



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	16
3.2.1	Mobiler Konfigurationsprozess	16
3.2.2	Workflow der App	17
3.3	Grundlegende Anforderungen	18
3.3.1	Nicht-Funktionale Anforderungen	20
3.3.2	Funktionale Anforderungen	21
4	Entwurf der Benutzeroberfläche	22
4.1	Auswahl der Anwendungsplattform	22
4.1.1	Anwendungsform	22
4.1.2	Anwendungstechnologie	22
4.1.3	Native Anwendungen	23
4.1.4	Web Anwendungen	24

4.1.5	Hybride Anwendungen	24
4.1.6	Abwägung	24
4.2	Untersuchung der Plattform Windows 8	24
4.2.1	Bedienkonzepte	24
4.3	Design der Ansichten	24
4.4	Interaktion der Ansichten	24
5	Implementierung	25
5.1	MVVM	25
5.2	Anwendungsarchitektur	25
5.3	Implementierung des Katalog-Workflows	25
5.4	Implementierung des Konfigurations-Workflows	25
6	Evaluation der Anwendung	26
6.1	Durchführung der Evaluation	26
6.2	Ergebnisse	27
6.2.1	Auswertung der Interviewfragen	27
6.2.2	Auswertung der Fragebögen	27
7	Fazit und Ausblick	28
	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, umso 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 Anpassung 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 mit den Abhängigkeiten der Einzelbauteile lassen sich mithilfe der booleschen Algebra verknüpfen. 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 der Reifen gefordert wird, wenn ein Motor und ein Getriebe ausgewählt wurde. Eine Beispiel Modellierung 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

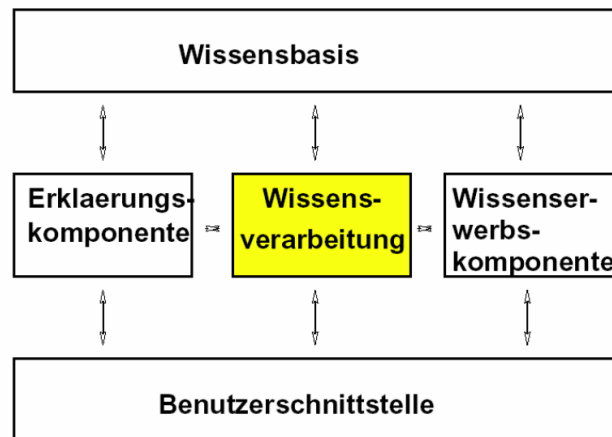


Abbildung 2.1: Aufbau eines Expertensystems [KELLER, s.6]

können 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 *Erklärungskomponente* 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

¹ <http://www.airbus.com/>

Kunden fördern [TABLET]. Die Vorteile durch den großen Bildschirm und 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 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 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 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 [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 Vereinfachung eines Konfigurationsprozesses bei der Verwendung von mobilen Anwendungen soll am konkreten Anwendungsbeispiel gezeigt werden. Die Analyse beginnt bei grundlegenden Fragen, wie ein solcher Prozess aufgebaut ist und wird im Folgenden anhand eines konkreten Kundenfall weiter ausgebaut.

Ein Auszug aus einem Konfigurationsprozess ist in Abbildung 3.1 zu sehen. Der Prozess beginnt mit der Produktauswahl. Zu Beginn möchte der Kunde ein konkretes Produkt auswählen. Dies geschieht über den Produktkatalog. In diesem findet er sein gewünschtes Produkt. 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

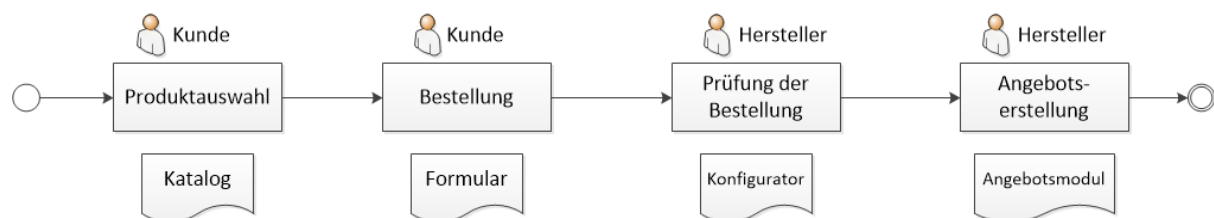


Abbildung 3.1: Auszug eines Konfigurationsprozesses

die Gültigkeit 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. Erstes 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:

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

Diese drei Punkte werden in der Modellierung des neuen Workflows verbessert.

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 herauszufinden.

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. Die Auswahl kann nach bestimmten Kriterien gefiltert werden. Bei der Bestellung eines Upgrades wird die Flugzeugnummer angegeben, womit eine schnelle Auswahl erfolgen kann. Sind die Flugzeuge ausgewählt, werden Upgrades aus einer Liste selektiert.

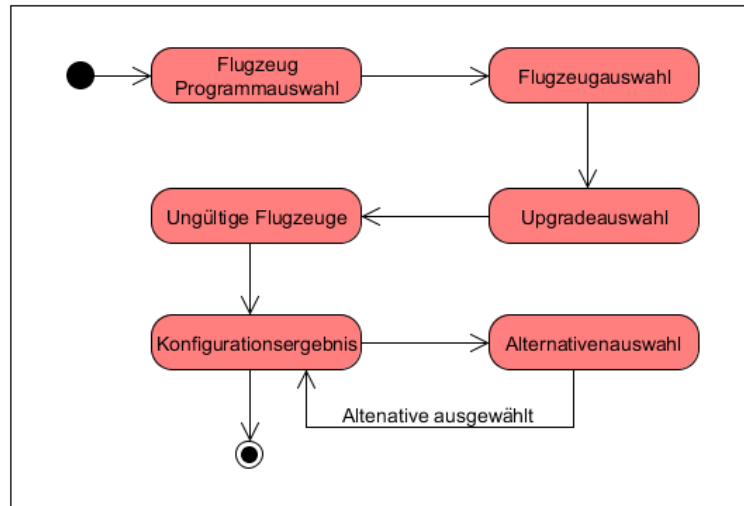


Abbildung 3.2: Programmablauf der Konfigurationslösung

Es sind alle verfügbaren Updates aufgelistet. Die Auswahl erfolgt ebenfalls mit der eindeutigen Nummer, welche bei der Bestellung angegeben wurde.

Nach der letzten Auswahl 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, werden 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 Flugzeuge eingebaut werden müssen. Aus dem Dokument wird ein Upgrade-Angebot erstellt, dass 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 mit der aktuellen Anwendung der Kunde die Konfiguration 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, der den Lösungsweg beinhaltet.

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 Idealfalls 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 unterstützen. Gleichzeitig wird hier ein besseres Verständnis für das Produkt ermöglicht. Der Hersteller hat den Vorteil der Beschleunigung des Prozesses. Er kann durch die direkte Auswahl der Produkte, sowie ein sofortiges Prüfen und ggf. eine Selektion der Alternativen die Konfiguration abschließen und ein konkretes Angebot erstellen.

Diese Umstellungen des Prozesses setzt die Verwendung eines mobilen Endgerätes, in diesem Fall einem Tablet-PC 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 der App

Der Anwendungsworkflow muss zu dem neuen fachlichen Prozess passen und ihn unterstützen. Dies geschieht durch Anpassungen an die Kommunikation mit dem Konfigurationsserver. Der aktuelle Client und der Server befinden sich auf der gleichen Umgebung. Somit können beide direkt miteinander kommunizieren und benötigen keine aufwändigen Webservice Schnittstellen. Für den neuen Prozess muss der Konfigurationsserver von der mobilen Umgebung erreichbar sein. Dies bedeutet, dass es eine Schnittstelle für die App geben muss. Die Kommunikation mit dem Webserver kann durch den mobilen Einsatz erschwert werden. Dies ist beispielsweise der Fall, wenn die Konfiguration beim Kunden vor Ort stattfindet und eine schlechte, bis gar keine Verbindung vorhanden ist. Aus diesem Grund ist bei den Nicht-Funktionalen Anforderung in Abschnitt 3.3.1 ein sogenannter Offline Modus enthalten, der die Zusammenstellung einer Konfiguration auch ohne Anbindung an den Konfigurator ermöglicht. Um eine flüssige Navigation gewährleisten zu können, muss die Kommunikation mit dem Server auf das Wesentliche konzentriert sein. Dies bedeutet, dass eine Konfiguration erst am Ende stattfindet und die einzelnen Schritte, die im derzeitigen Client durchgeführt werden zusammengelegt werden müssen.

In Abbildung 3.2.2 ist der Anwendungsverlauf der App zu sehen. Analog zu der Weboberfläche wird bei einer neuen Konfiguration zuerst ein Programm (**Programmauswahl**) ausgewählt. Nach der Auswahl gelangt der Benutzer in eine **Zusammenfassung** der aktuellen Konfiguration. Von dieser Ansicht aus können Flugzeuge (**Flugzeugauswahl**) oder Upgrades (**Upgradeauswahl**) selektiert werden. Sobald beides vorhanden ist, wird die Erreichbarkeit der Konfigurationsschnittstelle geprüft. Bei Verfügbarkeit befinden wir uns im "Online" Modus und es werden die Konfigurationsgruppen angezeigt. Nach erfolgter Konfiguration werden die Alternativen in einer separaten Ansicht (**Alternativenauswahl**) ausgewählt. Ist die Konfiguration vollständig, so kann das Upgrade direkt bestellt werden und den Update Vorgang beenden. Ist der Server nicht verfügbar, kann der Benutzer die aktuelle Konfiguration speichern und damit den

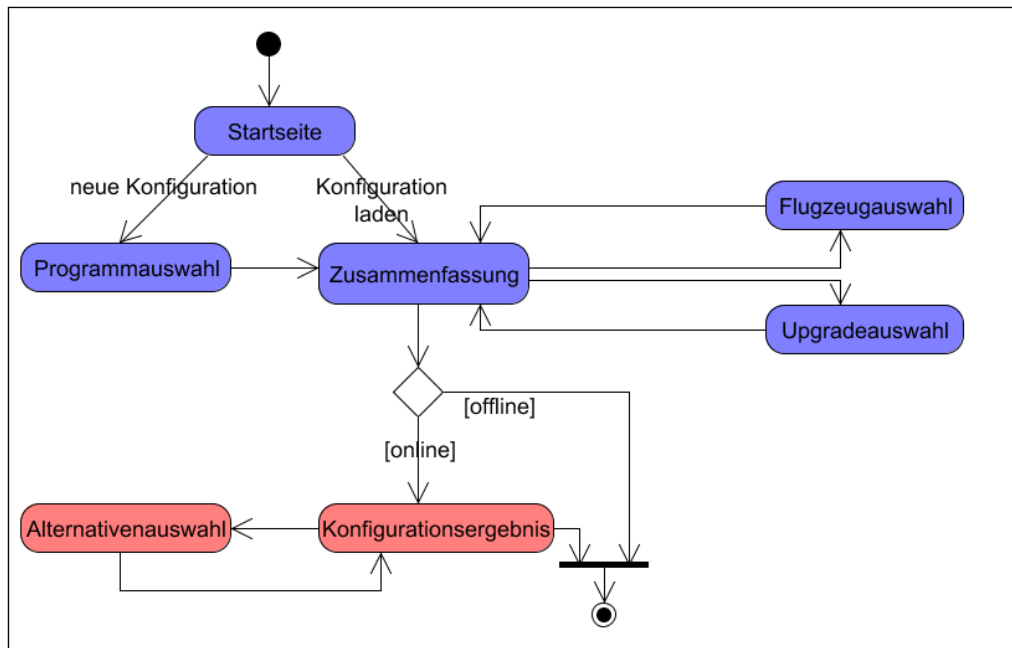


Abbildung 3.3: Programmablauf der Konfigurator-App

Prozess abschließen. Anschließend kann die Auswahl direkt über den Startbildschirm geladen werden.

Um für einen möglichst geringen Kommunikationsaufwand mit dem Konfigurationsserver zu sorgen, wird der Server erst nach erfolgter Auswahl bedient. Der Typ bezieht sich auf eine Art Die Kommunikation wird alle zuvor beinhalteten Einzelschritte auf dem aktuellen Client zu einem Zusammenfügen. Die Trennung von Online (rot) und Offline (blau) ist in der Abbildung ebenfalls aufgezeigt.

3.3 Grundlegende Anforderungen

Die Anwendung ist kein vom Kunden vorgegebenes Projekt. Dies hat zur Folge, dass es keine strikten Anforderungen an die App gibt. Die Anwendung soll, wie in Abschnitt 1.2 festgelegt den Konfigurationsprozess des Kunden vereinfachen und "mobil" verfügbar machen. Der Prototyp soll den Kunden überzeugen, seinen derzeitigen Konfigurationsprozess in den Neuen zu überführen. Die Anforderungen an die Software müssen demnach so festgelegt werden, dass sie für eine hohe Akzeptanz beim

Kunden sorgt. Damit die Anwendung akzeptiert wird, müssen alle Gütekriterien dieses Ziel verfolgen.

Die Festlegung der Kriterien ist nach Rücksprache mit den Entwicklern und dem Projektleiter des Kunden entstanden. Gemeinsam wurde eine Priorisierung der einzelnen Anforderungen entsprechend der Zielsetzung erarbeitet. Im Folgenden ist das Ergebnis in Funktionale und Nicht-Funktionale Anforderungen eingeordnet. Fachliche Funktionen der Anwendung sind in den funktionalen Anforderungen enthalten. Diese wurden anhand der bestehenden Software und den Möglichkeiten des Konfigurators erstellt. Weiterhin wurden zusätzliche Anforderungen, die aufgrund der mobilen Umgebung notwendig sind, hinzugefügt. Bei den Nicht-Funktionalen Kriterien steht die Usability, das Anpassen an die Umgebung , sowie die Kundenbegeisterung im Vordergrund.

3.3.1 Nicht-Funktionale Anforderungen

NR.	Anforderung	Beschreibung	Details	Priorität
N1	Einfache Bedienung	Die Anwendung soll von unerfahrenen Benutzern schnell bedient werden können.	Schwerpunkte der Softwareergonomie[ERGO]: <ul style="list-style-type: none"> • Selbstbeschreibungsfähigkeit • Lernförderlichkeit • Erwartungskonformität 	A
N2	Schnelle Bedienung	Die Auswahl der einzelnen Komponenten soll möglichst schnell getätigt werden können.	Es muss für eine flüssige Navigation zwischen den einzelnen Seiten gesorgt werden. Der Benutzer sollte keine langen Wartezeiten beim Bedienen der Anwendung haben.	A
N3	Offline-Online Modus	Die Anwendung soll einen Offline- und Online-Modus enthalten	Es sollen Konfigurationsdaten offline zusammengestellt werden können, die später oder zeitgleich online (mit Konfigurationsserver) geprüft werden.	A
N4	Alleinstellungsmerkmal	Um den Kunden zu begeistern ist ein besonderes Feature, welches die Anwendung als Alleinstellungsmerkmal besitzt wünschenswert.	Es sollen besonders "aufregende" Features, die eine Technologie bietet in die Anwendung integriert und in einem sinnvollen Kontext genutzt werden.	B

3.3.2 Funktionale Anforderungen

NR.	Anforderung	Beschreibung	Details	Priorität
F1	Upgradeauswahl	Es sollen Upgrades für Flugzeuge auswählbar sein	Bevor die Auswahl stattfindet, soll es verschiedene Filtermöglichkeiten geben, wie beispielsweise der Flugzeugtyp oder die Art des Upgrades.	A
F2	Flugzeugauswahl	Es müssen Flugzeuge eines bestimmten Kunden auswählbar sein.	Es soll auch hier, analog zu B1 die Möglichkeit der Auswahl von Flugzeugen auf einen vorher gefilterten Datensatz geben.	A
F3	Konfiguration	Nach erfolgter Auswahl werden die Ergebnisse der Konfiguration dargestellt.	Kommunikation mit dem Konfigurationsserver, sowie die Darstellung der Ergebnisse auf dem Client	A
F4	Alternativen	Bei mehreren Möglichkeiten einer Auswahl sollen Alternativen ausgewählt werden können	Übersichtliche Darstellung von Alternativenauswahl.	A
F5	Speichern	Die getätigte Auswahl soll gespeichert werden	Nach Abschluss der Konfiguration soll diese lokal gespeichert werden, um ein erneutes Laden zu ermöglichen.	B
F6	Laden	Getätigte Konfigurationen müssen geladen werden können	Beim Start der Anwendung sollen getätigte Konfigurationen geladen und angezeigt werden können	B

4 Entwurf der Benutzeroberfläche

4.1 Auswahl der Anwendungsplattform

4.1.1 Anwendungsform

4.1.2 Anwendungstechnologie

Nachdem der Workflow, sowie die Anforderungen der Anwendung erläutert wurden, folgt in diesem Abschnitt die Auswahl der passenden Anwendungsplattform. Es werden die drei verschiedenen Anwendungstypen nativ, web und hybrid untersucht [PLATTFORM]. Bei jeder dieser Plattformen werden die bekanntesten Technologien verwendet. Die Untersuchung beruht auf den in Kapitel 3.3 gestellten Anforderungen. Es werden folgende Auswahlkriterien verwendet:

- **Offlinemodus:** Können offline Daten beliebig gespeichert werden? Gibt es eine passende Technologie für einen solchen Modus?
- **Sicherheit:** Da bei der Konfiguration sensible Daten verwendet werden, muss die Anwendung entsprechend gesichert werden können. Die einzelnen Sicherungsmaßnahmen sollen hierbei untersucht werden.
- **Schnittstelle:** Welche Möglichkeit für eine Schnittstelle zu dem Konfiguratorserver gibt es?
- **Kundenpräferenz:** Gibt es eine bestimmte Technologie/Plattform, die vom Kunden explizit gewünscht ist? Hat die Technologie ein besonderes Alleinstellungsmerkmal?
- **Performance:** Für die Anzeige der einzelnen Produktbilder, sowie eine schnelle Navigation innerhalb der Anwendung ist die Performance des jeweiligen Typs wichtig.

Bei der Auswahl der Technologien werden nur Plattformen ausgewählt, die auf Tabletgeräten verfügbar sind. Es werden alle Kriterien für den Anwendungstyp bewertet und zusätzlich für die einzelnen Technologien.

4.1.3 Native Anwendungen

Native Anwendungen sind auf die jeweilige Zielplattform beschränkt. Dieser Typ einer Anwendung beschränkt sich auf das zuvor gewählte Betriebssystem. Der geschriebene Programmcode kann meistens nur auf der Zielplattform verwendet werden. Für die Bewertung von nativen Anwendungen werden die drei am Meisten genutzten Betriebssysteme im Tabletbereich verwendet: Android, iOS und Windows 8[NATIVE].

Offlinemodus: Bei einer nativen App ist keine Serveranbindung notwendig. Alle Daten können ohne eine Internetverbindung lokal auf dem Gerät selbst vorhanden sein. In einer nativen Anwendung können alle Dateioperationen, wie auf einem normalen Betriebssystem durchgeführt werden. Dies hat den großen Vorteil, dass auf den Speicher zugegriffen werden kann. Große Datensätze, die beim Offlinemodus anfallen, können damit auf dem Gerät gespeichert werden. In diesem Kriterium ist das Verhalten für alle drei Betriebssysteme gleich, womit keine Unterscheidung der einzelnen Technologien notwendig ist.

Sicherheit: Durch die native Umgebung ist die Anwendung nur auf dem Betriebssystem vorhanden und in einer geschlossenen Umgebung. Die Kommunikation mit dem Webserver muss allerdings selbstständig verschlüsselt werden, insbesondere wenn Die Anwendung bei einem Kunden in dessen Netz verwendet wird. Ein weiteres Problem, welches native Anwendungen besitzen sind die lokalen Sicherheitslücken eines jedes Betriebssystems. Regelmäßig tauchen neue versteckte Fehler im Betriebssystem auf, die Schaden zufügen könnten.

- **Android und iOS:** Bei beiden Betriebssystemen werden regelmäßig Sicherheitslücken offengelegt. Die Angriffe auf beide Systeme ist zunehmend, wobei das iOS-System im Vergleich besser abschneidet und in mehreren
- **Windows 8:** Durch die hohe Verbreitung von Windows Betriebssystemen sind diese Systeme besonders anfällig, bzw. oft verwendete Ziele bei Angriffen. Die breite Anzahl der Nutzer sorgt jedoch für ausreichend Lösungen zur Sicherung der Anwendung.

Schnittstelle: Für die Kommunikation mit dem Konfigurationsserver bieten sich bei einer nativen Anwendung Webservices an. Diese können in den Standards

4.1.4 Web Anwendungen

phone gapp, gwt,

4.1.5 Hybride Anwendungen

4.1.6 Abwägung

4.2 Untersuchung der Plattform Windows 8

4.2.1 Bedienkonzepte

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	15
3.3	Programmablauf der Konfigurator-App	18

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> 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] HAUFE 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_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
- [NATIVE] **STATISTA** Apple bleibt Marktführer, <http://de.statista.com/themen/580/tablets/infografik-der-weltweiten-marktanteile-der-tablet-betriebssysteme/>, 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/WebSec1.1.pdf>, 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
- [WIN8-1] **BART CLAEYS, QIXING ZHENG MSDN** Designfallstudie: vom iPad zur Windows Store-App, <http://msdn.microsoft.com/de-de/library/windows/apps/hh868262>, aufgerufen am 16.05.2013
- [WIN8-2] **MICROSOFT MSDN** Entwerfen großartiger Produktivitäts-Apps für Windows, <http://msdn.microsoft.com/de-de/library/windows/apps/hh868273>, aufgerufen am 16.05.2013

[WIN8-3] MICROSOFT **MSDN** Shopping-Apps, <http://msdn.microsoft.com/de-de/library/windows/apps/jj635241.aspx>, aufgerufen am 16.05.2013

[WIN8-4] MICROSOFT **MSDN** am 16.05.2013 Planen ihrer App, <http://msdn.microsoft.com/de-de/library/windows/apps/hh465427>, aufgerufe