



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

27.03.2013

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, 27.03.2013

Dane Leube

abstract

Das Voranschreiten der sogenannten mass-customization bei Produkten erfordert immer komplexere Produkte, um ein hohes Maß an Individualität erreichen zu können. Für den Kunden im Mittelpunkt steht die Auswahl der einzelnen Komponenten des Produktes. Die Komplexität der Produkte soll für den Kunden nicht sichtbar sein. Es muss somit ein Weg gefunden werden, wie eine komplexe Produktlandschaft für den Kunden vereinfacht dargestellt werden kann. Im Rahmen dieser Bachelorthesis wird mit der Umsetzung eines Produktkataloges auf eine mobile Zielumgebung versucht dieses Ziel zu erreichen. Hierbei werden die wichtigsten Bedürfnisse des Kunden analysiert und darauf aufbauend eine Anwendung konzipiert. Durch die optimierte Darstellung der einzelnen Produkte, sowie eine Überprüfung der Zusammenstellung im Hintergrund wird ein Mehrwert für den Kunden erzielt. Die Sicherstellung der Zielerreichung wurde durch eine Evaluation der Lösung erreicht.

Inhaltsverzeichnis

1	Einleitung	1
1.1	Motivation	1
1.2	Ziel der Arbeit	2
1.3	Vorgehensweise	2
2	Grundlagen	3
2.1	Produkt	3
2.1.1	Produktkatalog	4
2.1.2	Boolesche Algebra in der Produktmodellierung	4
2.2	Produktkonfiguratoren	5
2.2.1	CAS Configurator Merlin Enterprise	7
2.2.2	Anwendungsbeispiel der Arbeit	8
2.3	Mobile Anwendungen	8
2.3.1	Native Anwendungen	9
2.3.2	Web Anwendungen	10
2.3.3	Hybride Anwendung	11
3	Analyse der Anwendung	12
3.1	Aktueller Konfigurationsprozess	12
3.1.1	Übertragung auf den Kunden	14
3.2	Workflow Modellierung	15
3.2.1	Mobiler Konfigurationsprozess	16
3.2.2	Workflow des Anwendungsbeispiels	16
3.3	Anforderungsanalyse	18
3.3.1	Nicht-Funktionale Anforderungen	18
3.3.2	Funktionale Anforderungen	20
4	Entwurf der Benutzerschnittstelle	23
4.1	Auswahl der Anwendungsplattform	23
4.1.1	Anwendungsform	23
4.1.2	Anwendungstechnologie	27
4.2	Untersuchung der Plattform Windows 8	28
4.2.1	Design Richtlinien	28

4.2.2	Bedienkonzepte	29
4.2.3	Alleinstellungsmerkmal	30
4.3	Design der Ansichten	30
4.4	Interaktion der Ansichten	30
5	Implementierung	31
5.1	MVVM	31
5.2	Anwendungsarchitektur	31
5.3	Implementierung des Katalog-Workflows	31
5.4	Implementierung des Konfigurations-Workflows	31
6	Evaluation der Anwendung	32
6.1	Durchführung der Evaluation	32
6.2	Ergebnisse	33
6.2.1	Auswertung der Interviewfragen	33
6.2.2	Auswertung der Fragebögen	33
7	Fazit und Ausblick	34
	Abbildungsverzeichnis	i
	Literaturverzeichnis	iii

1 Einleitung

1.1 Motivation

Die Entwicklung von einer Massen-Produktion zu einer Massen-Individualisierung (engl. mass-customization) bei Produkten schreitet immer weiter voran.[MASS]. Mit der höheren Produktvielfalt können auch individuelle Kundenwünsche bedient werden. Bedingt durch die hohe Komplexität, die durch diesen Trend notwendig ist, wird eine Zusammenstellung des Produktes aufwändiger. Die Durchführung einer solchen Produkt-Individualisierung erfolgt nach einem gegebenen Workflow. Nach einer erfolgten Produktauswahl aus einem Produktkatalog in Papierform, wird die Zusammenstellung manuell geprüft, sodass der Kunde ein Feedback über die technische Realisierung erhält. Dieser Vorgang kostet viel Zeit, Geld und Kapazitäten innerhalb eines Unternehmens.

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. Der Vertrieb eines Produktes sollte möglichst nahe beim Kunden durchgeführt werden. Aus diesem Grund müssen Produktkataloge eine hohe Verfügbarkeit, sowie eine einfache Handhabung aufweisen. Deshalb sind Produktkataloge optimaler Weise mobil verfügbar. Die Mobilität wird durch das Verwenden sogenannter Apps auf mobilen Endgeräten erreicht. Der Nutzen solcher Apps im Geschäftsumfeld wird immer mehr erkannt [APP]. Der Anwender muss bei einer mobilen Anwendung im Zentrum stehen [DESIGN]. Dieses nutzerzentrierte Design wird durch eine hohe Gebrauchstauglichkeit, sowie einen hohen Nutzen für den Anwender erreicht. Aus diesem Grund muss die Anwendung ein durchdachtes Konzept bei der Bedienung beinhalten.

1.2 Ziel der Arbeit

Die Arbeit soll eine Möglichkeit aufzeigen, wie eine komplexe Produktlandschaft für den Kunden übersichtlich dargestellt werden kann. Hierzu sollen komplexe Abhängigkeiten der einzelnen Produkte im Hintergrund von einem Produktkonfigurator berechnet werden. Die Ergebnisse werden dem Benutzer auf eine einfache, verständliche Weise dargestellt. Durch die Umsetzung der Anwendung sollen die Prozesse der Produktkonfiguration auf das Wesentliche, die Produkte, konzentriert werden. Hierzu müssen Konzepte entwickelt werden, die den Benutzer in den Vordergrund stellt, um dessen Bedürfnisse am Besten gerecht zu werden. Für eine Bessere Integration dieses neuen Ansatzes wird der vorhandene Workflow ebenfalls überarbeitet und an die neue Zielsetzung angepasst.

Damit die resultierende Anwendung einen deutlichen Mehrwert erzielt, muss eine hohe Usability erreicht werden. Diese wird durch ein intuitives und damit einfach zu erlernendes Bedienkonzept erreicht. Als weitere Maßnahme muss eine geeignete Form der Anwendung gewählt werden, die allen Anforderungen entspricht. Hierzu soll eine ausgiebige Analyse der aktuellen Möglichkeiten durchgeführt werden.

1.3 Vorgehensweise

Ausgangspunkt der Arbeit ist eine ausgiebige Analyse des Ist-Zustandes eines Kundenprozesses beim Konfigurieren. Hierauf aufbauend werden die Anforderungen der Anwendung spezifiziert. Die Auswahl einer geeigneten mobilen Plattform erfolgt im nächsten Schritt. Diese werden anhand der spezifizierten Anforderungen gewählt. In Folge der Entscheidung über die Plattform folgt der Entwurf der Ansichten. Die entworfenen Elemente werden im Folgenden bei der Implementierung umgesetzt. Am Ende der Arbeit wird für die Sicherstellung der zuvor gestellten Ziele eine Evaluation der Arbeit.

Abriss: Kapitel 2 werden die Grundlagen der Arbeit behandelt. 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 und genauere Definition wird das Produkt zu Beginn beschrieben. Darauf aufbauend wird der Einsatz von Produktkonfiguratoren in diesem Segment behandelt. Mobile Anwendungen stellen die dritte Grundlage für diese Arbeit.

2.1 Produkt

Im Marketing wird ein Produkt als Ergebnis im Produktionsprozess definiert. Innerhalb des Prozesses entsteht das Produkt, welches am Ende eine Summe mehrerer materieller oder immaterieller Eigenschaften besitzt [PRODUCT]. 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 das einzelne Produkt. Sobald der Hersteller mehr als eine Variante einer Einzelkomponente für den Kunden zur Verfügung stellt, lässt sich ein Produkt individualisieren. Dies wird wie eingangs schon erwähnt als mass-customization bezeichnet.

Voraussetzung für diesen Trend ist eine veränderte Wahrnehmung des Kunden. Das Produkt darf nicht mehr als einzelnes Objekt gesehen werden. Für die individuellen Anpassungen muss der Kunde das Produkt als eine Zusammenstellung mehrerer Komponenten verstehen. Diese veränderte Wahrnehmung gilt es dem Kunden zu vermitteln und 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, müssen geeignete Formen der Darstellung gefunden werden.

2.1.1 Produktkatalog

Um dem Kunden einen Einblick in das Produkt zu verschaffen werden sogenannte Produktkataloge verwendet. Diese Kataloge sind meist in Papierform vorhanden und enthalten für den Kunden relevante Informationen über das 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 das 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 [BOOL1]. In der Aussagenlogik wird dies so verwendet, dass eine Aussage, wie "Heute regnet es" entweder wahr oder falsch sein kann [BOOL2]. Mithilfe von verschiedenen Operatoren lassen sich die Aussagen miteinander Verknüpfen, so dass auch komplexere Zusammenhänge möglich sind. Grundlegend zu nennen sind hier die Disjunktion, bei der einer von zwei Aussagen wahr sein muss, um den kompletten Ausdruck wahr werden zu lassen. Bei der Konjunktion müssen beide Aussagen zutreffend sein. Um Schlussfolgerungen durchführen zu

können ist die sogenannte Wenn-Dann Verknüpfung wichtig. Diese besagt, dass eine Aussage wahr ist, sobald eine Andere erfüllt ist.

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 Modellierung der Beziehung von Motor und Getriebe in folgender Form möglich:

$$\text{Verwendung von Motor A} \Rightarrow \text{Einbau von Getriebe A}$$

Bedeutung: Wenn der Motor A verwendet werden soll, dann muss das Getriebe A eingebaut werden, damit die Aussage (Verwendung des Motors A) wahr ist.

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:

$$\text{Motor A} \wedge \text{Getriebe A} \wedge (\text{Sitz A} \vee \text{Armaturenbrett B})$$

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

Die Definition, welche Probleme bei einem komplexen Produkt auftreten können wurde im vorigen Abschnitt geklärt. Das Problem, wie eine große Anzahl der booleschen Regeln, die für die Produktmodellierung benötigt werden verarbeitet werden können

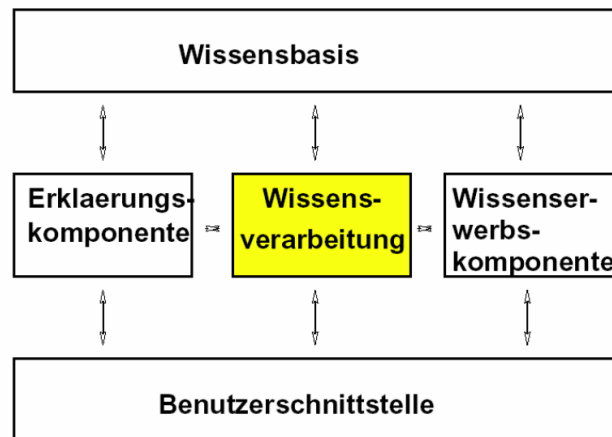


Abbildung 2.1: Aufbau eines Expertensystems [KELLER, s.6]

bleibt bestehen. Eine Lösungsmöglichkeit bieten sogenannte Produktkonfiguratoren. Das Ziel des Produktkonfigurators ist es, produktspezifisches Wissen für die Anwender bereit zu stellen, welches zuvor von Experten in das System eingepflegt wurde. Dieses hilft beim individuellen Zusammenstellung des Produktes durch die Verwendung des Produktwissens. Ein solches System wird in die Kategorie der Expertensysteme[PUPPE] oder wissensbasierte Systeme eingeordnet. Der Aufbau eines solches System ist in Abbildung 2.1 zu sehen.

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[EXPERT]:

- **generisch:** unabhängig vom aktuellen Anwendungsfall. Meist in einfachen Wenn-Dann-Regeln oder auf einem Modell beruhend.
- **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 *Erklärungs-komponente* unterstützt das Nachvollziehen des Ergebnisses durch Erläuterungen zu den getätigten Entscheidungen.



Abbildung 2.2: Architektur der Merlin Enterprise Komponenten

2.2.1 CAS Configurator Merlin Enterprise

Das Produkt CAS Configurator Merlin Enterprise ist die Konfigurationslösung der CAS Software AG für große 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 Configurators 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 Configurator 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.

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[RCP] basiert.

2.2.2 Anwendungsbeispiel der Arbeit

Damit die Arbeit anhand eines geeigneten Beispiels durchgeführt werden kann, wird der vorhandene Produktkonfigurator des Kunden Airbus verwendet¹. 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 Flugzeuges eingesetzt. Ein Beispiel für ein Upgrade in diesem Bereich können diverse Systemupgrades, wie bspw. ein neues Navigationssystem sein.

Dieses Anwendungsfeld ist besonders herausfordernd, 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 ausreichend 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 wird. Mit dieser Form der Anwendung sind neue Anwendungsgebiete in der Software möglich. Ein neues Einsatzgebiet ist der mobile Einsatz der App bei einem Kunden vor Ort [MOBILE1]. Hier können insbesondere Tablet-PCs die Kommunikation mit dem Kunden fördern [TABLET]. Die Vorteile durch den großen Bildschirm und

¹ <http://www.airbus.com/>

die Möglichkeit wie bei einem Blatt Papier den Kunden ins Verkaufsgespräch mit einzubeziehen sind hier überzeugend. Eine Bedienung der Geräte durch Touch-Eingaben ermöglicht eine bessere Interaktion mit der Software. Die Möglichkeiten beim Einsatz dieser Geräte im Geschäftsumfeld ist noch nicht ausgeschöpft und birgt auch weiterhin Potenziale [MOBILE2, Fazit]. In diesem Markt gilt es sich durch gute Anwendungskonzepte zu etablieren.

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 leistungstärkeren 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 einer großen 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.

2.3.1 Native Anwendungen

Native Anwendungen sind auf die jeweilige Zielplattform beschränkt [PLATTFORM]. Die Anwendung kann nur auf dem gewählten Betriebssystem ausgeführt werden. Die Programme für native Apps werden 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 [PLATTFORM2]. 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. Diese Seiten wurden mit zusätzlichen Funktionen, die ein mobiles Gerät zur Verfügung hat erweitert und agiert 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 (Google Chrome ca. 100, Firefox 93 und Safari 90 Stand: 4.4.2013, Quelle: [WEBAPP2]).

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 nativen und einer web App vereint. Dieses Ziel wird dadurch erreicht, in dem der Kern eine Web Anwendung ist, die durch das Einbinden in einen sogenannten nativen Container verwendet wird. 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 auch mehr Funktionen nativ implementieren. 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 werden anschließend spezifiziert.

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. Zu Beginn möchte der Kunde ein konkretes Produkt auswählen. Dies geschieht über den Katalog. Anschließend kann er mithilfe von eindeutigen Produktnummern die Bestellung über ein Formular weitergeben. Damit die Zusammenstellung des Kunden geprüft werden kann, werden die Daten beim Hersteller in das Konfigurationssystem eingepflegt. Dieses berechnet die komplexen Abhängigkeiten und prüft die Gültigkeit

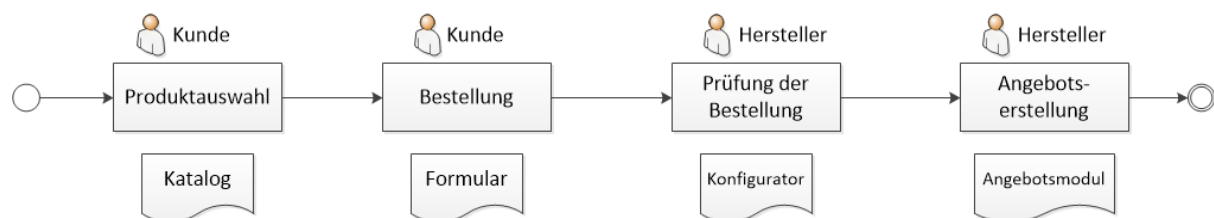


Abbildung 3.1: Auszug eines Konfigurationsprozesses

der Konfiguration. Der letzte Schritt ist die Angebotserstellung. Hierbei wird das konkrete Angebot für den Kunden erstellt.

Bei der Verwendung dieser Prozessbeschreibung entstehen Probleme, die eine Vereinfachung des Prozesses erschweren. Das erste Problem ist die Verwendung von Produktnummern. Diese Nummern werden für eine eindeutige Identifizierung des Produktes bei der Bestellung verwendet. Bei der Verwendung können Fehler entstehen. Es kann die falsche Produktnummer vom Kunden bei der Auswahl verwendet werden, so dass nicht das richtige Produkt bei der Angebotserstellung verwendet wird. Bei der Produktauswahl sind diese Informationen nicht notwendig, da der Kunde sich nur für das konkrete Produkt interessiert. Für die Vereinfachung des Prozesses 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 Prozess vereinfacht.

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, die benötigt werden 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, wodurch der Prozess wiederholt werden muss. Für die Verbesserung dieser Situation muss die Eingabe des Kunden direkt zum Konfigurator gegeben werden. Diese Maßnahme minimiert die Fehler bei der Erfassung.

Weiterhin entsteht durch den gegebenen Prozess ein Kommunikationsproblem. Dies tritt aufgrund der Trennung von Produktauswahl und Produktkonfiguration auf. Der Kunde ist nur bei der Auswahl beteiligt. Das Feedback für die Umsetzung der Zusammenstellung erfolgt erst nach der Weitergabe der Bestellung. Dies ist problematisch, wenn bei der Konfiguration Alternativen auftreten, bei denen der Kunde entscheiden muss 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 sollte ein schnelles Feedback bei der aktuellen Auswahl sein.

Zusammenfassend treten drei grundlegende Probleme bei einem Konfigurationsprozess auf:

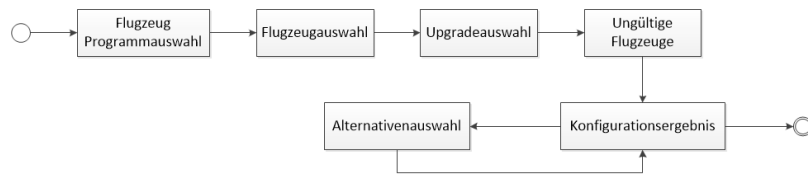


Abbildung 3.2: Programmablauf der Konfigurationslösung

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

Diese drei Punkte müssen im neuen Workflow verbessert werden.

3.1.1 Übertragung auf den Kunden

Beim Anwendungsbeispiel des Kunden 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 ebenfalls anhand des eindeutigen Flugzeugcodes. Beide Codes werden im nächsten Schritt von einem Produktkonfigurator erfasst und anschließend eine Überprüfung der Konfiguration durchgeführt.

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. Bei der Umstellung des Prozesses muss auch dieser Vorgang der Konfiguration verstanden werden, um die Eignung für den neuen Workflow zu überprüfen.

Abbildung 3.1.1 zeigt den Ablauf einer Konfiguration mit dem derzeitigen 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 verwendet werden. Anschließend folgt die Auswahl der entsprechenden Flugzeuge, welche 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 ebenfalls 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 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 die 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, 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 Mobiler 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. Dieses Zusammenlegen von Produktauswahl und Produktkonfiguration in einer Anwendung 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 Workflow des Anwendungsbeispiels

Für die Umsetzung der aus Abschnitt 3.2.1 erstellten Maßnahmen müssen die beiden vorhandenen Systeme für die Auswahl und Konfiguration auf ein gemeinsames Ge-

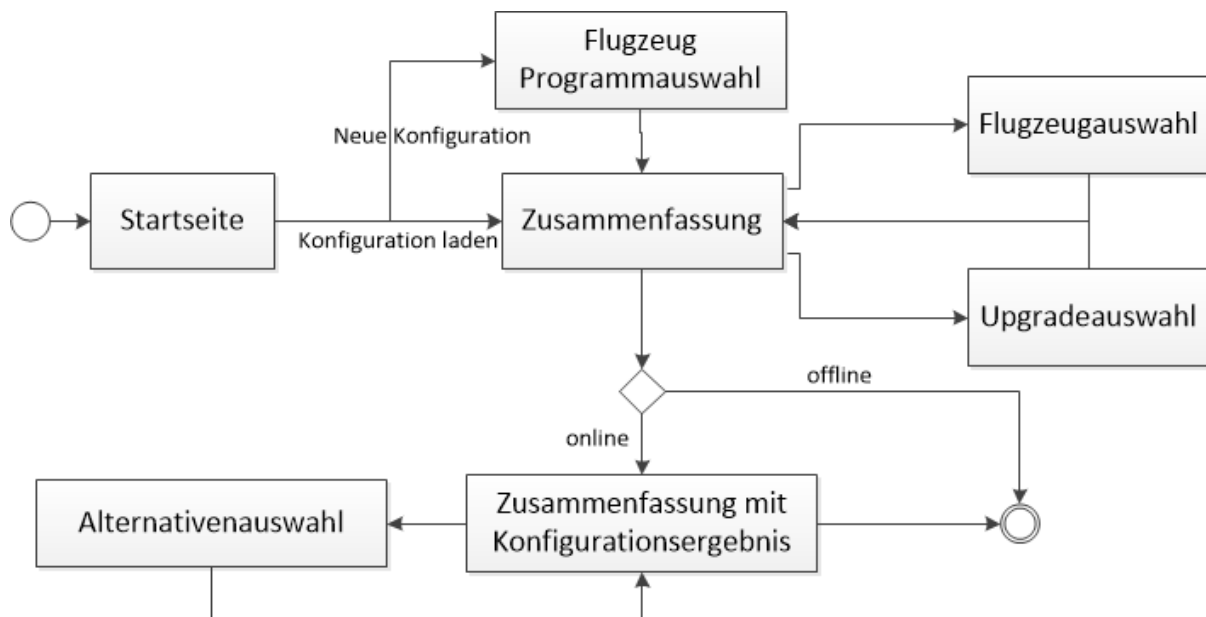


Abbildung 3.3: Programmablauf der Konfigurator-App

rät 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, da 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. Analog zu der Weboberfläche wird bei einer neuen Konfiguration zuerst ein Programm (**Programmauswahl**) 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 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 Prozess mit der Speicherung der Konfiguration beendet.

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. Für eine einfachere 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 zu ermöglichen. Als Basis für diese Anforderungen werden die 10 Heuristiken für Benutzerschnittstellen von Nielsen [NIELSEN] verwendet. Es lassen sich aus dem neuen Workflow folgende Anforderungen spezifizieren:

Einfache Bedienung: Dadurch, dass der Kunde kein Experte ist und die Anwendung nicht jeden Tag verwendet, muss die Bedienung für den Kunden einfach sein. 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 [ERGO] behandelt werden:

- Selbstbeschreibungsfähigkeit
- Lernförderlichkeit
- Erwartungskonformität

Nielsen verwendet hierfür die Begriffe Konsistenz, Erkennung vor Erinnerung und Hilfe bzw. Dokumentation.

Schnelle Bedienung: Die zweite Zielgruppe der Anwendung ist der Experte. Für diesen muss es die Möglichkeit einer schnellen Bedienung der Anwendung geben, so dass er möglichst effizient die Konfiguration 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.

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 3.3.1 dargestellt. Zu jeder Anforderung wird hier ein zusätzlicher Bezeichner definiert.

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

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 3.3.2 zu sehen. Hierbei werden die drei Prozessanforderungen Produktkatalog, Konfiguration und mobile Zielumgebung unterschieden.

NR.	Anforderung	Beschreibung	Prozesszuordnung
F1	Upgradeauswahl	Es sollen Upgrades für Flugzeuge auswählbar sein	Produktkatalog
F2	Flugzeugauswahl	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

4 Entwurf der Benutzerschnittstelle

Die Anforderungen der Anwendung für die Unterstützung des neuen Workflows wurden im vorigen Kapitel abgeleitet. 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, bevor die einzelnen Ansichten entworfen werden.

4.1 Auswahl der Anwendungsplattform

Aufgrund der vielen Möglichkeiten bei der Entwicklung von mobilen Anwendungen muss die Art und Technologie der Anwendung 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. Wichtig bei der Entscheidung ist die Festlegung der konkreten Kriterien. 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 Native 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 kommen, sind Operationen für das System optimiert. Mit einer nativen Anwendung wird damit die beste Performanz erreicht.
- **Betriebssystemspezifische Oberflächenelemente:** Native Anwendungen verwenden für die Benutzerschnittstelle die spezifische Oberflächenelemente. Die Steuerung mit Touch ist ebenfalls auf diese Anwendungsform optimiert. Dies ergibt eine vollständige Verwendungsmöglichkeit des bereitgestellten Frameworks.

Auf die einzelnen Kriterien bezogen erfüllt die native Anwendung alle Anforderungen. Der volle Funktionsumfang des Systems ist gegeben, wodurch die App ideal für die gewählte Zielplattform geeignet ist.

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 das 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, ohne komplette neue Implementierung auf mehreren Systemen ist aufgrund der Kriterien nicht relevant. Die wichtigen Eigenschaften der Anwendung Performanz und lokale Speicherung können mit dieser Anwendungsform gelöst werden, jedoch deutlich schlechter als bei der nativen Variante.

Hybride Anwendung: Durch die Verwendung der lokalen Schnittstellen im nativen Anwendungscontainer (siehe 2.3.3) können die vorhandenen 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 problemlose 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. Damit wird ein zusätzlicher Kommunikationsaufwand benötigt. 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 Bedienung 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. Bei der ersten Betrachtung der Anwendungsformen ist die Implementierung als Web Anwendung nicht sinnvoll. Die Anforderungen können zwar umgesetzt werden, jedoch ist ein erheblicher Mehraufwand für die Erreichung nötig. Ebenfalls hat die Anwendung im Vorhinein bereits Einschränkungen bzgl. der Performanz, sowie

der Bereitstellung von Oberflächenelementen. Die Vorteile einer Implementierung für mehrere Zielsysteme ist bei der Umsetzung kein Kriterium.

Schwieriger ist die Entscheidung zwischen hybrid und nativ. 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. 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.

4.1.2 Anwendungstechnologie

Mit der Entscheidung für eine mobile Anwendungsform kommen drei mögliche Technologien für die Implementierung in Frage. Die Plattformen Android ¹, iOS ² und Windows 8 ³ sind aufgrund ihrer Marktanteile (Android: 43,4% iOS: 48,2% Windows: 7,4% Quelle: [STUD]) 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 [MOBILE2, S.5]). Android hat die meisten Nutzer bei Privatanwendungen (siehe [STUD2]). 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. Ist die Implementierung mit einem Framework umgesetzt, stellt das Implementieren auf einer anderen Zielplattform keine Herausforderung da.

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

¹ <http://www.android.com/>

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

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

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 Zielpattform 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 Alleinstellungsmerkmale 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. Das wichtigste Design Element bei einer Windows 8 Anwendung sind sogenannte Kacheln. Diese Kacheln sind meist quadratisch und in jeder App vorhanden. Jede Funktion wird über eine Kachel erreicht. Sie soll mehr Informationen darstellen, als ein einfaches Logo oder Icon auf einem Button. Durch einen dynamischen Inhalt und unterschiedliche Größen wird dadurch dem Benutzer ein neues Benutzererlebnis gegeben. Die Organisation der Kacheln erfolgt in einem sogenannten Grid (engl. für Gitter). Dieses besteht aus mehreren Quadraten. 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.

Diese 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 eine Erweiterung der Ansicht signalisiert wird.

Beim Aufbau der Anwendung kann entweder eine hierarchische oder flache Struktur verwendet werden. Für den in Abschnitt 3.2.2 erstellen Workflow ist eine hierarchische Architektur passender. Der Ursprung geht hier immer von der Startseite, der sogenannten Hub-Page aus. Diese Seite ist der zentrale Startpunkt von der alle weiteren Aktionen ausgehen. Die zweite Ebene sind 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 wäre 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 zeigen. 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.

4.2.3 Alleinstellungsmerkmal

4.3 Design der Ansichten

4.4 Interaktion der Ansichten

5 Implementierung

5.1 MVVM

5.2 Anwendungsarchitektur

5.3 Implementierung des Katalog-Workflows

5.4 Implementierung des Konfigurations-Workflows

6 Evaluation der Anwendung

Damit ein Kunde einen besseren Einblick in das Produkt erhält, muss der Konfigurationsprozess vereinfacht werden. Die Vereinfachung des Prozesses wird ebenfalls durch eine einfache Bedienung der Anwendung erreicht. Die Sicherstellung, dass dieses Ziel bei der Entwicklung der App erreicht wurde, soll anhand einer heuristischen Evaluation herausgefunden werden. Hierbei wird eine bestimmte Anzahl an Benutzern herausgesucht, die eine Untersuchung der Anwendung durchführen. Die Heuristiken der einzelnen Benutzer ergeben sich aus bekannten Benutzungsprinzipien (siehe: [?]). Die Ergebnisse der Evaluation werden anhand der 10 Heuristiken von Nielsen [?] ausgewertet.

6.1 Durchführung der Evaluation

Für die Umstellung des Prozesses, wie in Kapitel 3 beschrieben muss die Anwendung den neuen Prozess unterstützen. Hier muss die Evaluation die Frage beantworten, ob die Produktauswahl, sowie der Konfigurationsprozess verständlich ist. Damit ein Unterschied festgestellt werden kann, werden bestimmte Aufgaben der Anwendung zuerst auf der alten Konfigurationsoberfläche durchgeführt und anschließend mit der neuen Anwendung. Die Durchführung erfolgt hierbei in einer Interview Form, bei der eine Hilfestellung bei der Bedienung gegeben wird. Anschließend erhält der Anwender einen Fragebogen, der im Anhang zu finden ist. Mit diesem Vorgang erhält man zwei Rückmeldungen. Die Erste durch die verschiedenen Fragen der Benutzer bei der Anwendung und die zweite durch den ausgefüllten Fragebogen. Für eine aussagekräftige Evaluation wurden 6 Testpersonen ausgewählt. Die Personen teilen sich in drei Experten, die bereits das alte System kennen und drei Anwender, die zum ersten Mal das System verwenden.

6.2 Ergebnisse

6.2.1 Auswertung der Interviewfragen

6.2.2 Auswertung der Fragebögen

7 Fazit und Ausblick

Abbildungsverzeichnis

2.1	Aufbau eines Expertensystems [KELLER, s.6]	6
2.2	Architektur der Merlin Enterprise Komponenten	7
3.1	Auszug eines Konfigurationsprozesses	12
3.2	Programmablauf der Konfigurationslösung	14
3.3	Programmablauf der Konfigurator-App	17

Listings

Literaturverzeichnis

- [MASS] B. JOSEPH PINE **Harvard Business Press** Mass customization, 1.Auflage , 1993
- [APP] STEPHAN VERCLAS, CLAUDIA LINNHOF-POPIEN **Springerverlag** Smart Mobile Apps: Mit Business-Apps ins Zeitalter mobiler Geschäftsprozesse, 1.Auflage , Dezember 2011
- [DESIGN] CHRISTIAN KRAUS **2kit consulting** Grundlagen und Erfolgsfaktoren für Apps im Mobile Business, Whitepaper , 16.05.2012
- [PRODUCT] **Wirtschaftslexikon** Produkt, <http://www.wirtschaftslexikon24.com/d/produkt/produkt.htm>, aufgerufen am 25.07.2013
- [BOOL1] HELMUT HEROLD, BRUNO LURZ, JÜRGEN WOHLRAB **Pearson Studium** Grundlagen der Informatik, München 2006
- [BOOL2] GERALD TESCHL, SUSANNE TESCHL **examen.press** Mathematik für Informatiker, Pearson Studium, Wien 2009, 3. Auflage
- [PUPPE] FRANK PUPPE **Springerverlag** Einführung in Expertensysteme, 1. Auflag, 1988
- [KELLER] HUBERT B. KELLER **DH Karlsruhe** Wissensbasierte Systeme - Einführung, Vorlesung SS 2007
- [EXPERT] 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
- [TABLET] HAUFЕ Tablets setzen sich durch, http://www.haufe.de/marketing-vertrieb/vertrieb/vertrieb-tablets-setzen-sich-durch_130_155918.html, aufgerufen am 26.07.2013

- [MOBILE1] HANS H. BAUER, THORSTEN DIRKS, MELCHIOR D. BYANT **examen.press** Erfolgsfaktoren des mobile Marketing, Springer, 2008
- [MOBILE2] **Pierre Audoin Consultants** Tablets im Unternehmenseinsatz, http://www.berlecon.de/studien/downloads/PAC_Trendstudie_Microsoft_Mobility_May_11.pdf, Mai 2011
- [RCP] RALF EBERT Eclipse RCP, http://www.ralfebert.de/eclipse_rcp/EclipseRCP.pdf, Version 1.1, 19.08.2011
- [ERGO] CHRISTIANE RUDLOF **Unfallkasse Post und Telekom** Handbuch Software-Ergonomie. Usebility Engineering., <http://www.ukpt.de/pages/dateien/software-ergonomie.pdf>, S.52, 2. Auflage, Tübingen 2006
- [PLATTFORM] 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
- [PLATTFORM2] SMARTDIGITS Native App oder Webb App, <http://www.smart-digits.com/2012/02/native-app-oder-web-app-teil-1-definitionen-und-entscheidungskriterien/>, aufgerufen am 23.07.2013
- [WEBAPP1] BUNDESAMT FÜR SICHERHEIT IN DER INFORMATIK **SecureNet** Sicherheit von Webanwendungen, https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Publikationen/Studien/WebSec/WebSec_pdf.pdf?_blob=publicationFile, 1. Version, August 2006
- [WEBAPP2] ANDREAS SCHAFFRY **CIO** Die Zukunft mobiler Anwendungen, <http://www.cio.de/strategien/2910477/index.html>, aufgerufen am 27.07.2013
- [NIELSEN] JAKOB NIELSEN **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
- [STUD] **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

[STUD2] **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-4CDAFDBB3F3C/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/in-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