht aufbereitet. Die Verbindung zur View-Ebene ist dabei bidirektional, damit sowohl Benutzereingaben, als auch Veränderungen im ViewModel registriert werden.

Die Anforderung für einen einfachen Austausch der Datenquelle wird durch die Unabhängigkeit des Models erfüllt. Hierdurch können die Daten sowohl auf dem Gerät, als auch mit Webserviceschnittstellen geladen werden. Das MVVM Entwurfsmuster wird von der Technologie unterstützt und es sind bereits Codebeispiele vorhanden [Bug13].

Mit der Trennung von View und ViewModel ist ein einfaches Erstellen von Benutzeroberflächen möglich. Aufgrund der Unabhängigkeit beider Komponenten können diese auch separat entwickelt werden. Hierdurch wird es beispielsweise möglich, dass ein Designer und ein Programmierer unabhängig voneinander arbeiten können. Diese Vorteile sind der Grund für die Entscheidung zur Verwendung des MVVM Design Patterns anstatt des gängigen Model-View-Controllers (MVC), bei dem die View vom Controller abhängig ist und ein unabhängiges Testen der beiden Schichten nicht möglich ist.

### 5.1.2. Anwenden des Entwurfsmusters

Bei der Implementierung der Anwendung mussten für die Anzeige der Daten immer wiederkehrende Eigenschaften der Datensätze verwendet werden. Ein Beispiel für eine solche Eigenschaft ist der Name oder die Beschreibung eines Flugzeuges oder Upgrades. Da beim MVVM Entwurfsmuster die View nicht auf das Model zugreift, kennt es diese Datenobjekte nicht. Aus diesem Grund muss das ViewModel diese Daten konvertieren und ein neues Objekt bereitstellen.

Ebenfalls muss bei einer Manipulation oder Auswahl der Daten die Konvertierung rückgängig gemacht werden, damit das Model die Änderungen vornehmen kann. Diese Vorgehensweise hat bei einer Veränderung der Daten zur Laufzeit Vorteile, da das ViewModel den Zeitpunkt der Persistierung entscheiden kann. Im Anwendungsbeispiel ist dies jedoch ein zusätzlicher Aufwand, der nicht benötigt wird, da keine Daten manipuliert werden, sondern eine Auswahl getätigt wird. Die eigentliche Datenbasis wird in der Anwendung nicht verändert. Für die Vermeidung des zusätzlichen Aufwandes wurde eine neue Komponente eingeführt. Auf diese haben alle drei Ebenen im MVVM Zugriff. In dieser Komponente sind die festen Datenelemente definiert und es wird nur ein lesender Datenzugriff ermöglicht. Die konkrete Implementierung der Datensätze wird weiterhin im Model vorgenommen. Damit muss keine Konvertierung der Daten erfolgen, um der View einen Zugriff auf die Daten zu geben.

Abbildung 5.2 zeigt einen Ausschnitt des Klassendiagramms. Anhand der Auswahl des Upgrades, wie im Navigationsverlauf (siehe 4.9) zu sehen, wird die Anwendung des MVVM Entwurfsmusters demonstriert. Es werden hierfür die drei Ansichten SelectUpgradeTypePage, SystemUpgradePage und SelectUpgradePage benötigt. Alle drei erben von der Klasse Page, welche die vom Framework bereitgestellte Oberflächenklasse ist und für eine Seite der Anwendung steht. Die Definition der Page und der

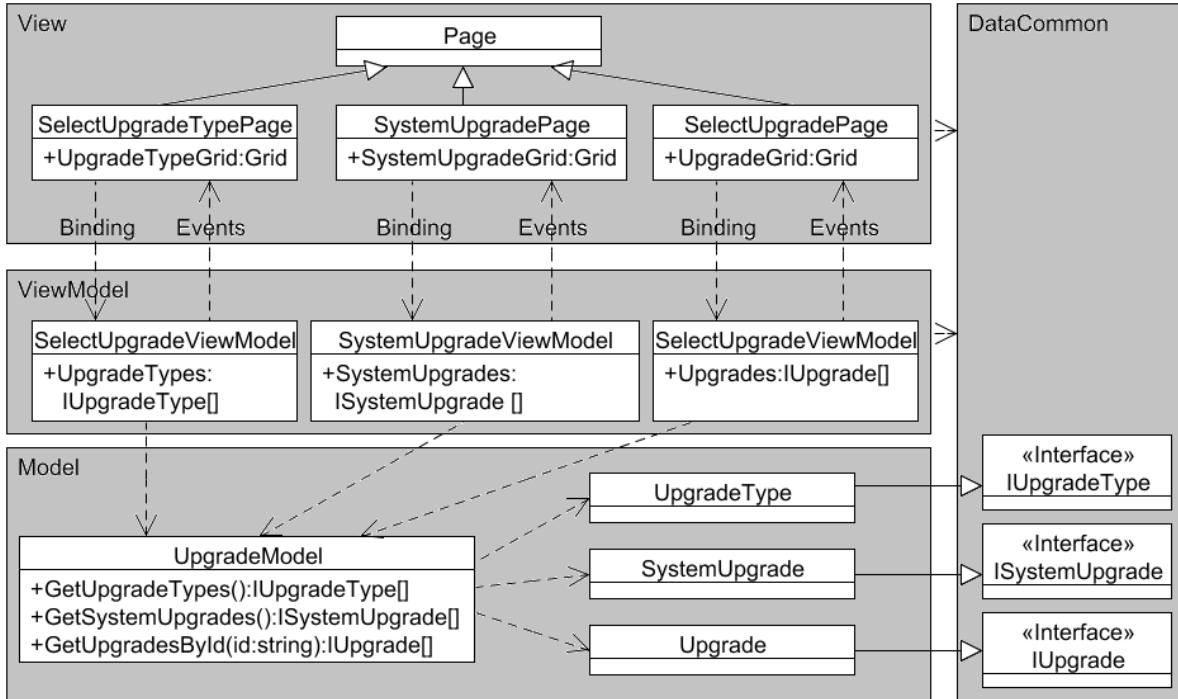


Abbildung 5.2.: Klassendiagramm im MVVM Entwurfsmuster

enthalteten Oberflächenelemente erfolgt in sogenanntem XAML Code. Dieser ist eine XML-basierte Sprache für das deklarative Erstellen von Benutzerschnittstellen. Alle diese Komponenten lassen sich in die View-Schicht einordnen.

Zu jeder Ansicht existiert ein passendes ViewModel. Damit eine bidirektionale Verbindung zustande kommt, wird für die Kommunikation von der View zum ViewModel sogenannte Bindings verwendet. Ein solches Binding bindet ein Oberflächenelement an das passende Datenelement im ViewModel. Im Beispiel ist das Grid Oberflächenelement an ein Array im ViewModel gebunden. Die Kommunikation in die entgegengesetzte Richtung erfolgt mit Events. Bei einer Änderung der Daten im ViewModel wird ein sogenanntes PropertyChangedEvent ausgeführt. Nachdem die View das Event erhalten hat, werden die Daten in der Ansicht aktualisiert.

Für die drei ViewModel Klassen existiert eine gemeinsame Model Klasse. Da die Daten voneinander abhängig sind, ist eine gemeinsame Verwendung sinnvoll. Das Model stellt für jedes ViewModel die passenden Schnittstellen bereit. Die Datenobjekte UpgradeType, SystemUpgrade und Upgrade sind in der Model Schicht vorhanden und werden als Datenbasis verwendet. Die bereitgestellten Methoden verwenden Objekte dieser Klassen. Damit die Datenobjekte auch in der View verwendet werden können,

ist die Definition als Interface in die DataCommon Komponente ausgelagert. Diese kann von allen drei Schichten der Anwendung verwendet werden. Im Beispiel wird die Verwendung in den ViewModel Klassen demonstriert. Hier werden die Datenelemente, die für das Binding mit der View benötigt werden als ein Array des Interface bereitgestellt. Dies sorgt für eine einheitliche Kommunikation der drei Ebenen und vermeidet eine Konvertierung der Daten im ViewModel.

## 5.2. Navigation

Eine besondere Herausforderung bei der Implementierung ist die Navigation. Diese muss zuerst in das Entwurfsmuster eingeordnet werden. Es muss entschieden werden, in welcher Ebene navigiert wird. Der Wechsel der Ansichten ist eine Aufgabe der View. Diese bestimmt, wie zu einer neuen Seite navigiert wird. In dieser Ebene werden auch die konkreten Navigationsmethoden angeboten. Andererseits ist die Entscheidung darüber, wann in welche Ansicht gewechselt wird eine Angelegenheit des ViewModels. Diese wertet die Auswahl eines Klicks aus und ist damit für dessen Bearbeitung zuständig. Weiterhin werden Objekte zwischen den beiden ViewModels ausgetauscht. Aus diesem Grund muss es eine Möglichkeit für den Austausch geben.

Die Lösung des Problems erfolgt mit dem Ansatz der Inversion of Control [Fow04]. Bei diesem Prinzip geht es um die Auflösung von Abhängigkeiten. Anstatt ein Objekt direkt zu erzeugen, wird es von einer zentralen Stelle verwendet. Die Abhängigkeit wird damit von außerhalb des aktuellen Codes hinzugefügt. Dieser Ansatz hat besonders bei Softwaretests große Vorteile, da andere Objekte für Testzwecke verwendet werden können.

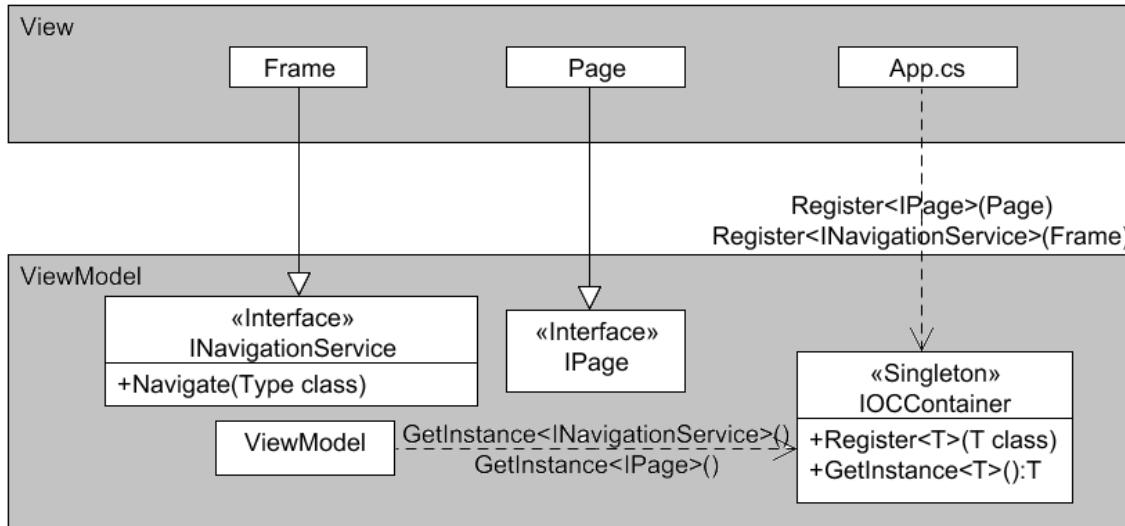


Abbildung 5.3.: Navigation mit Inversion of Control

Die Umsetzung des Prinzips ist in Abbildung 5.3 dargestellt. Damit eine Verwendung der einzelnen Klassen aus der View im ViewModel möglich ist, wird für jede Ansicht (Page) ein passendes Interface (IPage) definiert. Eine Navigation wird beim Windows 8 Framework mit der Frame Klasse durchgeführt. Dieses implementiert das INavigationService Interface, welches zwei Navigationsmethoden beinhaltet. Eine Methode kann mit einem Parameter für das neue ViewModel ausgeführt werden, die Andere ohne Parameter. Beim Start der App Klasse werden die einzelnen Pages und der Navigationsframe mit den definierten Interfaces im IOCContainer registriert. Der Container befindet auf der ViewModel Ebene. Für eine Navigation werden die Zielpage und der Navigationsservice mit der GetInstance<T> Methode erhalten. Anschließend kann die passende Navigationsmethode entweder mit oder ohne Parameter ausgeführt werden.

In Codebeispiel 5.4 wird ein solcher Navigationsvorgang im ViewModel demonstriert. Dieser Auszug ist aus der Flugzeugprogramm Auswahl entnommen. Nachdem ein Programm ausgewählt wurde, wird zuerst die Auswahl im Model gespeichert. Im zweiten Schritt erhält man die Zusammenfassungsseite mit dem Interface (ISummary) vom Container. Die Navigation erfolgt anschließend mit dem Navigationsservice.

Mit diesem Ansatz gelingt es, die Darstellung und die Entscheidung, welche Ansicht bei welchem Inhalt verwendet wird, der View Schicht zu überlassen. Der Wechsel von Ansichten wird mit Methoden durchgeführt, die eine Ansicht bereitstellt. Das

```

private void SaveSelectionAndNavigateToSummaryPage(DataCommon data)
{
    var selectedProgramm = GetSelectedProgramm(data.UniqueId);
    _model.SelectAircraftProgramm(selectedProgramm);
    var classToNavigate = SimpleIoc.Default.GetInstance<ISummary>();
    var navigationService = SimpleIoc.Default.GetInstance<
        INavigationService>();
    navigationService.Navigate(classToNavigate.GetType());
}

```

Abbildung 5.4.: Auszug des AircraftFamilyViewModels (siehe Anhang A)

ViewModel kann entscheiden, wann eine Navigation durchgeführt wird. Durch die Verwendung des Containers ist eine zentrale Stelle vorhanden, die jederzeit verwendet werden kann. Bei einer Navigation können Parameter übergeben werden, die eine Kommunikation der ViewModels ermöglicht.

## 5.3. Ergebnisse der Implementierung

Die Entwürfe des vorigen Kapitels konnten detailliert bei der Implementierung abgebildet werden. Im Folgenden werden die wichtigsten implementierten Ansichten kurz vorgestellt.

### 5.3.1. Startseite und Flugzeugprogrammauswahl

Das Ergebnis für die Startseite (siehe 5.5) konnte vom Entwurf übernommen werden. Für die Anzeige des aktuellen Status der Konfiguration wurde eine Signalfarbe und ein passendes Icon hinzugefügt. Ebenfalls wurde der Text durch passende Icons ausgetauscht (vgl. 4.8).

Bei der Auswahl des Flugzeugprogramms (siehe 5.6), dass direkt nach dem Start einer neuen Konfiguration angezeigt wird, wurde der Begriff Familie statt Programm verwendet. Die Darstellung enthält ein zugehöriges Flugzeug in der passenden Kachel.



Abbildung 5.5.: Auszug der implementierten Startseite



Abbildung 5.6.: Auszug der implementierten Flugzeugprogrammauswahl

### 5.3.2. Upgradeauswahl

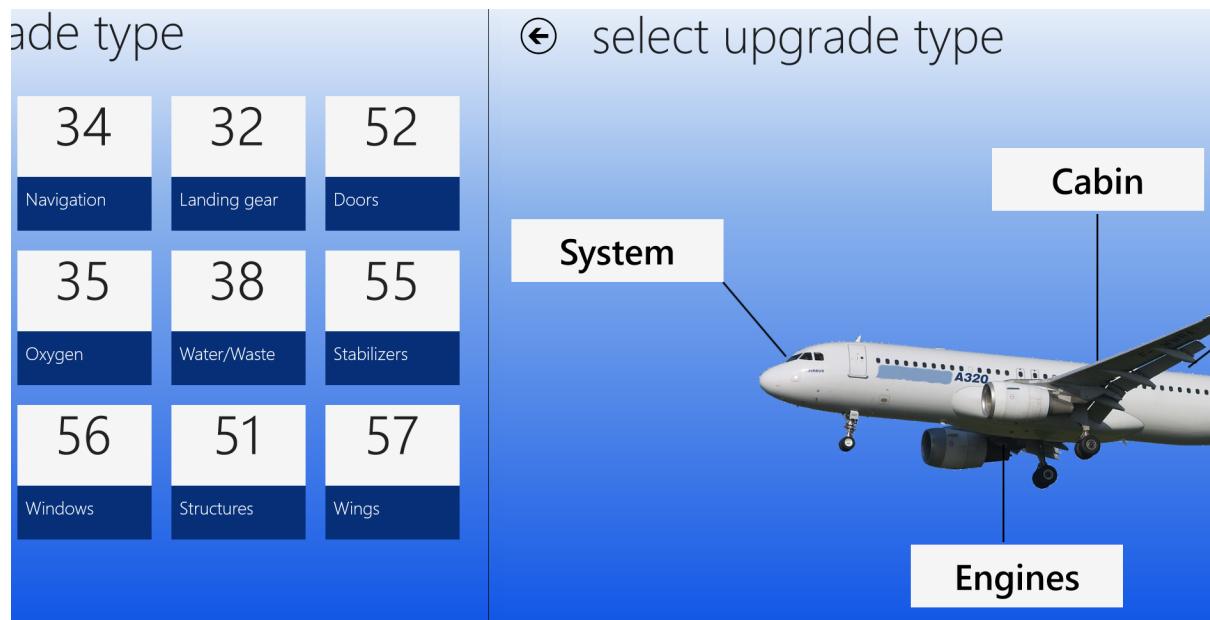


Abbildung 5.7.: Auszug der implementierten Upgradetypauswahl

In Abbildung 5.7 ist die Auswahl des Upgrade Typs zu sehen. Diese wurde mit der FlipView (vgl. 4.12) umgesetzt. Beim Aufruf der Seite ist die Upgradetyp Auswahl (rechtes Bild) zu sehen. Um in den Expertenmodus (linke Seite) zu kommen, wird eine Wisch-Geste nach von links nach rechts durchgeführt. Dieses "flippen" unterstützt den Experten bei einer schnelleren Navigation.

Bei der Auswahl der Upgrades, wie in Abbildung 5.8 zu sehen, sind viele Informationen auf der Ansicht enthalten. Auf der rechten Seite können die einzelnen Unterkategorien ausgewählt werden. Die Details zu der Auswahl erscheinen auf dem rechten Bildschirmrand. Die einzelnen Upgrades werden auf das Wesentliche reduziert. Nach der erfolgten Auswahl wird die derzeitige Selektion auf der unteren AppBar angezeigt. Ein Abschluss der Auswahl sowie die Navigation in die Zusammenfassungsseite erfolgt mit Hilfe eines Button.

## ⌚ 23 Communication

11 Installation of single HF system (SFE)

11 Installation of single HF system and full provision for second one (SFE)

13 Installation of alternate Radio Management Panel (RMP)

50 Installation of additional Audio Control Panel (ACP) and jack panel

**51 Boomsets alternate equipment**

51 Boomsets alternate equipment with two jack plugs

Boomsets alternate equipment

Boomset alternate equipment - ex

DC resistance, soft ear cushions, 70-inch straight cord

Boomset alternate equipment - Sennheiser

DC resistance, soft ear cushions, 70-inch straight cord

Objective  
To provide alternate equipment for the basic boomsets

Basic Aircraft  
two jack panels, one for the connectors, one at each pilot seat

Description  
The basic boomsets are replaced by the new ones and 538B specifications are met

Epac selection mode  
Individual

Current Selection:  1046GT2115

select upgrades

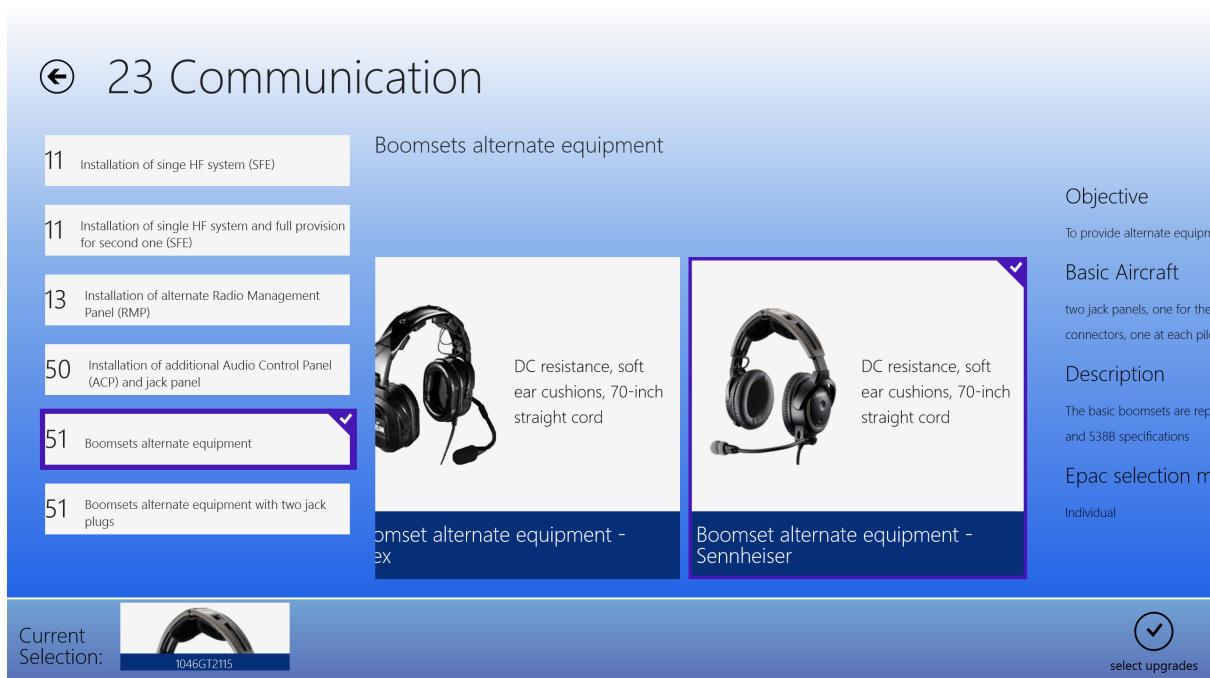


Abbildung 5.8.: Auszug der implementierten Upgradeauswahl

### 5.3.3. Flugzeugauswahl

Die Auswahl der Flugzeuge (siehe 5.9) ist mit den Kacheln im Grid umgesetzt worden. Für eine schnelle Auswahl von großen Datensätzen wurde der semantische Zoom in dieser Ansicht verwendet. Damit dem Benutzer mehr Möglichkeiten gegeben werden, kann er die Gruppierung der Flugzeuge mit einer Auswahlliste selbst festlegen. Abbildung 5.10 zeigt den semantischen Zoom mit den verschiedenen Flugzeugtypen gruppiert. In dieser Ansicht wird die Anzahl der Elemente in der Gruppe angezeigt.



Abbildung 5.9.: Auszug der implementierten Flugzeugauswahl



Abbildung 5.10.: Herausgezoomte Ansicht der Flugzeugauswahl

### 5.3.4. Konfigurationsansicht

Eine Übersicht der Konfigurationsergebnisse wird in der Zusammenfassungsseite (siehe 5.11) gezeigt. Im Screenshot sind zwei Konfigurationsgruppen angezeigt, die

beide komplett sind und somit auch bestellt werden können. Der Abschluss der Konfiguration erfolgt mit den Buttons in der unteren AppBar. Die obere Leiste zeigt die Navigationsmöglichkeiten, die in der kompletten Anwendung vorhanden sind. Auf der rechten Seite können die Hauptansichten ausgewählt werden und auf der linken Seite kann zur Startseite navigiert werden.

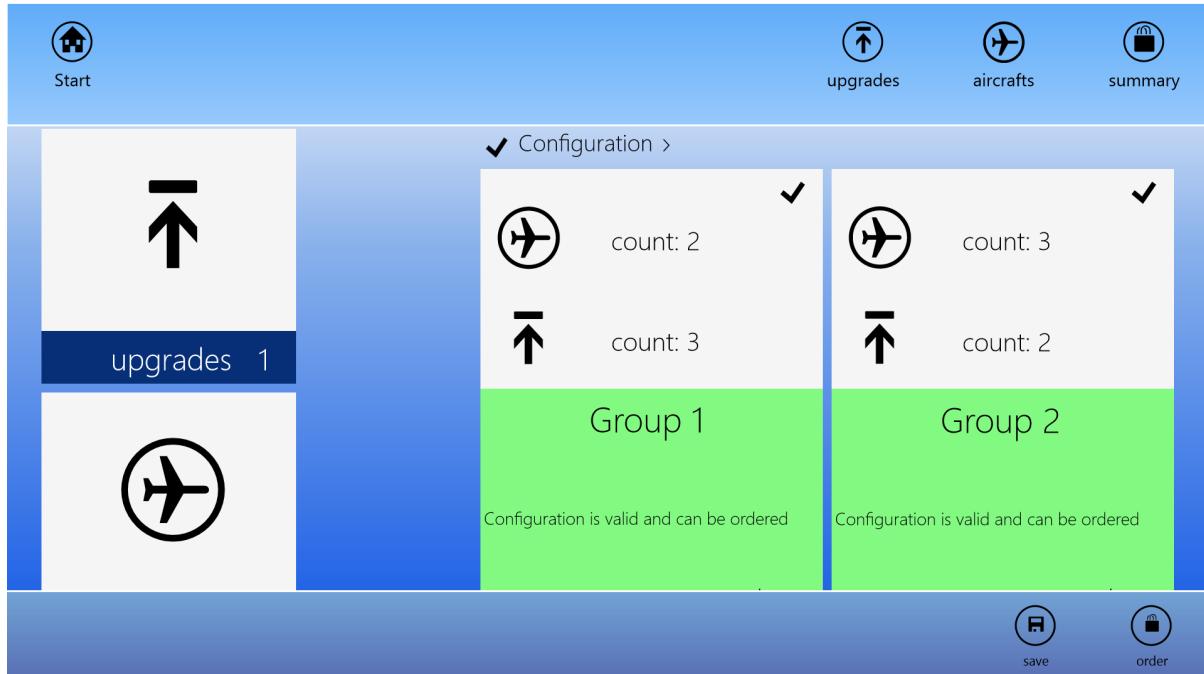


Abbildung 5.11.: Konfigurationsgruppen und Navigationsbar in der Zusammenfassung

Die Auswahl der Alternativen ist in Abbildung 5.12 dargestellt. Hier werden zuerst die möglichen Alternativen angezeigt, die ausgewählt werden können. Zusätzlich sind die zuvor ausgewählten Upgrades vorhanden sowie die zur Konfigurationsgruppe gehörigen Flugzeuge.

## ⌚ Group 2

⌚ alternatives >



CN23.51.136...  
details >

⌚ upgrades >



HF - Antenna, HFDR antenna coupler, HFDR transceiver

Single HF system - Rockwell Collins 900

⌚ aircrafts >



N-2456



N-3065



N-2716

AFR01  
A320

Alternative 1  
Count of new TDUs: 1  
select >

Abbildung 5.12.: Alternativenauswahl für die zweite Konfigurationsgruppe

# **6. Evaluation der Anwendung**

Die Anwendung wurde anhand der spezifizierten Anforderungen in Kapitel 3.3 umgesetzt. Beim Entwurf der Ansichten sind die Heuristiken von Nielsen [Nie95] sowie die Designrichtlinien von Microsoft für die Zielplattform umgesetzt worden. Eine Überprüfung, ob die Ziele, die zuvor gestellt wurden, erreicht sind, wird mithilfe einer Evaluation der Anwendung sichergestellt. Der Aufbau und die Zielsetzung dieser wird im Folgenden beschrieben.

## **6.1. Durchführung der Evaluation**

Bei der Evaluation steht die Erfüllung der gestellten Anforderungen im Vordergrund. Wenn diese erfüllt sind, so kann der modellierte Prozess auch umgesetzt werden. Für die Untersuchung der verschiedenen Anforderungstypen werden unterschiedliche Tests durchgeführt.

### **6.1.1. Funktionale Anforderungen**

Die funktionalen Anforderungen aus Abschnitt 3.3.2 werden mit Anwendern evaluiert. Hierzu wird die alte Weboberfläche, die aktuell für die Konfiguration verwendet wird, als Ausgangspunkt verwendet. Da in der neuen Anwendung ein zusätzlicher Produktkatalog integriert ist, wird kein direkter Vergleich der beiden Lösungen durchgeführt. Stattdessen werden gezielte Aufgaben an die Benutzer gestellt, die die Umsetzbarkeit der Anforderungen überprüft. Die einzelnen Aufgabenstellungen werden anhand einer Skala von 1, sehr gut, bis 5 mangelhaft bewertet. Bei der Bewertung wird die Frage, wie gut oder schlecht die Aufgabe durchgeführt werden kann, als Bewertungsgrundlage verwendet. Die resultierenden Ergebnisse sind dabei die subjektive Meinung des Testers.

Da die aktuelle Konfigurationslösung für Experten entwickelt wurde, wird die Benutzerevaluation in einer Interview Form durchgeführt. Die auftretenden Fragen werden protokolliert, so dass eine zusätzliche Auswertung der Probleme im Anschluss erfolgen kann. Ebenfalls sind die Aufgaben so gestellt, dass sie mit beiden Anwendungen durchgeführt werden können. Da die Weboberfläche nicht die gleichen Funktionen besitzt, wie die App, werden die Anforderungen, die nur für die Tablet-Anwendung definiert wurden, bei der Evaluation nicht berücksichtigt.

### **6.1.2. Nicht-Funktionale Anforderungen**

Eine Überprüfung der nicht-funktionalen Anforderungen ist eine größere Herausforderung, da hier meist subjektive Meinungen entstehen. Aus diesem Grund wurden bei der Spezifikation der Anforderungen die zehn Heuristiken von Nielsen verwendet. Anhand dieser kann eine Auswertung der Anwendung erfolgen. Hierzu werden zwei unterschiedliche Arten von Tests durchgeführt. Die zuvor genannten Benutzertests werden mit zusätzlichen Fragen zur Verwendung der App erweitert. Die Fragestellungen beziehen sich auf eine subjektive Wahrnehmung der nicht-funktionalen Anforderungen. Für eine objektivere Sicht wird eine zusätzliche Expertenevaluation durchgeführt, bei der die zehn Heuristiken von Nielsen bewertet werden sollen. Die Experten kennen sich mit dem System sowie mit Benutzerschnittstellen aus und können so gezielt die Eigenschaften des Systems untersuchen. Bei der Durchführung wird der Experte die Anwendung ohne Vorgaben überprüfen und anschließend eine Bewertung anhand der vorgegebenen Kriterien abgeben.

### **6.1.3. Auswahl der Testmenge**

Damit eine Evaluation sinnvoll ist, muss eine geeignete Anzahl von Testergebnissen vorliegen. Laut Untersuchungen von Nielsen [Ni95] findet eine Anzahl von fünf Personen bereits 75% aller Usability Probleme. Weiterhin hat eine Anzahl von drei Benutzern das beste Verhältnis von Kosten und Nutzen. Aus diesem Grund werden vier Benutzertests und drei Expertentests durchgeführt.

## **6.2. Ergebnisse**

Nach der Durchführung der Evaluation sind drei Ergebnisse vorhanden. Die jeweiligen Fragebögen der Experten und Benutzer sowie die Auswertung der Benutzerfragen. Bei der vorhandenen Konfigurationslösung sind Fragen zum Expertenwissen aufgetreten. Im Gegensatz dazu sind bei der Verwendung der Tablet Anwendung keine fachlichen Fragen aufgetreten. Dies zeigt die unterschiedlichen Zielgruppen der beiden Anwendungen. Eine wiederkehrende Frage ist beim Abschluss der Konfiguration aufgetreten. Hier ist nicht ersichtlich gewesen, wie diese Funktion durchgeführt werden kann. Diese Frage ist bei allen vier Testern aufgekommen. Die weiteren Aufgaben waren klar und konnten ohne Problem durchgeführt werden.

### **6.2.1. Auswertung der Benutzerergebnisse**

Das Ergebnis der Fragebögen in Bezug auf die funktionalen Anforderungen ist in Abbildung 6.1 zu sehen. Auffällig ist die besonders gute Bewertung für die Auswahl einer neuen Konfiguration und der Alternativenauswahl. Diese beiden Ansichten und deren Funktionen konnten während des Tests schnell gefunden, verstanden und verwendet werden. Die drei Hauptfunktionen Flugzeugauswahl, Upgradeauswahl und Darstellung der Konfigurationsergebnisse sind durchschnittlich als Gut bewertet worden. Für eine optimale Verwendung dieser Funktionen hat das Wissen eines Kunden gefehlt, der seine eigenen Flugzeuge kennt sowie eine grobe Zuordnung der Upgrades in die einzelnen Bereiche vornehmen kann.

Das Problem mit dem Abschluss der Konfiguration, welches sich bereits bei den Fragen der Anwendern herausgestellt hat, ist auch in der Bewertung ersichtlich. Drei der vier Anwender haben dem Abschluss der Konfiguration mit befriedigend bewertet. An dieser Stelle sollte eine Verbesserung für die Erfüllung der Anforderungen stattfinden.

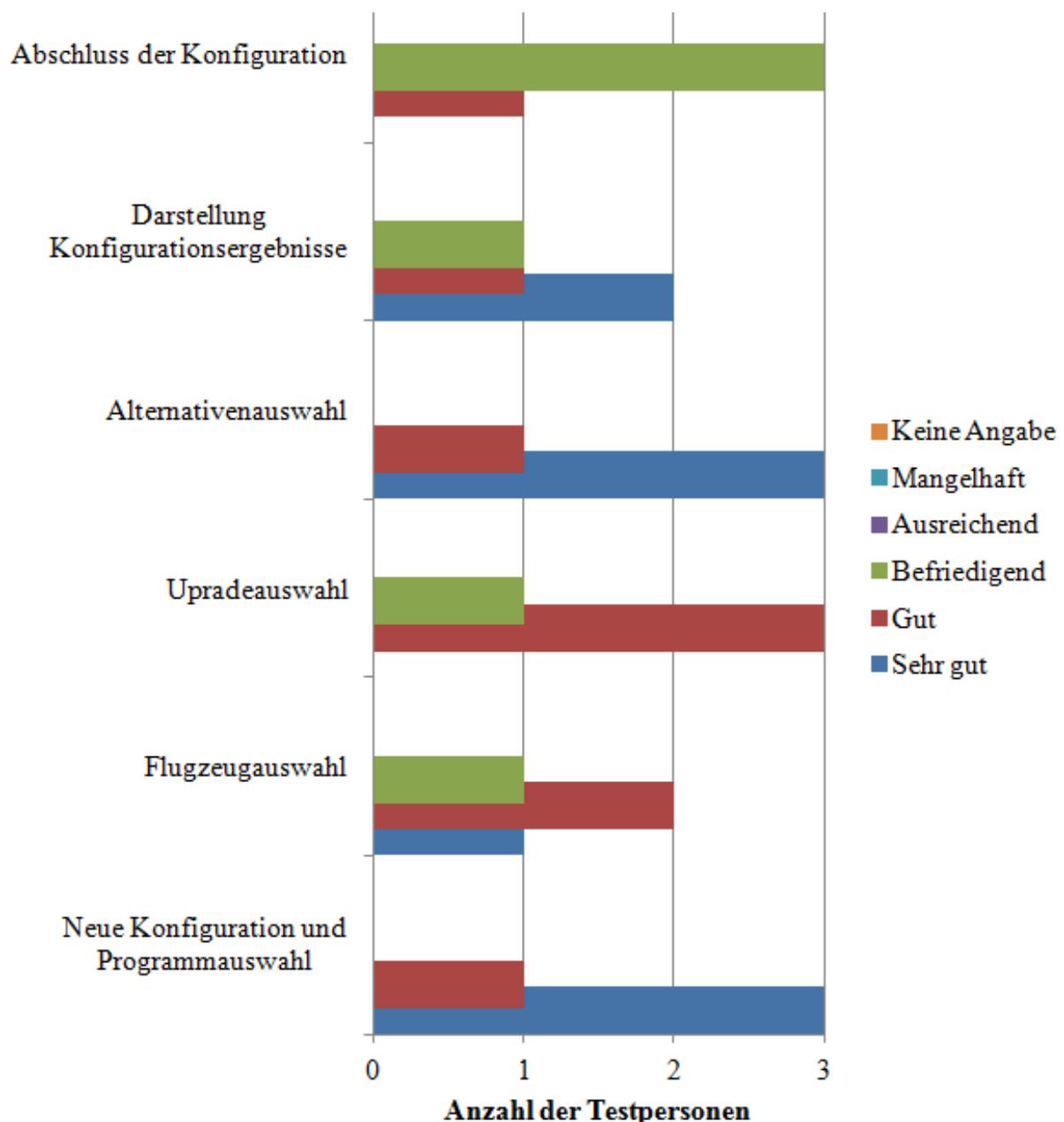


Abbildung 6.1.: Auswertung der Funktionalen Anforderungen im Fragebogen

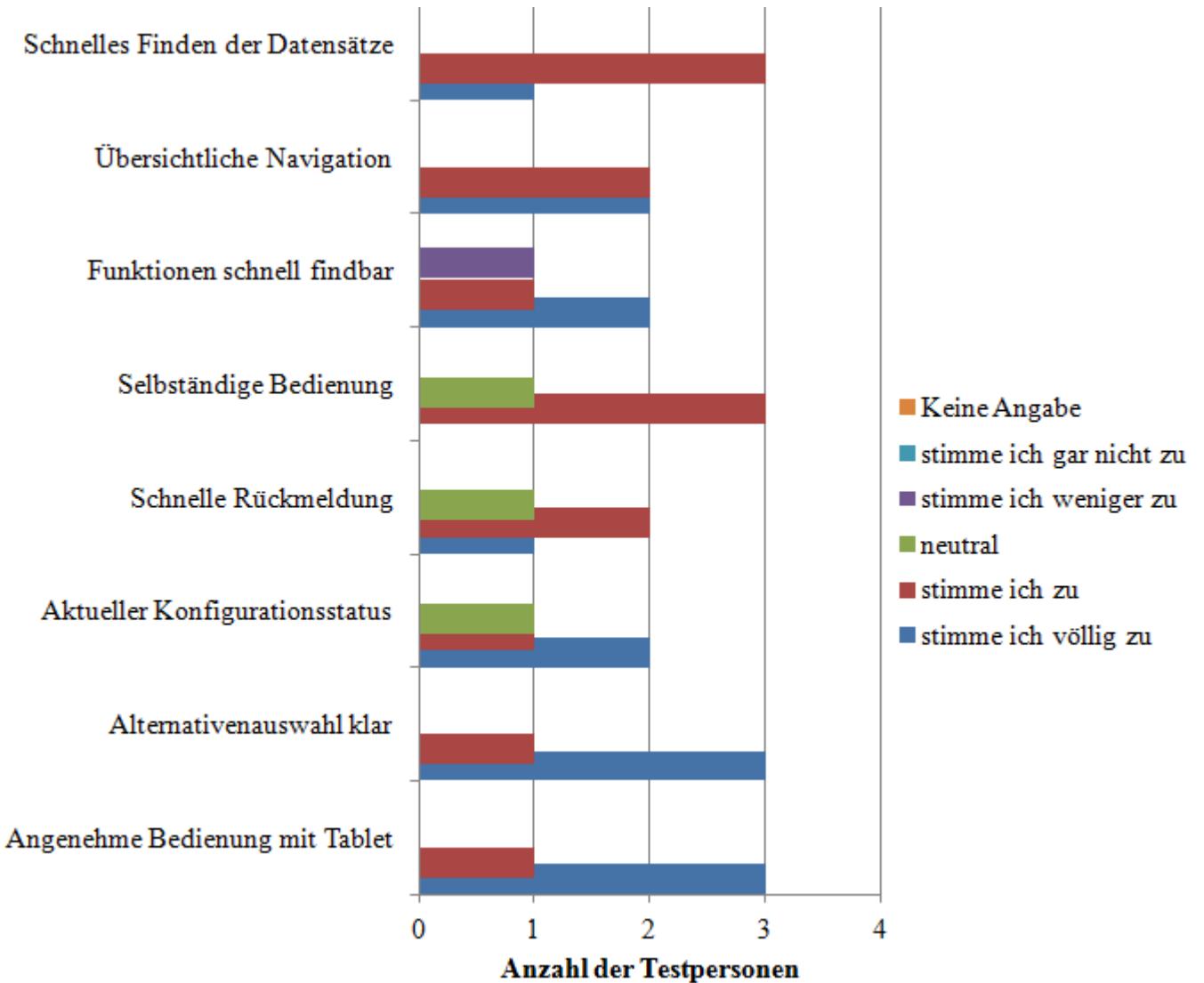


Abbildung 6.2.: Ergebnisse der Usability Fragen zu der Tablet Anwendung

Die Bewertungen des zweiten Teils der Evaluation sind in der Grafik 6.2 zu sehen. Besonders positiv fallen die Ergebnisse für die Bedienung mit dem Tablet sowie die Verständlichkeit der Alternativenauswahl auf. Es gelang damit ein Verständnis für die Auswahl von Alternativen zu schaffen, was dabei helfen sollte den Konfigurationsprozess mit dem Kunden gemeinsam durchzuführen. Diese Tatsache unterstützt die gute Bewertung beim Punkt selbständige Bedienung. Die Maßnahmen für ein flexibles und tolerantes Navigationskonzept sind im Kriterium übersichtliche Navigation sehr gut bewertet worden.

Das Ergebnis der guten Bedienbarkeit auf dem Tablet resultiert auf der ausgewählten Plattform sowie den verwendeten Bedienelementen in der Entwurfsphase. Bei der Implementierung wurde der Entwurf sehr gut umgesetzt, wodurch die gute Bewertung erklärt werden kann.

Bei der ersten Verwendung der Anwendung reagiert die App langsamer als beim zweiten Mal. Der Grund hierfür ist, dass beim initialen Start die Daten erst in den Hauptspeicher geladen werden müssen. Bei der nächsten Verwendung sind diese bereits im Speicher enthalten. Da eine Verzögerung nur bei wenigen Ansichten, wie der Navigation von der Upgradeauswahl in die Zusammenfassung auftritt, ist dieses Problem aufgrund der ansonsten positiven Bewertungen bei dem Kriterium der schnellen Rückmeldung zu vernachlässigen. Zur Lösung dieses Problems sollten die Daten vorgeladen werden und in der Wartezeit ein Ladebalken angezeigt werden. Eine andere Situation liegt bei der Bewertung für die Darstellung des aktuellen Konfigurationsstatus vor. Dieser Punkt ist für die Anwendung essentiell, da der Benutzer zu jeder Zeit wissen muss, in welchem Status er sich befindet. Da auch hier die anderen Bewertungen positiv waren, ist keine eindeutige Ursache des Problems auszumachen.

## **6.2.2. Auswertung der Expertenergebnisse**

Bei den Experten sind die Fragen gezielt nach den nicht-funktionalen Anforderungen gestellt worden. Hierbei wurden die zehn Heuristiken von Nielsen als Grundlage verwendet und jeder dieser Punkte sollte bewertet werden. Die Befragung hat die Stärken und Schwächen der Anwendung aufgezeigt. In Abbildung 6.3 ist das Ergebnis der Auswertung zu sehen. Für die Flexibilität der Nutzung wurde der Expertenmodus (siehe 4.5) entworfen. Das Ziel des Modus wurde, wie die positive Bewertung zeigt, eindeutig erreicht. Eine klare Umsetzung der Designvorgaben ist durch die gute Bewertung von Konsistenz und Standards sowie des ästhetischen und minimalistischen Designs gegeben.

Die Heuristiken Hilfe und Dokumentation sowie Fehlerbehandlung sind weniger gut bewertet worden, da der Fokus der nicht-funktionalen Anforderungen auf anderen Kriterien lag. Diese Punkte sind außerdem weniger relevant, da ein Mitarbeiter des Herstellers bei der Anwendung der App dabei ist. Die Punkte Fehlertoleranz, Benutzerkontrolle und System stimmt mit der Wirklichkeit überein erhalten ebenfalls eine gute Bewertung, obwohl diese nicht im Vordergrund standen. Dies zeigt ein positives Gesamtbild der Anwendung.

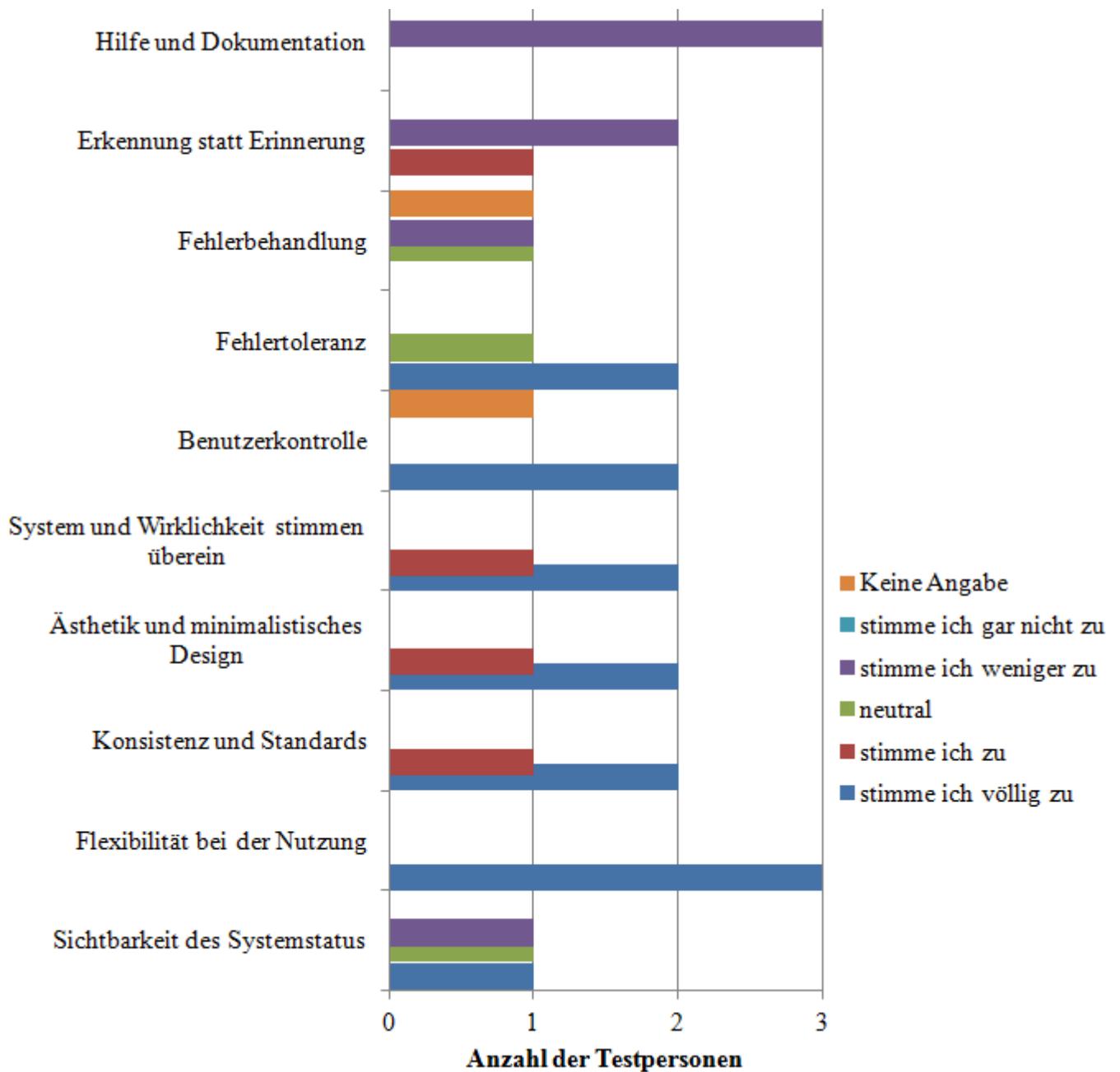


Abbildung 6.3.: Auszug aus der Expertenbefragung

Auch bei den Experten ist die dauerhafte Sichtbarkeit des Systemstatus, der analog zum Konfigurationsstatus aus den Benutzertests zu sehen ist, weniger vorhanden. Der Punkt Erkennung statt Erinnerung ist mit dieser Bewertung verwandt und erhält deshalb eine schlechtere Bewertung. An dieser Stelle fehlt ein Leitfaden in der Anwendung, so dass die folgenden Schritte deutlicher werden.

## 6.3. Konsequenzen aus der Evaluation

Die Ergebnisse der Evaluation haben zwei Probleme der Anwendung aufgeworfen. Zum einen ist der Abschluss des Auswahlprozesses, zum anderen ist der aktuelle Status der Konfiguration bzw. des Systems nicht sichtbar. Beide Problemstellen beziehen sich auf die Seite der Zusammenfassung. Diese ist als Orientierungshilfe für den Anwender konzipiert worden. Aus diesem Grund muss in dieser Ansicht der aktuelle Systemstatus und der Abschluss der Konfiguration aufgenommen werden.

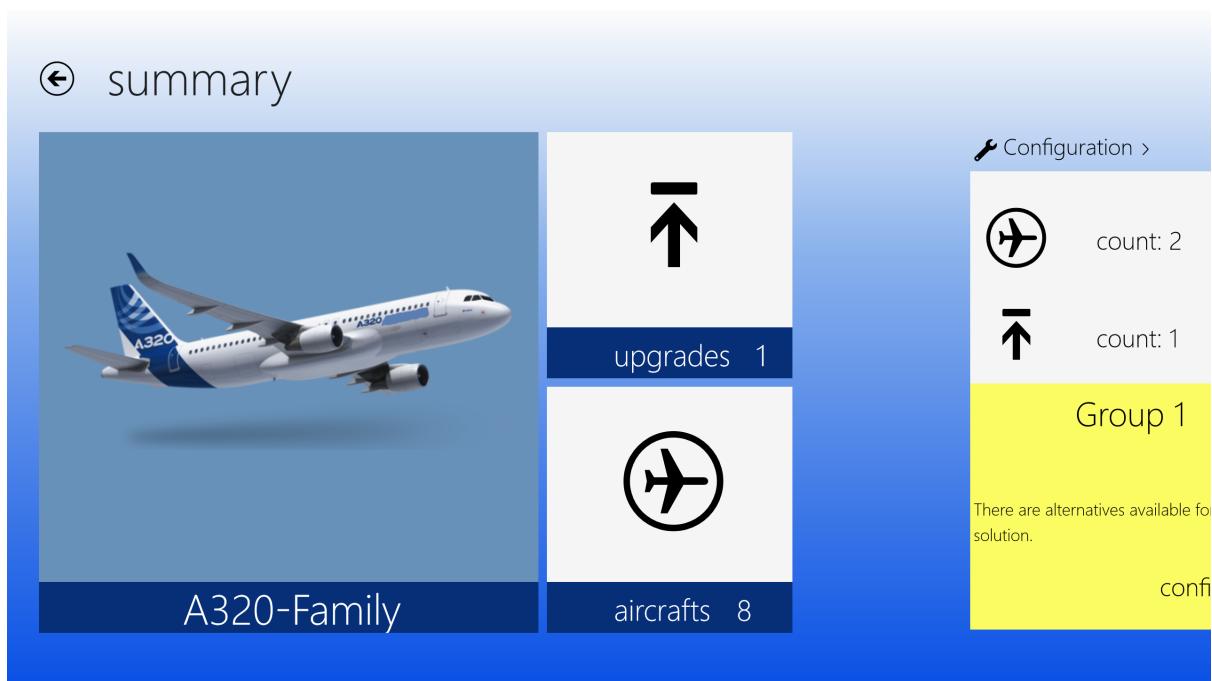


Abbildung 6.4.: Neue Zusammenfassungsansicht, nach dem Redesign

Abbildung 6.4 zeigt den neuen Entwurf der Zusammenfassung. Die Idee ist der Mathematik entnommen, bei der Eingabe A + Eingabe B = Ergebnis C ergibt. Angewendet

auf die Ansicht bedeutet dies, dass zuerst die Eingabedaten Flugzeug und Upgrades angezeigt werden und als Resultat die Konfigurationsgruppen gebildet werden. Weiterhin wird die Navigation zur Flugzeug- und Upgradeauswahl mit einer Kachel durchgeführt. Für eine bessere Übersicht erhalten die Kacheln die Anzahl der ausgewählten Elemente. Dies verhindert ein zu langes Scrollen bei größeren Selektionen. Der Abschluss der Konfiguration wird durch ein dauerhaftes Anzeigen der AppBar verbessert. Hierdurch sind die Buttons für einen Abschluss der Konfiguration dauerhaft sichtbar.

Für eine zukünftige Entwicklung der App sind die weiteren Heuristiken interessant. Diese waren im Evaluationsbogen der Experten, jedoch nicht in der Zielsetzung der Anforderungen enthalten. Aufgrund der unterschiedlichen Bewertung der Experten in diesen Kriterien können hier weitere Verbesserungen durch die Umsetzung der Punkte erzielt werden.

## 7. Fazit und Ausblick

In der vorliegenden Arbeit wurde ein Weg gezeigt, wie ein komplexes Produkt übersichtlich dargestellt werden kann. Nach einer Einführung in die Thematik und die Anwendungsdomäne wurde eine ausgiebige Analyse des Ist-Zustandes eines Konfigurationsprozesses durchgeführt. Die aufgetretenen Probleme sind in einem neuen Prozess behoben worden, der zusätzliche Anpassungen aufgrund der mobilen Zielumgebung enthält. Aus dem neuen Workflow sind die Anforderungen der Anwendung spezifiziert worden. Diese sind für eine Auswahl des passenden mobilen Anwendungstypen, sowie der entsprechenden Technologie verwendet worden. Nach der Auswahl der Plattform sind die einzelnen Prozessschritte für die mobile App konzipiert worden. Die folgende Implementierung wurde mit einem passenden Architektur-Pattern umgesetzt. Eine Evaluation am Ende hat die Erfüllung der Ziele bestätigt und konnte weitere Verbesserungsmöglichkeiten aufzeigen.

Die zugrundeliegenden Komplexität des Produktes wurde durch eine übersichtliche Aufbereitung des Inhalts in einer mobilen, touchgesteuerten Umgebung erreicht. Hierbei wurden besondere Bedienelemente der ausgewählten Technologie für eine einfache und schnelle Bedienung erfolgreich eingesetzt. Der Kunde kann die Anwendung damit genauso bedienen wie ein Mitarbeiter des Herstellers.

Innerhalb der Arbeit ist eine Verbindung zwischen einem Produktkatalog und einer Konfigurationslösung zu einer gemeinsamen Komponente entstanden. Durch die Software aus einer Hand können die Geschäftsprozesse für eine Individualisierung des Produktes schneller durchgeführt werden und der Kunde erhält ein schnelleres Feedback zu seiner Auswahl. Die App ist somit für eine Vereinfachung des Konfigurationsprozesses bereit und kann einen ökonomischen und komfortablen Weg der Konfiguration bieten, wodurch ein deutlicher Mehrwert beim Einsatz entsteht.

Der neue mobile Workflow wurde für ein konkretes Beispiel aus der Luftfahrt umgesetzt. Für die Zukunft ist eine Anwendung des neuen Prozesses auf weitere Geschäftsfelder interessant. Voraussetzung für die Umsetzung ist ein komplexes Produkt

mit vielen Abhängigkeiten der einzelnen Bauteile, die für einen Kunden verständlich dargestellt werden müssen. Aus der Umsetzung auf einem weiteren Gebiet kann der modellierte Prozess weiter verbessert werden, damit die Effizienz bei der Durchführung einer Konfiguration für die Individualisierung eines Produktes weiter steigt.

# A. Anhang

## A.1. Quellcodeauszug

Im der Abbildung A.1 ist das vollständige ViewModel für die Auswahl des Flugzeugprogramms zu sehen. Die Klasse wurde für ein Beispiel der Navigation in der Anwendung verwendet und ist im Text referenziert.

```
/// <summary>
/// This View Model has the logic for aircraft family selection.
/// Normally its the first view if
/// a new configuration is started.
/// </summary>
public class SelectAircraftFamilyViewModel : GridHolderViewModel
{
    private AircraftModel _model;
    private ICommand _familySelectedCommand;

    public SelectAircraftFamilyViewModel()
    {
        Headline = "select aircraft family";
        _model = new AircraftModel();
        InitializeDataSource();
    }

    private void InitializeDataSource()
    {

        DataGroupElements = new ObservableCollection<IIdentifiable>
        {
            new AircraftProgrammGroup(_model.
                GetAllAircraftProgramms())
        };
    }
}
```

```

public ICommand SelectAircraftCommand
{
    get { return _familySelectedCommand ?? (_familySelectedCommand
        = new RelayCommand<DataCommon>(
            SaveSelectionAndNavigateToSummaryPage)); }
    set
    {
        _familySelectedCommand = value;
        OnPropertyChanged();
    }
}

private void SaveSelectionAndNavigateToSummaryPage(DataCommon data)
{
    var selectedProgramm = GetSelectedProgramm(data.UniqueId);
    _model.SelectAircraftProgramm(selectedProgramm);
    var classToNavigate = SimpleIoc.Default.GetInstance<ISummary>()
        ;
    var navigationService = SimpleIoc.Default.GetInstance<
        INavigationService>();
    navigationService.Navigate(classToNavigate.GetType());
}

private AircraftProgramm GetSelectedProgramm(string uniqueId)
{
    return _model.GetAllAircraftProgramms().FirstOrDefault(programm
        => programm.UniqueId.Equals(uniqueId));
}

public override void Initialize(object parameter)
{
}

```

Listing A.1: Vollständige SelectAircraftFamilyViewModel Klasse für die Flugzeugprogrammauswahl

## A.2. Evaluationsbögen

Die Fragebögen für die Experten und die Benutzer sind auf den folgenden beiden Seiten angehängt.

# Evaluationsbogen Experte

Heuristiken	stimme ich völlig zu	stimme ich zu	neutral	stimme ich weniger zu	stimme ich gar nicht zu	Keine Angabe
<b>Sichtbarkeit des Systemstatus:</b> Das System informiert den Benutzer jederzeit über den aktuellen Systemstatus. Dies passiert in einer vertretbaren Zeit.						
<b>System und Wirklichkeit stimmen überein:</b> Das System verwendet die Sprache des Benutzers. Es spricht auf eine natürliche Weise mit dem Benutzer						
<b>Benutzerkontrolle:</b> Bei einem Fehler des Benutzers wird ihm immer ein "Notausgang" geboten.						
<b>Fehlertoleranz:</b> Das System ist fehlerverzeihend und versucht zuerst selbstständig eine Lösung für einen Fehler zu finden						
<b>Konsistenz und Standards:</b> Die Anwendung besitzt ein gleichmäßiges Erscheinungsbild und verwendet die Vorgaben der Plattform						
<b>Ästhetik und minimalistisches Design:</b> Unnötige Informationen werden nicht angezeigt. Der Benutzer wird nicht mit Informationen überhäuft.						
<b>Fehlerbehandlung:</b> Bei einem auftretendem Fehler wird dem Benutzer durch klare Meldungen Hilfe angeboten.						
<b>Erkennung anstatt Erinnerung:</b> Der Benutzer muss sich nicht an Aktionen bei der Ausführung erinnern müssen.						
<b>Hilfe und Dokumentation:</b> Es werden dem Benutzer Hinweise bei der Benutzung gegeben, falls etwas unklar sein sollte.						
<b>Flexibilität bei der Nutzung:</b> Es bestehen unterschiedliche Arten der Nutzung. Ein Power-User kann die Wege schneller bedienen.						

# Evaluationsbogen Benutzer

Webgui Lösung	Sehr Gut	Gut	Befriedigend	Ausreichend	Mangelhaft	Keine Angabe
Beginne eine neue Konfiguration und wähle das Flugzeugprogramm A320						
Wähle die Flugzeuge der Flugzeugversion AFR 11 aus.						
Wähle ein alternatives Boomset als Upgrade aus.						
Wähle eine Alternative aus.						
Darstellung der Konfigurationsergebnisse.						
Abschluss der Konfiguration.						

Tablet- App	Sehr Gut	Gut	Befriedigend	Ausreichend	Mangelhaft	Keine Angabe
Beginne eine neue Konfiguration und wähle das Flugzeugprogramm A320						
Wähle die Flugzeuge der Flugzeugversion AFR 11 aus.						
Wähle ein alternatives Boomset als Upgrade aus.						
Wähle eine Alternative aus.						
Darstellung der Konfigurationsergebnisse.						
Abschluss der Konfiguration.						

Aussagen	stimme ich völlig zu	stimm e ich zu	neutral	stimme ich weniger zu	stimme ich gar nicht zu	Keine Angabe
Ich traue mir zu die Anwendung ohne Hilfestellung zu bedienen.						
Die Navigation der Anwendung ist übersichtlich.						
Ich weiß zu jeder Zeit, welche Konfigurationsschritte noch zu machen sind.						
Bei jeder Aktion bekomme ich eine schnelle Rückmeldung.						
Die Bedienung mit dem Tablet ist angenehm.						
Ich finde die gewünschten Datensätze schnell						

Ich habe die Alternativenauswahl verstanden						
Ich kann alle Funktionen schnell finden.						

Welche Teile der Anwendung fand ich besonders gelungen?

Welche Teile der Anwendung sind noch verbesserungsfähig?

### A.3. Zusätzliche Evaluationsergebnisse

Im Folgenden wird das Ergebnis des ersten Teils vom Benutzerfragebogen dargestellt. In Abbildung A.1 ist die vollständige Auswertung der Fragebögen für die Fragen zur vorhandenen Konfigurationslösung zu sehen. Da beide Anwendungen unterschiedliche Zielgruppen und Ziele verfolgen, wurde kein Vergleich durchgeführt. Die Fragen waren für die Darstellung der Unterschiede, sowie für das Durchführen der gemeinsamen Aktionen hilfreich.

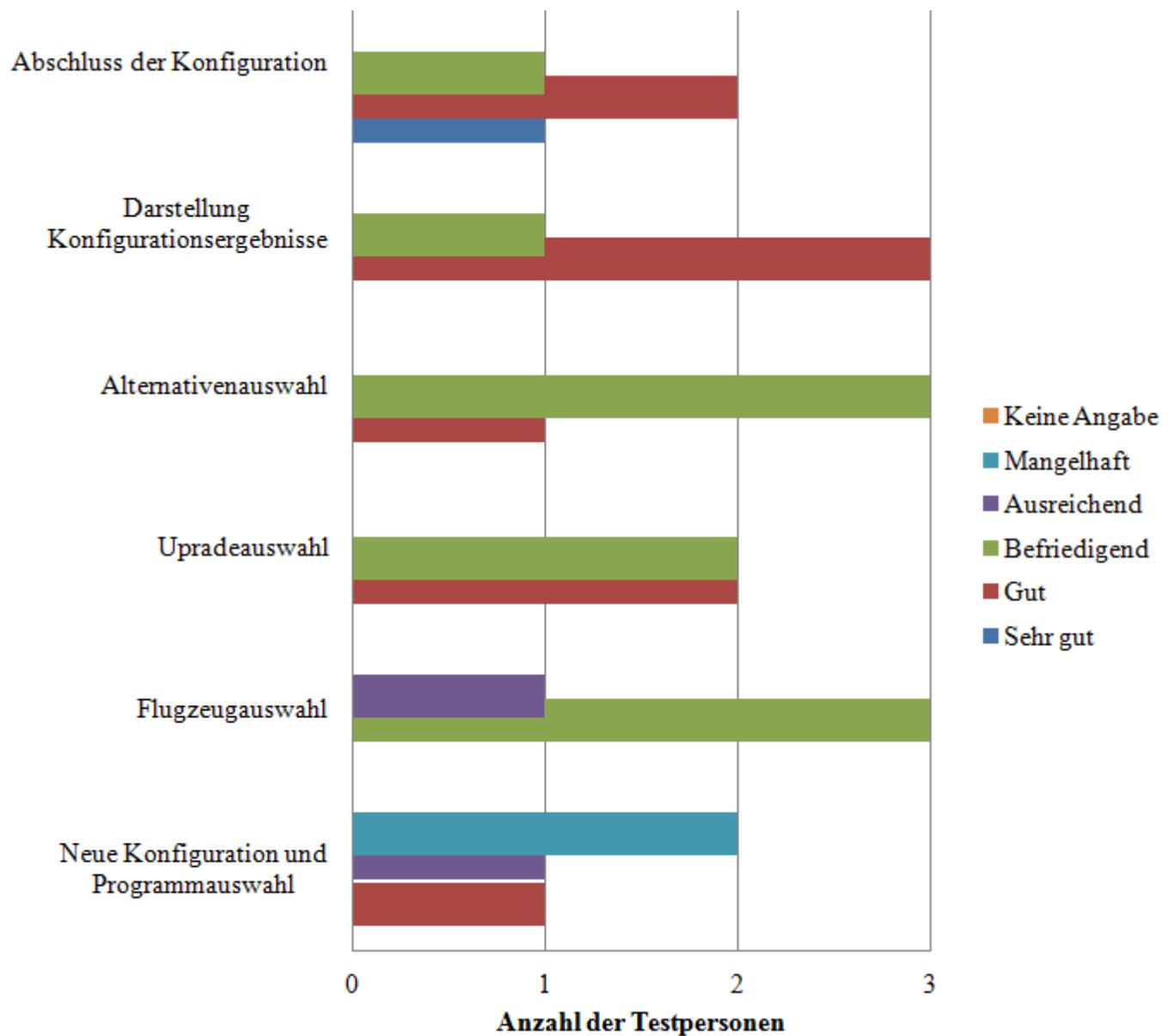


Abbildung A.1.: Ergebnis der Fragen zur vorhandenen Konfigurationslösung

# Abbildungsverzeichnis

2.1.	Aufbau eines Expertensystems (Quelle: [Kel07, s.6]) . . . . .	6
2.2.	Architektur und Datenfluss der Merlin Komponenten . . . . .	8
3.1.	Auszug eines Konfigurationsprozesses . . . . .	13
3.2.	Programmablauf des Anwendungsbeispiels . . . . .	16
3.3.	Workflow der Konfigurator-App . . . . .	19
3.4.	Anwendungsbereiche der Tablet Anwendung . . . . .	20
4.1.	Grid mit eingezeichneten Kacheln (Quelle: [Win8-5], Microsoft) . . . . .	32
4.2.	Semantischer Zoom in zwei Ansichten (Quelle: [Win8-6], Microsoft) . . . . .	35
4.3.	Entwurf der Zusammenfassung Seite . . . . .	36
4.4.	Entwurf der Produktkategorie Auswahl . . . . .	38
4.5.	Entwurf der Produktdetail Ansicht . . . . .	39
4.6.	Entwurf der Flugzeugauswahl . . . . .	40
4.7.	Darstellung der Konfigurationsergebnisse . . . . .	41
4.8.	Entwurf der Startseite . . . . .	42
4.9.	Navigation in der App . . . . .	44
4.10.	Obere AppBar für die Navigation . . . . .	45
4.11.	Untere AppBar für die Navigation . . . . .	45
4.12.	Flip View in der Upgrade Bereich Auswahl . . . . .	47
4.13.	Dropdown List bei der Flugzeugauswahl . . . . .	47
5.1.	Komponenten und Kontrollfluss im MVVM Entwurfsmuster . . . . .	49
5.2.	Klassendiagramm im MVVM Entwurfsmuster . . . . .	51
5.3.	Navigation mit Inversion of Control . . . . .	53
5.4.	Auszug des AircraftFamilyViewModels (siehe Anhang A) . . . . .	54
5.5.	Auszug der implementierten Startseite . . . . .	55
5.6.	Auszug der implementierten Flugzeugprogrammauswahl . . . . .	55
5.7.	Auszug der implementierten Upgradetypauswahl . . . . .	56

5.8. Auszug der implementierten Upgradeauswahl . . . . .	57
5.9. Auszug der implementierten Flugzeugauswahl . . . . .	58
5.10. Herausgezoomte Ansicht der Flugzeugauswahl . . . . .	58
5.11. Konfigurationsgruppen und Navigationsbar in der Zusammenfassung .	59
5.12. Alternativenauswahl für die zweite Konfigurationsgruppe . . . . .	60
6.1. Auswertung der Funktionalen Anforderungen im Fragebogen . . . . .	64
6.2. Ergebnisse der Usability Fragen zu der Tablet Anwendung . . . . .	65
6.3. Auszug aus der Expertenbefragung . . . . .	67
6.4. Neue Zusammenfassungsansicht, nach dem Redesign . . . . .	68
A.1. Ergebnis der Fragen zur vorhandenen Konfigurationslösung . . . . .	77

# **Tabellenverzeichnis**

3.1. Nicht-Funktionale Anforderungen mit Bezeichner . . . . .	22
3.2. Funktionale Anforderungen mit Bezeichner . . . . .	25
4.1. Gegenüberstellung und Bewertung der Zielkriterien für die Anwen- dungsform . . . . .	30
4.2. Gegenüberstellung und Bewertung der Zielkriterien für Plattform . . . .	31

# Literaturverzeichnis

- [Pin93] B. JOSEPH PINE **Harvard Business Press** Mass customization, 1.Auflage , 1993
- [VL11] STEPHAN VERCLAS, CLAUDIA LINNHOFF-POPIEN **Springerverlag** Smart Mobile Apps: Mit Business-Apps ins Zeitalter mobiler Geschäftsprozesse, 1.Auflage , Dezember 2011
- [Kra12] CHRISTIAN KRAUS **2kit consulting** Grundlagen und Erfolgsfaktoren für Apps im Mobile Business, Whitepaper , 16.05.2012
- [Wi13] **Wirtschaftslexikon** Produkt, <http://www.wirtschaftslexikon24.com/d/produkt/produkt.htm>, aufgerufen am 25.07.2013
- [HLW06] HELMUT HEROLD, BRUNO LURZ, JÜRGEN WOHLRAB **Pearson Studium** Grundlagen der Informatik, München 2006
- [TT09] GERALD TESCHL, SUSANNE TESCHL **examen.press** Mathematik für Informatiker, Pearson Studium, Wien 2009, 3. Auflage
- [Pup88] FRANK PUPPE **Springerverlag** Einführung in Expertensysteme, 1. Auflag, 1988
- [Kel07] HUBERT B. KELLER **DH Karlsruhe** Wissensbasierte Systeme - Einführung, Vorlesung SS 2007
- [Rei13] GERALD REIF **Deutsches Forschungszentrum für Künstliche Intelligenz** Expertensysteme, <http://www.dfki.uni-kl.de/aabecker/Mosbach/Experten/Reif-node8.html>, aufgerufen am 18.07.2013
- [Hau13] HAUFE Tablets setzen sich durch, [http://www.haufe.de/marketing-vertrieb/vertrieb/vertrieb-tablets-setzen-sich-durch\\_130\\_155918.html](http://www.haufe.de/marketing-vertrieb/vertrieb/vertrieb-tablets-setzen-sich-durch_130_155918.html), aufgerufen am 26.07.2013
- [BDB08] HANS H. BAUER, THORSTEN DIRKS, MELCHIOR D. BRYANT **examen.press** Erfolgsfaktoren des mobile Marketing, Springer, 2008

- [PAC11] **Pierre Audoin Consultants** Tablets im Unternehmenseinsatz, [http://www.berlecon.de/studien/downloads/PAC\\_Trendstudie\\_Microsoft\\_Mobility\\_May\\_11.pdf](http://www.berlecon.de/studien/downloads/PAC_Trendstudie_Microsoft_Mobility_May_11.pdf), Mai 2011
- [Ebe11] RALF EBERT Eclipse RCP, [http://www.ralfebert.de/eclipse\\_rcp/EclipseRCP.pdf](http://www.ralfebert.de/eclipse_rcp/EclipseRCP.pdf), Version 1.1, 19.08.2011
- [Rud06] CHRISTIANE RUDLOF **Unfallkasse Post und Telekom** Handbuch Software-Ergonomie. Usability Engineering., <http://www.ukpt.de/pages/dateien/software-ergonomie.pdf>, S.52, 2. Auflage, Tübingen 2006
- [Clo13] CLOUDSHERPAS Native, Hybrid and Mobile Web Apps, <http://www.cloudsherpas.com/services/custom-development/mobile-apps/native-hybrid-and-mobile-web-application-development/>, aufgerufen am 23.07.2013
- [Sma13] SMARTDIGITS Native App oder Webb App, <http://www.smardigits.com/2012/02/native-app-oder-web-app-teil-1-definitionen-und-entscheidungskriterien/>, aufgerufen am 23.07.2013
- [Bun06] BUNDESAMT FÜR SICHERHEIT IN DER INFORMATIONSTECHNIK **SecureNet** Sicherheit von Webanwendungen, [https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Publikationen/Studien/WebSec/WebSec\\_pdf.pdf?blob=publicationFile](https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Publikationen/Studien/WebSec/WebSec_pdf.pdf?blob=publicationFile), 1. Version, August 2006
- [Sch13] ANDREAS SCHAFFRY **CIO** Die Zukunft mobiler Anwendungen, <http://www.cio.de/strategien/2910477/index.html>, aufgerufen am 27.07.2013
- [Nie95] JAKOB NIELSEN **Norman Group** 10 Usability Heuristics for User Interface Design, <http://www.nngroup.com/articles/ten-usability-heuristics/>, 1. Januar 1995, aufgerufen am 30.07.2013
- [Stu13] **Mobile-Studien.de** Tablet-Markt 2013: Wer hat den besseren Durchblick, <http://mobile-studien.de/2013/05/tablet-markt-2013-wer-hat-den-besseren-durchblick/>, aufgerufen am 01.08.2013
- [Stu13-2] **Mobile-Studien.de** Marktanteile mobile Betriebssysteme Q1 2013, <http://mobile-studien.de/marktanteile-betriebssysteme/marktanteile-mobiler-betriebssysteme-q1-2013/>, aufgerufen am 02.08.2013

- [Win8] MICROSOFT **MSDN** Windows 8 User experience Guidelines, [http://download.microsoft.com/download/C/0/A/C0AEF0CC-B969-406D-989A-4CDAFDDB3F3C/Win8\\_UXG\\_GA.pdf](http://download.microsoft.com/download/C/0/A/C0AEF0CC-B969-406D-989A-4CDAFDDB3F3C/Win8_UXG_GA.pdf), aufgerufen am 04.06.2013
- [Win8-1] MICROSOFT **MSDN** Index of UX guidelines for Windows Store apps, <http://msdn.microsoft.com/en-us/library/windows/apps/hh465424.aspx>, aufgerufen am 04.06.2013
- [Win8-2] **Ratio** Interpreting the Design Language, <http://www.windows8designhandbook.com/interpreting-the-windows-8-design-language.html>, aufgerufen am 04.06.2013
- [Win8-3] PETER KIRCHNER **MSDN** Developer Week (DWX) - Windows 8 App-Grundlagen, <http://blogs.msdn.com/b/pkirchner/archive/2013/06/26/developer-week-dwx-windows-8-app-grundlagen.aspx>, aufgerufen am 02.08.2013
- [Win8-4] MICROSOFT **MSDN** Planen ihrer App, <http://msdn.microsoft.com/de-de/library/windows/apps/hh465427>, aufgerufen am 16.05.2013
- [Win8-5] MICROSOFT **MSDN** Laying out an app page, <http://msdn.microsoft.com/en-US/library/windows/apps/hh872191>, aufgerufen am 13.08.2013
- [Win8-6] MICROSOFT **MSDN** Guidelines for Semantic Zoom, <http://msdn.microsoft.com/en-us/library/windows/apps/hh465319.aspx>, aufgerufen am 13.08.2013
- [Smi10] JOSH SMITH **Josh Smith Advanced MVVM**, <http://files.cnblogs.com/kingmoon/Advanced.MVVM.pdf>, 1. Auflage, 15.02.2010
- [Bug13] LAURENT BUGNION **msdn magazine** Verwenden des MVVM-Musters in Windows 8, <http://msdn.microsoft.com/de-de/magazine/jj651572.aspx>, aufgerufen am 07.08.2013
- [Fow04] MARTIN FOWLER **martinfowler.com** Inversion of Control Containers and the Dependency Injection pattern, <http://www.martinfowler.com/articles/injection.html>, aufgerufen am 08.08.2013
- [Ni95] JAKOB NIELSEN **Nielsen Norman Group** How to Conduct a Heuristic Evaluation, <http://www.nngroup.com/articles/how-to-conduct-a-heuristic-evaluation/>, 1. Januar 1995, aufgerufen am 10.08.2013