Endbericht

erstellt von	Daniel Böhm
	Frederick Geist
	Andrey Behrens
	Timm Vollmer
Abgabe	09.08.2009
Auftraggeber	Rent-A-Jet
Prüfer	Prof. Dr. Dr. H. Neunteufel
	Hochschule Wismar
	Fachbereich Wirtschaft

Inhaltsverzeichnis

1 Einleitung

Dieser vorliegende Endbericht beschreibt die theoretischen Grundlagen und die technischen Aspekte der Software, die durch das Unternehmen ThinkLogics für das Unternehmen Rent-A-Jet AG erstellt wurde. Im Rahmen dieses vorliegenden Endberichtes werden Abläufe dokumentiert sowie wichtige Softwarefunktionalitäten dargestellt.

Neben dem genannten Lieferumfang werden zum besseren Verständnis die Geschäftsprozesse sowie diverse Abläufe bzw. Funktionalitäten graphisch repräsentiert.

1.1 Ausgangssituation

Das Unternehmen Rent-A-Jet AG mit Sitz in Wismar hat im Jahr 2008 in einer ordentlichen Hauptversammlung das Maßnahmepaket “Optimierung der wichtigsten Geschäftsbereiche für die Erreichung mittel- und langfristiger Umsatzsteigerung und kurzfristiger Kostenminimierung” verabschiedet. Das beschlossene Paket gliederte sich wie folgt:

- Optimierung interner Prozesse durch IT-Lösungen
- Optimierung des Marketingmixes für das Jahr 2009 durch Outsourcing aller Werbemaßnahmen an ein Drittunternehmen
- Optimierung des Controllings durch Einsatz einer externen Unternehmensberatung und Einrichtung einer eigenen Controllingabteilung
- Optimierung des Humankapitals durch Einstellung neuer Pilotinnen und Piloten in Festanstellung, sowie Anschluss des Unternehmens an den Mitarbeiterpool für Flugbegleiterinnen und Flugbegleiter des Wismarer Vereins zur Förderung des Berufes der Flugbegleiterinnen und Flugbegleiter e.V.

Konkrete Maßnahmen zur Erreichung von Kostenminimierung sowie Modernisierung von internen Prozessen sollte durch die Maßnahme OPT-IT2 erreicht werden.

Die Maßnahme OP-IT 2 beinhaltete die Implementierung einer neuen Software. Ziel dieser neuen Software ist, die Mitarbeiter in der Verwaltung bei allen internen Prozessen wie Rechnungserstellung, Zahlungsverfolgung und Angebotserstellung zu unterstützen.

1.2 Aufgabenstellung

Im Rahmen einer Ausschreibung erhielt das Unternehmen ThinkLogics den Zuschlag eine neue innovative Software zu entwickeln, die den genannten Anforderungen gerecht wird. Zudem hatte das Unternehmen Rent-A-Jet AG eine Standardsoftware genutzt, die nicht den heutigen Anforderungen entsprach.

Daher galt es nicht nur eine kostengünstige Software zu entwickeln, sondern eine Software, die sich individuell an die Bedürfnisse des Unternehmens anpasst. Zudem sollte eine vereinfachte Bedienbarkeit gewährleistet sein, damit Mitarbeiter zeitnah mit der neuen Software konfrontiert werden und somit laufende Geschäftsprozesse bestehen bleiben.

Individualisierte Softwareapplikationen ermöglichen eine Erweiterbarkeit von Funktionen und spätere Wünsche des Kunden können berücksichtigt werden. Besondere bzw. weitere spezielle Vorgaben wurden vom Auftraggeber nicht gestellt.

1.3 Einsatzgebiet

Die entwickelte Software ist auf die Bedürfnisse und Belange des Unternehmens Rent-A-Jet AG zugeschnitten. Der gewerbliche Zweck der Rent-A-Jet AG ist die Vermietung von Flugzeugen an Geschäfts- sowie privaten Kunden. Dementsprechend setzt die Software eine individualisierte Anpassung voraus. Mit geringem Aufwand sollen Mitarbeiter in der Lage sein neue Kundenstammdaten anzulegen, Angebote sowie Rechnungen zu erstellen und Verträge auszudrucken.

Neben diesen genannten Einstellungen, bietet die Software die Möglichkeit auch administrative Einstellungen, hinsichtlich der Verwaltung von Flugzeugen sowie den damit verbundenen Eigenschaften, vorzunehmen.

Im Rahmen der administrativen Verwaltung sind folgende Einstellungen bzw. Modifikationen möglich:

- Anlegen/ Ändern von neuen Flugzeugherstellern
- Anlegen/ Ändern von neuen Flugzeugtypen
- Anlegen/ Ändern von Flugzeugplätzen
- Anlegen/ Ändern von Mehrwersteuersätzen
- Anlegen/ Ändern von Vorgangstypen
- Anlegen/ Ändern von Leistungstypen
- Anlegen/ Ändern von Zufriedenheitstypen
- Erstellen von Statistikformaten

1.4 Technische Ausgangssituation

Voraussetzung für die Softwareerstellung ist die Kompatibilität mit IBM-PCs, da das Unternehmen Rent-A-Jet AG ausschließlich IBM-PCs nutzt. Die Forderung eine webbasierte Anwendung zu erstellen hat das Unternehmen ThinkLogics, unter Einbeziehung von kostengünstigen Aspekten, entsprechend berücksichtigt. Neben diesen genannten Forderungen hat ThinkLogics z.B. im Rahmen der Reporterstellung von Statistiken verschiedene Ausgabeformate berücksichtigt und implementiert.

Auf detaillierte Aspekte der Umsetzung und Implementation der genannten Lösung wird in den nachfolgenden Kapiteln eingegangen.

2 Theoretische Grundlagen

2.1 Vorgehensmodell

Da die Erstellung von Software ein komplexer Entwicklungsprozess ist, muss zu Beginn eines Projekts ein passendes Vorgehensmodell gewählt werden. Das Vorgehensmodell unterteilt den Entwicklungsprozess in zeitlich und inhaltlich begrenzte Phasen wodurch die Fertigstellung der Software Schritt für Schritt erfolgen kann. Außerdem geben Sie den Projektablauf vor.

Bekannte Vorgehensmodelle sind:

- Wasserfallmodell
- Spiralmodell
- V-Modell

Das Wasserfallmodell ist ein lineares Vorgehensmodell, bei dem die einzelnen Projektphasen nacheinander durchlaufen werden. Jedes Phasenergebnis ist Ausgangspunkt für die nächst tiefere Phase. Eine Wiederholung bzw. Iteration einzelner Phasen findet nicht statt. Ein Phasenergebnis stellt den Abschluss einer Phase und den Beginn der darauf folgenden Phase dar.

Das Spiralmodell ist ein iteratives Vorgehensmodell. Dabei wird ein bestimmter Projektzyklus mehrfach durchlaufen. Das Ergebnis eines Durchlaufs ist ein Teilergebnis des Gesamtprojekts. Ein Zyklus kann in folgende vier Phasen unterteilt sein:

1) Festlegung der Ziele
2) Analyse
3) Realisierung und Überprüfung
4) Planung des nächsten Zyklus

Das V-Modell ist eine Beschreibung der Projektaktivitäten und der Projektergebnisse / -produkte die während der Softwareentwicklung durchzuführen bzw. zu erstellen sind. Dabei werden alle Projektaktivitäten V-förmig angeordnet, Analyse und Entwurf im absteigenden Ast, Realisierung, Integration und Softwaretest im aufsteigenden Ast. Eine zeitliche Abfolge wird nicht beschrieben, Aktivitäten können jedoch auf Spiral- oder Wasserfallmodelle übertragen werden.

2.2 Aufgabenbereiche

Innerhalb eines Entwicklungsprojekts gibt es verschiedene Aufgabenbereiche, die im Folgenden kurz beschrieben werden.

2.2.1 Projektmanagement

Grundsätzliche Aufgabe des Projektmanagements ist die Planung, Überwachung und Steuerung des gesamten Projektverlaufs. Das Projektmanagement ist die zentrale Anlaufstelle für Auftraggeber und Projektmitglieder.

Aufgaben Projektmanagements:

- Festlegen der Projektrahmenbedingungen
- Beteiligung an der Aufwandsschätzung
- Planung des Projektverlaufs
- Beauftragung der Projektmitarbeiter
- Kontinuierliche Ermittlung und Bewertung des Projektstatuses
- Einleiten von steuernden Maßnahmen
- Kommunikation mit Auftraggeber und Projektmitglieder
- Projektabschluss

2.2.2 Analyse und Architektur

Als Grundlage für die gesamte Entwicklungs- und Testarbeit muss, basierend auf dem Anforderungen des Kunden, eine technische Gesamtkonzeption der Software erarbeitet werden.

Aufgaben Analyse und Architektur:

- Analyse der Kundenanforderungen
- Erstellung eines Fachkonzepts mit einer Beschreibung der Umsetzung
- Definition von Entwicklungspaketen
- Definition der Anforderungen für die Erhebung von Testdaten

2.2.2.1 Entwicklung

Die Entwicklung ist zuständig für die technische Umsetzung des Fachkonzepts.

Aufgaben Entwicklung:

- Bereitstellen einer Entwicklungsinfrastruktur
- Umsetzung der Entwicklungspakete

2.2.2.2 Datenerhebung

Im Rahmen der Datenerhebung werden alle benötigten Daten gesammelt, technisch aufbereitet und zur Verfügung gestellt. Dabei sind die Anforderungen aus Analyse und Architektur zu beachten.

Aufgaben Datenerhebung:

- Erfassung und Aufbereitung aller benötigten Daten
- Erfassung von exemplarischen Vorgängen
- Definition von wichtigen Reports

2.2.2.3 Qualitätsmanagement

Das Qualitätsmanagement und die Qualitätssicherung verfolgen das Projekt während der gesamten Projektdauer. Es soll sichergestellt werden, dass das Projekt bestimmten Qualitätsanforderungen gerecht wird. Das Qualitätsmanagement teilt sich dabei in zwei Bereich auf. Zum einen müssen die Qualitätsanforderungen definiert werden, zum Anderen müssen in der Qualitätssicherung die Einhaltung der Anforderungen überprüft werden.

Aufgaben Qualitätsmanagement:

- Qualitätsrichtlinien definieren (z.B. Programmierrichtlinien)
- Einhaltung der Richtlinien überprüfen (z.B. Code Inspektion, Unit-Tests, User-Acceptance-Tests)

2.2.2.4 Dokumentation

Im Rahmen der Dokumentation werden alle Dokumente die innerhalb des Projekts benötigt werden erstellt.

Aufgaben Dokumentation:

- Erstellungen von Kundenpräsentationen
- Erstellung von Handbüchern
- Erstellung von Berichten

2.3 Softwareapplikation

Grundsätzlich kann eine Software als klassische Desktop- oder Client-Server Anwendung realisiert werden. Zu den Client-Serveranwendungen zählen auch die Webanwendungen. Beides bietet Vor- und Nachteile.

Vorteile von Webanwendungen

- Keine Softwareinstallation am Arbeitsplatz
- Dadurch geringe Kosten am Arbeitsplatz
- Ortsunabhängigkeit
- Zentrale Verwaltung der Daten bietet Datensicherheit und Datenqualität
- Zentrale Verwaltung des Programmcodes ermöglicht schnellere Updates und Fehlerbeseitigungen
- Zur Zeit preiswerte Entwicklung und Pflege
- Systemunabhängigkeit

Nachteile von Webanwendungen

- Internetverbindung wird benötigt
- Geeigneter Webserver wird benötigt
- Darstellung der Anwendung ist abhängig von Browser des Clients
- Integration in bestehenden Desktopanwendungen (z.B. Outlook) schwierig

Die Rent-A-Jet Angebotssoftware verfolgt einen weborientierten Ansatz. Die Erreichbarkeit ist allerdings auf das lokale Intranet / LAN des Unternehmens begrenzt. Eine Internetverbindung wird nicht benötigt, lediglich eine Anbindung an das Local Area Network (LAN) der Firma Rent-A-Jet muss vorhanden sein. Die Bedienung der Software erfolgt ausschließlich durch Vertriebsmitarbeiter der Firma Rent-A-Jet, welche Kundenanfragen schriftlich oder telefonisch entgegen nehmen und mit Hilfe der Software weiterverarbeiten.

Der weborientiert Ansatz bietet eine ideale Grundlage für eventuelle spätere Erweiterungen. Dies können Onlinebestellmöglichkeiten für Kunden, Onlineabfrage des Auftragsstatuses oder Vernetzung mehrerer Standorte über das Internet sein.

2.4 Programmiersprache

Bei der Erstellung der Rent-A-Jet Angebotssoftware kamen folgende Programmiersprachen zum Einsatz:

- Serverseitig: PHP
- Clientseitig: JavaScript

Zur Formatierung der Webseitenausgabe wurde HTML und CSS verwendet. Die wichtigsten Sprachen werden in den folgenden Abschnitte kurz beschrieben.

2.4.1 PHP

PHP ist eine Open-Source-Software die 1995 erstmals durch Rasmus Lerdorf vorgestellt wurde. Der Begriff PHP steht für Hypertext Preprocessor und entstand ursprünglich durch die Bezeichnung „Personal Home Page Tools“. PHP ist eine serverseitige Interpretersprache die hauptsächlich zur Erstellung von dynamischen Webanwendungen verwendet wird. Seit dem ersten Release wurde die Sprachen kontinuierlich weiterentwickelt. Am 30 Juni 2009 erschien PHP in der Version 5.3. Für den Einsatz von PHP basierenden Webanwendungen wird ein Apache Webserver benötigt.

2.4.2 Javascript

Javascript ist eine Scriptsprache die es ermöglicht Anwendungen innerhalb des Browsers auszuführen. Die Javascript Anwendung wird dabei direkt in die Webseite integriert und zur Laufzeit vom Browser interpretiert. Ihre Ursprünge hat Javascript in der 1995 von Netscape entwickelten Sprache LiveScript. LiveScript diente in erste Linie dazu Formulareingaben eines Benutzers vor dem Absenden zu überprüfen. Die Sprache wurde seitdem weiterentwickelt und ist mittlerweile fester Bestandteil aller gängigen Browsern.

Nachteil der Sprache: Die Interpretation kann von Browser zu Browser abweichen. Aus Sicherheitsgründen kann die Ausführung bei manchen Webseitenbesuchern sogar ganz deaktiviert sein. Obwohl dies technisch durchaus möglich ist, eignet sich Javascript nicht für komplexe und essentiell wichtige Anforderungen. Die Sprache dient vielmehr einer optionale Unterstützung des Besuchers beim Bedienen der Webseite. Weiterhin ist der Quellcode im Vergleich zu serverseitigen Sprachen für jeden Benutzer einsehbar.

2.4.3 AJAX / Web 2.0

Der Begriff Web 2.0 wurde im Dezember 2003 in der US-Ausgabe "Fast Forward 2010 - The Fate of IT" einem Fachmagazin für IT-Manager, in dem Artikel "2004 - The Year of Web Services" von Eric Knorr erstmals gegenüber einer breiten Öffentlichkeit erwähnt. Mit dem Begriff Web 2.0 ist der Trend zu einer veränderte Nutzung des Internets gemeint. Konkret heißt dass weg von statischen Seiten, deren Inhalt zentral von einzelnen Webentwicklern / Redakteuren erstellt wird, hin zu interaktiven Seiten, deren Inhalt von Benutzern der Seite erstellt oder verändert wird. Gute Beispiele sind Foren, Wikis, Social Plattforms wie Xing, Facebook und andere.

Im Zuge dieser Entwicklung spielt Javascript eine wesentlich Rolle. Erst mit Javascript können interaktive Webanwendungen entwickelt werden, die es dem Besucher ermöglichen schnell und einfach Inhalt der Webseite zu beeinflussen. Dabei sind Parallelen zu klassischen Desktopanwendungen zu erkennen z.B. können Elemente einer Webseite per Drag and Drop verschoben werden.

Ein wesentlicher Grund für den vermehrten Einsatz von Javascript ist die Tatsache, dass Interaktion zwischen Benutzern und Webseite Früher nur durch Neuladen der gesamten Webseite möglich war. Mit Hilfe von „Asynchronous JavaScript and XML“ kurz AJAX, welches eine asynchronen Datenübertragung zwischen dem Webserver und dem Browser ermöglicht, können nun Daten zwischen Server und Browser ausgetauscht werden ohne die komplette Seite neu zu laden. Bei der Rent-A-Jet Angebotssoftware werden beispielsweise beim Hinzufügen von Zwischenstopps die Länge der Gesamtstrecke sowie die Gesamtkosten neu berechnet und angezeigt, ohne die Seite neu zu laden.

2.5 Datenbanken

Bei der Erstellung der Rent-A-Jet Angebotssoftware wurde eine relationale MySQL Datenbank eingesetzt. MySQL ist Open Source und wurde ursprünglich durch das schwedische Unternehmen MySQL AB entwickelt. Seit Februar 2008 wird MySQL durch die Firma Sun Microsystemens weiterentwickelt. Laut Sun ist MySQL mit 65.000 Downloads am Tag die derzeit erfolgreichste Open Source Datenbank weltweit. Ebenfalls konnte laut Hersteller der Abstand zu bekannten, kommerziellen Datenbanksystemen wie MS SQL und Oracle in den vergangenen Jahren weiter verringert werden. MySQL belegt nach MS SQL und Oracle Platz drei der am meisten genutzten Datenbanken. In Folge der Übernahme durch Sun steht ein duales Lizenzmodell mit folgenden Optionen zur Verfügung:

- OpenSource für nicht kommerzielle Nutzung
- Proprietär für kommerzielle Nutzung

Die kommerzielle Nutzung erfordert eine MySQL Enterprise Lizenz welche aktuell in der Basic Variante $599,00 pro Server und Jahr kostet. Die Preise für MySQL Server sind im Vergleich zu alternativen Microsoft SQL Server um ein vielfaches günstiger. Durch den Einsatz von MySQL können somit Kosten in teilweise erheblichen Umfang eingespart werden. An dieser Stelle eine kleine Anmerkung: Trotz dieses Kostenvorteils ist die erstellte Software nicht an eine MySQL Datenbank gebunden. Durch Abändern der Softwarekonfiguration ist eine Anbindung an:

- Oracle
- MS SQL
- Postgres

problemlos möglich. Ggf. müssen bei der Erstellung der Datenbank kleinere Änderungen am SQL-Skript zur Datenbankerstellung vorgenommen werden, da sich die Datenbanksysteme an dieser Stelle leicht unterscheiden. Im operativen Betrieb können alle Datenbanken ohne Programmieraufwendungen eingesetzt werden.

2.6 Frameworks und Librarys

2.6.1 Das CakePHP Framework

Seit PHP in der Version 5 erschienen ist wird erstmals objektorientierte Programmierung unterstützt. Dadurch schließt sich die Lücke zu bekannten Sprachen wie Java. PHP eignet sich zunehmend für komplexe Webanwendungen und findet auch im Unternehmensumfeld immer mehr Zuspruch. Seit der Veröffentlichung von PHP 5 sind immer mehr Frameworks am Markt verfügbar. Dadurch müssen wiederkehrende Aufgaben nicht immer wieder neu programmiert werden, Entwickler können auf fertige und sichere Komponenten eines Frameworks zurückgreifen. Dies spart Kosten und steigert die Qualität der Software insgesamt, erfordert aber eine zum Teil nicht unerhebliche Einarbeitungszeit.

Zu Beginn des Projekts stand die Wahl eines geeigneten PHP Frameworks. Die Auswahl beeinflusst alle weiteren Entwicklungsschritte wesentlich, da ein objektorientierter Ansatz eine andere Softwarearchitektur erfordert als ein prozeduraler Ansatz. Folgende Kriterien wurden bei der Auswahl berücksichtigt:

- Zeitlicher Aufwand für Einarbeitung / Lernkurve
- Bekanntheitsgrad
- Ausgereiftheit
- Dokumentation & Größe der Community
- Model View Controller Architektur
- Keine zusätzlichen Kosten / Open Source

Nach einem Vergleich der bekanntesten Frameworks entschied sich die Entwicklung für CakePHP. Ausführliche Dokumentationen und kurze Einarbeitungszeit waren ausschlaggebend da sich die Frameworks in den anderen Punkten nicht wesentlich unterscheiden.

Ablauf eines HTTP Request:

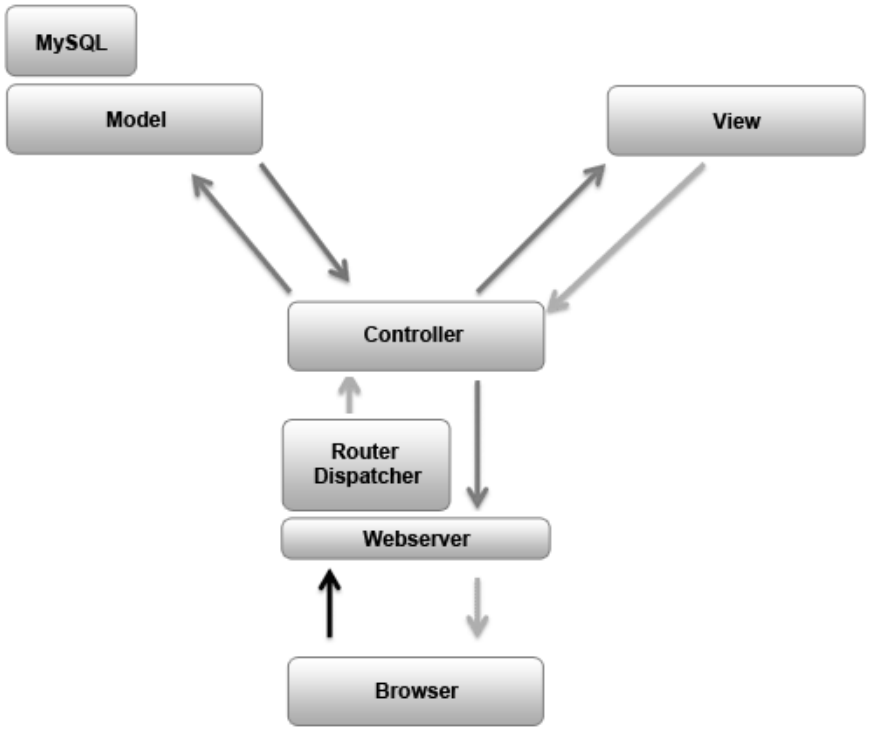


Abbildung 1: Ablauf eines HTTP Requests mit Hilfe von CakePHP

Der Browser sendet zu Beginn einen HTTP Request an den Webserver. Im CakePHP Router und Dispatcher wird der Aufruf analysiert.

Beispiel: http://www.rent-a-jet.de/flugzeuge/view/3

Aus der URL werden automatisch folgende Komponenten extrahiert:

flugzeuge	Controllername bzw. Controller Klasse
view	Methode innerhalb der Klasse, hier Anzeigen der Flugzeugdaten
3	Argument welches in der Methode verfügbar ist, hier die Flugzeugnummer.

Es wird nun die entsprechende Controller Klasse geladen und die beinhaltete Methode aufgerufen. Weiterhin werden die Argumente übergeben.

Der angesprochene Controller ruft im nächsten Schritt das entsprechende Datenmodell auf. Ein Datenmodell liefert Daten aus unterschiedlichen Quellen. Zum Beispiel kann das Datenmodell Einträge aus einer MySQL Datenbank lesen. Wie bereits beschrieben sind an dieser Stelle aber auch andere Datenquellen möglich (MS SQL, Oracle, Webfeeds, lokale XML Dateien usw.) Abschließend werden die Daten an die Controller Klasse übergeben.

Im nächsten Schritt lädt der Controller ein bestimmtes View. Views sind HTML Templates die nun mit den vorhandenen Daten befüllt werden. Das Ergebnis, eine HTML Seite, wird zurück an den Controller übermittelt, welcher die Daten über den Webserver an den Browser sendet.

Innerhalb eines CakePHP Controllers stehen ein Vielzahl von Funktionen, den sogenannten CakePHP Core Components, zur Verfügung. Die Core Components unterstützen den Entwickler unter Anderem beim Senden von E-Mails, Laden und Speichern von Cookies, Authentifizierung und Sessionhandling. Es können eigene Komponenten entwickelt werden. Zum Beispiel eine Komponente „Entfernungsberechnung“. Der modulare Ansatz von CakePHP ist übersichtlich und erleichtert die Erweiterung.

2.6.2 Jquery Library

Jquery ist ein kostenloses und sehr bekanntes JavaScript Framework. Es wurde im Januar 2006 erstmals auf dem BarCamp (NYC) in New York vorgestellt und wird seitdem laufend weiterentwickelt. Das Framework ist unter Anderen bei Google, Dell und Microsoft im Einsatz und deshalb bereits sehr populär.

Jquery bietet folgende Funktionalitäten:

- Schnelle Document Object Model (DOM) Selektion und Manipulation
- Erweitertes Event-System
- Werkzeuge für Effekte und Animationen (z.B. Drag and Drop)
- Ajax Unterstürzung
- Weitere Entwicklungswerkzeuge zur schnelle Erstellung von JavaScript Anwendungen

Wie bereits beschrieben hat JavaScript den Nachteil, dass Quellcode je nach Browsertyp und -version unterschiedlich interpretiert werden kann. Dies kann im schlimmsten Fall dazu führen, dass Besucher einer Webseite Inhalte nicht sehen oder bestimmte Funktionen nicht nutzen können. Hier setzt Jquery an. Ein weiteres wichtiges Feature ist die Cross-Browser Funktionalität. Alle Jquery Komponenten sind auf den gängigsten Browsern getestet. Aufwendige Kompatibilitätstests bei der Entwicklung einer JavaScript Anwendungen entfallen oder können minimiert werden. Dadurch können Kosten gespart und die Produktivität gesteigert werden. Weiterhin kann die Userbility der Anwendung gesteigert werden.

3 Softwareentwicklung

3.1 Vorgehensmodell

Als Vorgehensmodell wurde das SE-Book (Software Engineering Book) der Firma T-Systems genutzt. Das SE-Book basiert auf der theoretischen Grundlage des V-Modells und erweitert das V-Modell um praktische Handreichungen etwa bei der Wahl von Versionsverwaltungssystemen oder Dokumententemplates.

Das SE-Book erlaubt als Methode der Softwareentwicklung sowohl ein klassisches Wasserfallmodell, als auch agile Entwicklungsansätze wie SPICE.

Im Rahmen des Projektes wurde für die Analyse, Entwicklung und Teile des Tests ein agiler Ansatz benutzt, d.h. dass Analyse, Entwicklung und Tests parallel verliefen. Abweichungen gab es dort, wo dies aus praktischen Gründen sinnvoll erschien. Konkret während der initialen Anforderungsanalyse vor Abgabe des Angebots und beim abschließenden Systemtest.

3.2 Modellierung des Softwaresystems

Als Modellierungssprache wurde UML 2.1 gewählt. Dabei gab es eine enge Anlehnung an spezifische Eigenarten des Beratungshauses OOSe, etwa hinsichtlich der Modellierung Fachlicher und Nichtfachlichr Anforderungen.

Modellierungswerkzeug war Magic Draw in der Version 2.1

Das UML-Modell beschreibt vollständig den Bereich der Analyse.

Lediglich als Rumpf sind fachliche Klassen für das MVC-Pattern vorhanden. Der erreichte Stand ist ausreichend für weiterführende Entwicklungsarbeiten in späteren Projekten. Insbesondere lassen sich Zusammenhänge zwischen Controllern und Datenbanken-Tabellen erkennen.

Wesentliche Methoden der Controller wurden als Attribut der Klasse modelliert. Nicht modelliert wurden vererbte Methoden aus dem gewählten CakePHP-Framework.

Der ursprüngliche Ansatz ein vollständigen Modellierung aller Abläufe innerhalb der Software wurde fallengelassen. Es hat sich gezeigt, dass diese Abläufe durch das CakePHP-Framework ausreichend dokumentiert sind. Durch eine fixe Modellierung bestände zudem die Gefahr, dass sich bei zukünftigen Änderungen von CakePHP das Modell nicht mehr korrekt ist.

3.3 IT-Konzept

Das IT-Konzept ist vollständig in der Analysedokumentation und dem Fachkonzept beschrieben.

3.4 Softwarearchitektur

Nach Auftragsannahme wurde entschieden, die Software mit dem PHP Framework CakePHP zu schreiben. Die Softwarearchitektur entspricht vollständig dem Standardverfahren zur Entwicklung mit CakePHP. Zur Gesamtarchitektur sei daher auf die Dokumentation von CakePHP verwiesen.

Es gibt keine Abweichungen zur Standardarchitektur.

Projektspezifische Programmierungen sind im UML-Modell / Analysedokument beschrieben.

3.5 Benutzeroberfläche

Ein weiterer wichtiger Bestandteil der Software ist die Benutzeroberfläche. Sie stellt alle für die Nutzung der Software relevanten Informationen zur Verfügung und bietet Interaktionsmöglichkeiten zwischen Software und Benutzer. Erst durch die Benutzeroberfläche können individuelle Anforderungen des Benutzers mit Hilfe der Anwendungslogik verarbeitet werden.

Grundsätzliche Anforderungen Benutzeroberfläche:

- Einfache Bedienbarkeit (Userbility)
- Leichte Zugänglichkeit (Accessibility)
- Ansprechendes Desgin

Die Erstellung des Designs erfolgte mit Hilfe von Adobe Photoshop. Die einzelnen Designelemente wurden innerhalb der Webseite mit HTML und CSS positioniert. Dabei wurde ein weitgehend tabellenloses Layout verwendet. Dies steigert die Übersichtlichkeit des HTML Codes und reduziert die Webseitengröße insgesamt.

Innerhalb der Benutzeroberfläche kommt das beschriebene Jquery JavaScript Framework zum Einsatz. Zum Beispiel werden Tooltips beim Überfahren von Links und Buttons eingeblendet oder das Ergebnis der Entfernungsberechnung beim Auswählen einer Flugroute angezeigt. Zur Vermeidung von Redundanzen wurde darauf geachtet, dass die Entfernungsberechnung nach wie vor serverseitig erfolgt. Hierbei kommen die beschriebenen Ajax Funtkionen von Jquery zum Einsatz welche durch folgendes Diagramm veranschaulicht werden:

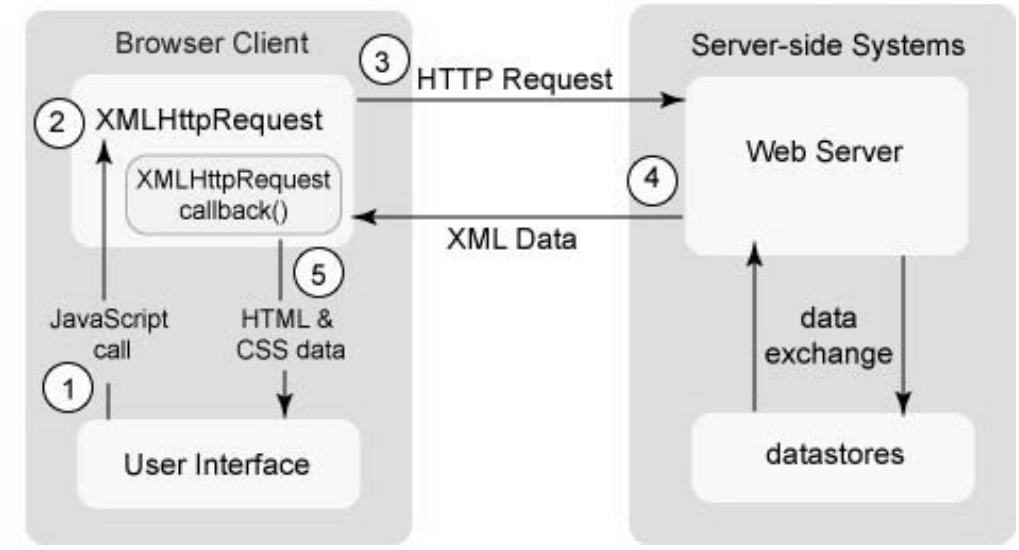


Abbildung 2: Ablauf eines AJAX Requests

Erklärung: Wählt der Benutzer eine Flugstrecke werden die benötigten Daten zur Entfernungsberechnung durch Javascript zusammengestellt und mittels eines XML HTTP Requests an den Webserver übermittelt. Serverseitig berechnet CakePHP die Entfernung. Benötigte Geokoordinaten der Flugplätze bezieht CakePHP dabei direkt aus der Datenbank. Nach der Berechnung wird das Ergebnis als XML File zurück an den Browser gesendet. Das XML File wird vom Browser mit Hilfe von Jquery eingelesen. Die erhaltenen Informationen werden aufbereitet und in das DOM der Webseite integriert. Dies alles geschieht ohne die Webseite neu zu laden und kann beliebig oft wiederholt werden.

Weitere Jquery Funktionen innerhalb des Rent-A-Jet Frontends:

- Laden von Adressdaten bei Auswahl eines Kunden
- Aktualisierung des Formulars „Angebot erstellen“ beim Auswahl des Zeitchartertyps
- Kalender zur Auswahl des Start- und Enddatums
- Anzeigen einer Flugroutenübersicht inkl. Ergebnis der Entfernungs- und Gesamtpreisberechnung
- Aktualisierung der Kosten beim Hinzufügen von zusätzlichen Flugbegleitern
- Aktualisierung der Flugzeugliste bei Überschreitung der maximalen Flugstrecke eines Flugzeugtyps
- Aktualisierung der Flugzeugliste bei Überschreitung der maximalen Personenanzahl eines Flugzeugtyps
- Warnmeldung falls nachträglich Änderungen der Personenanzahl die maximale Personenanzahl des bereits gewählten Flugzeugtyps überschreiten.
- Warnmeldung falls nachträglich hinzugefügte Flugbegleiter die maximale Personenanzahl des bereits gewählten Flugzeugtyps überschreiten.
- Warnmeldung falls nachträglich hinzugefügte Strecken die maximale Reichweite des bereits gewählten Flugzeugtyps überschreiten.

Alle Funktionen zur Interaktion wurden in der Datei

/chartersoftware/app/webroot/js/rentajet.js

gesammelt. Beim Ausführen von Ajax Requests an den Webserver wird eine Ladeanzeige mit Ladeinformationen im Footerbereich angezeigt.

4 Qualitätsmanagement

4.1 Begriffe der Qualitätssicherung

Innerhalb dieser Richtlinie wird das Standard-Vokabular des Software-Engineering verwendet (vgl. IEEE 610) sowie einige spezielle Begrifflichkeiten, die noch genauer erläutert werden.

Prüfgegenstände sind die zugrunde gelegten Betrachtungseinheiten. Zu den Prüfgegenständen bei der Softwareentwicklung gehören z.B. die Anforderungsspezifikation, die Softwarespezifikation, Diagramme, Quellcode, die ausführbare Anwendung und die Testdokumente selbst.

Unterschieden wird grundsätzlich zwischen einem Fehler und einer Ursache. Ein Fehler ist das Abweichen eines berechneten, beobachteten oder gemessenen Werts oder eines Zustands von dem entsprechenden spezifizierten richtigen Wert bzw. Zustand (z.B. die Ausgabe eines falschen Funktionswerts). Die Fehlerursache kann dabei unbekannt sein. Außerdem können mehrere Fehler dieselbe Ursache haben.

Weiter differenzieren wir zwischen einem Defekt und einem Ausfall.

Bei einem Defekt weicht ein Merkmal des Prüfgegenstands von seiner Spezifikation ab. Defekte können, müssen aber nicht als Fehlerursache die Funktionstüchtigkeit des Prüfgegenstands beeinträchtigen (z.B. fehlerhaft realisierte Anforderungen, aber auch mangelhafte Lesbarkeit, Verletzung von Dokumentationsrichtlinien).Ein Ausfall bedeutet, dass der Prüfgegenstand aufgrund eines Defekts nicht mehr in der Lage ist, die geforderte Funktion auszuführen (z.B. Absturz, Livelock- oder Deadlock-Zustand).

Prüfen ist das gezielte Suchen nach Fehlern und Defekten des Prüfgegenstands. Geprüft werden sämtliche Module, die bei der Softwareentwicklung entstehen.

Die Qualitätssicherung ist die Gesamtheit aller die Softwareentwicklung begleitenden Tätigkeiten zur Vermeidung von Defekten, zu denen außer Prüfen auch die Planung und Kontrolle dieser Tätigkeiten gehören.

4.2 Administration der Software

Die einfache Softwareanwendung bietet dem Kunden die Möglichkeit, die Software selbst administrativ zu verwalten. Durch die einfache Handhabung entfällt die Administration seitens von ThinkLogics und reduziert die administrativen Kosten in einem erheblichen Maße.

Bereits im Leistungsumfang enthalten ist das Handbuch, in dem die Verwaltung der Software dokumentiert ist. Weiterhin im Leistungsumfang enthalten ist eine individualisierte Schulung, in dem die administrativen Funktionalitäten der Software erklärt werden.

Im Rahmen von Erweiterungen der Softwarefunktionalitäten oder sonstigen individualisierten Maßnahmen wird ThinkLogics bei Anfrage sehr gerne tätig.

4.3 Test

Test ist ein relevanter Bestandteil der Qualitätssicherung und ohne ein ausführliches Testing kann die vertraglich vereinbarte Qualität nicht realisiert werden.

4.3.1 Warum wird getestet

Mit Testing wird die gewünschte Qualität des Kunden erreicht. Durch intensives Testing können Fehlerquellen ausgeschlossen werden und die Verfügbarkeit wird erhöht.

Durch das rechtzeitige Aufdecken von potentiellen Fehlerquellen können im Vorfeld Fehler vermieden werden und somit Kostensenkungspotentiale erreicht werden. Denn entdeckte Fehlerquellen in der sogenannten „Go-Live“ Phase sind nicht nur aus der finanziellen Perspektive bedenklich, sondern schädigen das Image des Softwaredienstleisters.

Leistungen gelten als nicht erbracht und verzögern das Projekt in einem erheblichen Umfang. Ferner können durch fehlerhafte Softwarefunktionalitäten Sicherheitsprobleme auftreten, die einen enormen Schaden anrichten können

4.3.2 Vorgehensweise beim testen

Der Grundgedanke des Testen wird in 6 W Fragen beantwortet. Die Fragen lauten:

- warum
- was
- wer
- wozu
- wie
- wann

sollen wir testen?

Wenn wir diese Fragen alle beantwortet haben ist der Grundstein für ein erfolgreiches Testing gelegt. Zu Beginn eines Projektes wird bei Think Logics das Qualitätssicherungsteam zusammengestellt, welches dann den weiteren Vorgang des Testings einleitet, überwacht und durchführt.

Der 1. Arbeitsschritt des Qualitätssicherungsteam besteht in der Festlegung von Terminen sowie von Testszenarien und der Einrichtung einer Testumgebung, die den Anforderungen des Kunden gerecht wird.

Jedes Testszenario wird dokumentiert um entstandene Fehler nachzuvollziehen. Bei Fertigstellung von Modulen werden Testingszenarien absolviert und durch das Qualitätssicherungsteam geprüft. Mögliche Fehlerquellen und Defects werden zeitnah durch das Trackingsystem direkt ans Entwicklerteam gesendet.

4.3.3 Was wird getestet

Um die Software zu Testen werden verschiedene Softwarefunktionalitäten getestet. Innerhalb der Software wird die Verfügbarkeit, Funktionalität, Belastbarkeit sowie die Sicherheit getestet. Des weiteren wird das Betriebssystem und die benötigen Programme auf die gleichen Kriterien geprüft wie die Rent-A-Jet Software um eine sichere Plattform für die Rent-A-Jet Software zu bieten.

4.3.4 Wer testet?

Das Testing wird durch verschiedene Personen durchgeführt. Jedes Testszenario wird von zwei verschiedenen Testern durchgeführt um Fehler auszuschließen. Die Hauptverantwortung für das Testing liegt beim Leiter des Qualitätssicherungsteams, der auch entscheidet wer welches Modul prüft.

Bei Fertigstellung der Software in der Endphase, werden Mitarbeiter des Kunden eingeladen um gemeinsam diverse Testszenarien zu absolvieren.

Durch diese Testszenarien besteht die Möglichkeit, noch vor der Auslieferung der Software, ein Feedback vom Kunden zu erhalten und um gegebenenfalls Korrekturen an der Software durchzuführen.

4.3.5 Wann wird getestet?

Das Testing beginnt mit dem Start des Projektes. Am Anfang wird die zusätzlich benötigte Software getestet, wie das Server Betriebssystem sowie XAMPP.

Wenn die ersten Module von den Entwicklern freigegeben werden, werden diese direkt in der Testumgebung geprüft und ein Testbericht an die Entwickler weitergeleitet.

Nach erreichen des Beta Status der Software wird der erste große Testlauf gestartet. Die Ergebnisse werden an die Entwickler weitergeleitet und die Fehler aus der Software entfernt.

Nach der Fehlerkorrektur wird ein weiterer Testlauf mit Unterstützung der Kunden Mitarbeiter durchgeführt. Fehler und Wünsche werden wieder an die Entwickler weitergeleitet. Wenn diese alle entfernt wurden und die Dokumente, die der Kunde erhalten soll, geschrieben sind wird die Software ausgeliefert.

4.4 Standards

Standards werden innerhalb der Software eingesetzt um eine Einarbeitung in den Quellcode schneller zu ermöglichen.Des weiteren werden Standards im Bereich des Frontendes verwendet um eine einheitliche Darstellung zu gewährleisten.

4.4.1 Coding Standards

Im Bereich des Quellcodes werden verschiedene Standards verwendet. Sinn und Zweck davon ist es, dass der Quellcode leicht verständlich ist und dadurch eine Erweiterbarkeit gegeben ist.

Wie Martin Fowler in seinem Buch „Refactoring - Improving the design of existing code“ auf Seite 15 schrieb: „Any fool can write code that a computer can understand. Good programmers write code that humans can understand.“ (Auf Deutsch: Jeder Dussel kann Code schreiben den ein Computer versteht, gute Porgrammierer schreiben Code den Menschen verstehen.) ist für eine erweiterbare Software Voraussetzung, dass Coding Standards eingehalten werden.

Der komplette Quellcode ist kommentiert für eine bessere Verständlichkeit. Dabei wurde darauf geachtet das jede Zeile mit einem Kommentar versehen ist.

Des weiteren werden so wenig wie möglich Abkürzungen innerhalb der Software sowie des Quellcodes verwende. Sollten doch welche zum Einsatz kommen, dann werden keine zweideutigen Abkürzungen verwendet um Verwechselungen auszuschließen.

Des weiteren wird mit Einrückungen gearbeitet. Jede Funktion wird eingerückt,damit es sowohl übersichtlicher ist als auch eine Verfolgung der einzelnen Funktionen in verschachtelten Anweisungen problemlos möglich ist.

```
class EntfernungenController extends AppController
{
    public $uses = array('Flugplatz');

    public function berechnen($start =0, $ziel=0)
    {
            $this->Flugplatz->order = 'Flugplatz.name ASC';
            $this->set('Flugplaetze',$this->Flugplatz->find('list'));
            $this->set('zeitzonenliste', timezone_identifiers_list
());

            if ($start!=null && $ziel != null)
            {
                  //Noch keine Daten ausgewÃ¤hlt
                  $this->data['Entfernung']['start_id'] = $start;
                  $this->data['Entfernung']['ziel_id'] = $ziel;
            }else{
                  //bereits Daten ausgewÃ¤hlt
                  $start = $this->data['Entfernung']['start_id'];
                  $ziel = $this->data['Entfernung']['ziel_id'];
            }
            $this->data['Entfernung']['distance'] = $this->
Kalkulationen->CalcEntfernung($start, $ziel);


    }

}
```

Abbildung 3: Codebeispiel

4.4.2 Formate

Bei Formaten handelt es sich um festgelegte Farben, Schriftarten, Ausgabeformate und Button Größen innerhalb des Frontends der Rent-A-Jet Software.

Innerhalb der Rent-A-Jet Software wird als Schriftart Arial verwendet welche für Überschriften in der Größe 15 und in Texten in Größe 11 umgesetzt wurde.

Die Button Größe ist auf 150 px und mit der Farbe #EF5A00 festgelegt wurden.

Bei den Ausgabeformaten werden stark frequentierte Formate eingesetzt. Zum Einsatz kommen Portable Document Format (PDF), Character Separated Value (CSV) und Extensible Hypertext Markup Language (XML).

4.5 Modularisierung

Die Modularisierung bietet den Vorteil, dass während der Entwicklungsphase einzelne Module getestet werden können und nicht erst nach Fertigstellung der Software.

Durch diesen Prozess ist es möglich schon während der Entwicklung mit dem Testing zu beginnen. Durch das parallele Testing wird der gesamte Entwicklungszeitraum verringert vor raus wieder finanzielle Vorteile für den Kunden entstehen.

Durch die Modularisierung ist eine Erweiterbarkeit der Software gegeben da nur neue Module hinzugefügt werden müssen. Neue Module können in die Rent-A-Jet Software eingepflegt werden ohne das eine komplette Überarbeitung nötig ist.

5 Testing

Um die Funktionalität einer Software zu messen werden Softwaretest während der Entwicklung durchgeführt. Ein Softwaretest wird als analytische Maßnahme definiert und fungiert somit als Prüfgegenstand. In der Informatik ist daher die Durchführung von Softwaretest ein wesentlicher Bestandteil der Softwareentwicklung.

5.1 Ziele

Ziel der Durchführung von Testszenarien ist das Aufspüren und Ermitteln von Fehlerquellen, nicht jedoch die Fehlerbehebung. In der Praxis gilt jedoch die weit verbreitete Annahme, dass ein erfolgreiches Testszenario d.h. ein Test ohne gefundene Fehler als ein Beleg für ein technisch einwandfreies Programm fungiert. Diese Annahme ist jedoch falsch. Vielmehr tragen Softwaretest dazu bei, dass Fehlerquellen drastisch minimiert werden und die Softwarefunktionalitäten verifiziert werden.

In unserem Projekt für das Unternehmen Rent-A-Jet hat das Unternehmen ThinkLogics im Rahmen des Testing folgende Zielidentifikationen festgelegt:

- Der Fehler muss nachvollziehbar sein
- Der Fehler muss rechtzeitig vor der Einführung der Software ermittelt werden
- Die Behebung des Fehlers muss sich rentieren

Demzufolge korrelierten die Ziele mit organisatorischen Maßnahmen. Der Projektplan wurde trotz des sehr engen Zeitplans strikt eingehalten. Durch die Einhaltung des Projektplans innerhalb der vereinbarten Zeit hat ThinkLogics die Lieferung mitsamt der Funktionalitäten erbracht und diese sogar noch um weitere Funktionalitäten erweitert. Diese werden nachstehend in Kapitel 6.2 beschrieben.

Das Unternehmen ThinkLogics war darauf bedacht, nicht nur Funktionalitäten zu erweitern um den Anforderungen unseres Kunden gerecht zu werden, sondern führte intensive Testphasen durch um Fehlerfolgekosten auf ein Minimum zu reduzieren. Das Unternehmen Rent-A-Jet AG hat eine Software erworben, die den höchsten Qualitätsanforderungen entspricht.

Nachstehend werden die (Neben-) Ziele des Testings zusammengefasst:

- Kürzere Entwicklungszeiten
- Bessere Planbarkeit (Termin & Kosten)
- Bessere Aufgabenverteilung
- Steigerung der Qualität
- Reduktion von Fehlerfolgekosten

Resultierend daraus werden dem Anwender bzw. Benutzer einer Software die erwarteten Funktionalitäten zum vereinbarten Zeitpunkt geliefert, die im ausreichenden Maße sichergestellt werden.

5.2 Planung

Für die Durchführung von Testing Szenarien wurden diverse Hilfsmittel in Anspruch genommen. Das Testing Team war in erster Linie für die Durchführung von Funktionstest verantwortlich. Die sogenannten Funktionstest (auch funktionale Test genannt) bezeichnen die Kontrolle eines Softwareproduktes gegen die vom Kunden gestellten Anforderungen. Im Rahmen dieses funktionalen Test wurde nicht nur die vom Kunden gestellten Anforderungen – also das Sollverhalten der Software geprüft – sondern auch eine entsprechende Gegenprobe absolviert. D.h. mögliche Fehlerquellen in der funktioalen Software- Oberfläche sollten geprüft werden.

Dabei wurde das Testing-Konzept nach den Anforderungen des Kunden erstellt. Die genannten Zielvorgaben bzw. Anforderungen der Software wurden im Rahmen der Angebotserstellung der Rent-A-Jet AG definiert. Demzufolge stellte sich im Vorfeld die Fragestellung:

- Was ist überhaupt zu testen?
- Wie soll getestet werden?

Demzufolge sollten die Workflows folgende Funktionalitäten abbilden, die für die Planbarkeit von Testingszenarien relevant sind und zumindest in der ersten Fragestellung für Klarheit sorgen sollten.

Zur Ermittlung der zu testenden Qualitätsmerkmale wurden die Benutzeranforderungen aus dem Angebot herangezogen, die nachstehend aufgelistet sind:

Workflow Charterflug:

1) Anfrage seitens des Kundenanforderungen
2) Finanzielles Angebot erstellen und verschicken
3) Bei Vorliegen der Antwort
    1. Wenn Antwort positiv, Vertrag vorbereiten
    2. Wenn Antwort negativ, Gründe erfragen → dann Ende
4) Wenn Vertrag unterschrieben zurück
    1. Flugzeug, Crew und Catering bereitstellen
    2. Rechnung erstellen und verschicken, Zahlung verfolgen
    3. Flug durchführen
    4. Kundenzufriedenheit erfragen und abspeichern

Workflow: Kostenverfolgung

1) Zahlungseingang verfolgen
2) Wenn nicht zum Termin
    1. Mahnwesen

Analysen:

1) Kundenzufriedenheit analysieren
2) Ablehnungsgründe der Angebote analysieren
3) Profitabilität der Flugzeuge analysieren

Weiterhin galt es folgende Funktionalitäten in der Testing-Planung zu berücksichtigen, die die Berechnung des Angebots betreffen:

- Anzahl der nötigen Zwischenlandungen (pro Landung: Erhöhung der Charterzeit um 45 Min)
- Kosten (Anteil an Fixkosten plus Anteil an Personalkosten plus hourly cost*Flugzeit)

Weitere Testing Szenarien ergaben sich in der zusätzlichen Softwarefunktionalität:

- Ausgabe des Angebotes als Brief auf Word oder PDF oder direkt als e-mail an den Interessenten (mit Bild des Flugzeuges)
- Ausgabe des Vertrages (ohne Bild)
- Ausgabe der Rechnung

Im Rahmen der Planungsphase wurde der zeitliche Aufwand für den Testing Bereich geschätzt. Dabei galt es genügend Freiräume zu schaffen für eventuelle Änderungen oder für Eintreten von unvorhergesehenen Komplikationen. Ein genereller Testablauf wird in der Testpraxis wie folgt dargestellt:

Bei der Durchführung wurden sämtliche Testszenarien entsprechend protokolliert. Diese werden nachstehend im Kapitel Resultate und Dokumentationen erläutert.

5.3 Hilfsinstrumentarien

Im Rahmen der Testingphase wurden Hilfsinstrumentarien eingesetzt. Im genannten Projekt wurde das Tracking System von Berlios verwendet. Das System diente nicht nur als eine effektive Kommunikationsplattform zwischen dem Entwickler- und Testingteam, sondern fungierte auch als Informations- bzw. Dokumentationsquelle. Sämtliche Daten bzw. Fehlerquellen sind in diesem Trackingsystem hinterlegt. Mit diesem Vorgehen werden Tests anschließend in Protokollen dokumentiert. Ziel dieser Dokumentationshinterlegung ist die Sammlung sämtlicher Fehlerquellen, so wird gewährleistet, dass Fehler nicht verloren gehen oder mehrfach korrigiert werden.

Die nachstehende Abbildung zeigt, welche Daten für die Entwicklung relevant sind, damit entsprechende Fehler behoben werden können.

Submitter	Name der Testers, der Fehler entdeckt hat
Priority	Priorität des Defects
Category	Entsprechende Einteilung in Backend, GUI, Handbuch
Assigned to	Name des Entwicklers oder Verantwortlichen, der für Fehler zuständig ist

Summary	Betreffszeile
Status	Status des zu bearbeiteten Defects
Add Comment	Eingabe der Fehlerbeschreibung

Abbildung 4: Bugtracking Formular

5.4 Erzielte Resultate und Dokumentation

Das Testergebnis sollte in geeigneter Form dokumentiert werden, damit Fehler für den Kunden auch zum späteren Zeitpunkt nachvollziehbar sind, diese sind im entsprechenden Testbericht dokumentiert. Im Rahmen der Testing Aktiväten wurden nicht sämtliche Testing Aktiväten dokumentiert, sondern lediglich fehlerhafte Testszenarien. Das im Anhang beigefügte Dokument beinhaltet Fehlermeldungen mit allgemeinen Angaben, die bereits im vorherigen Abschnitt geschildert sind.

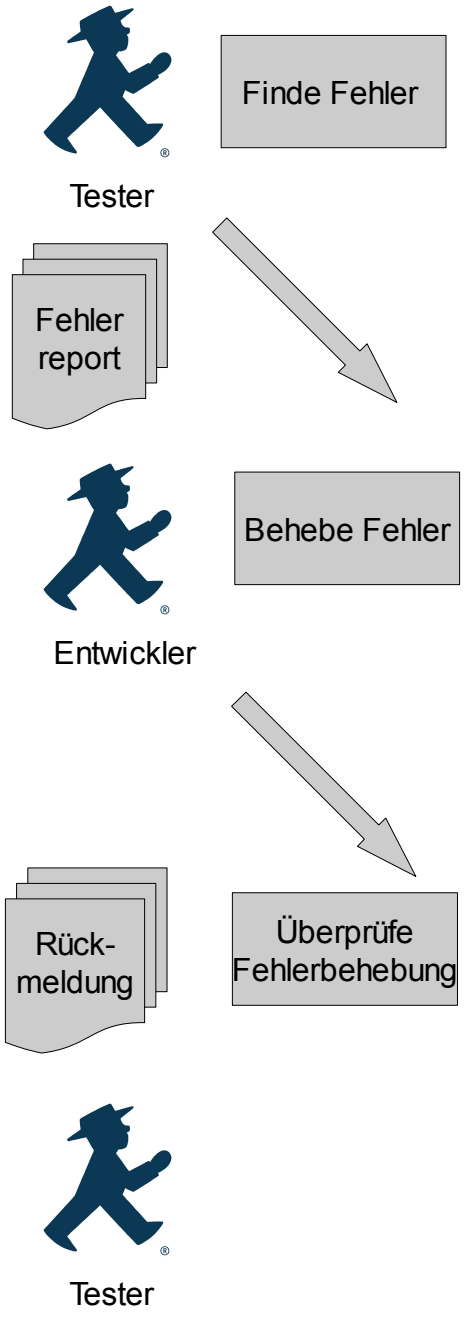
Die angehängte Dokumentation beinhaltet die „drei-Stufen-Theorie“, die den Testablauf in verständlicher Weise darstellen sollen:

1) Planung – Worum geht es eigentlich?
2) Durchführung – Wie wurde getestet?
3) Auswertung – Was wäre das korrekte Ergebnis?

Im Rahmen der Testausführung wurden bekannte fehlerhafte Testszenarien wiederholt getestet Dies war insbesondere in der Phase des Regressionstest der Fall. Der Regressionstest hatte zum Ziel, bereits bekannte fehlerhafte Testszenarien zu wiederholen. Dies war insbesondere dann der Fall, wenn die Software z.B. überarbeitet wurde. Die wiederholte Ausführung von Testszenarien stellte sicher, dass

- der Fehler richtig korrigiert wurde
- der richtige Fehler korrigiert wurde

Die nachfolgende Abbildung stellt den gesamten Testablauf in vereinfachter Form dar:



6 Weiterentwicklung

6.1 Fazit

Alle funktionalen und nicht funktionalen Anforderungen wurden umgesetzt. Insbesondere wurde auf eine leichte Erweiterbarkeit und vollständige Dokumentation geachtet.

Bei zukünftigen Erweiterungen muss ggf. Aufwand für die Einarbeitung in CakePHP beachtet werden. In der Annahme auf vorhandene Kenntnisse in der UML-Modellierung und Programmierung mit PHP wird der Aufwand mit ca. 10 Tagen pro Entwickler gerechnet. Nach dieser Zeit sind einfache Erweiterungen (etwa zusätzliche Datenfelder) innerhalb weniger Stunden möglich.

6.2 Angebotssoftware

6.2.1 Vergleich Soll/Ist Zustand

Analyse – Use Cases

Basierend auf der Leistungsbeschreibung des Kunden erfolgte eine Analyse. Die Analyse ist vollständig. Primäres Dokument der Analyse ist jedoch das textorientierte Analysedokument. Alle Use Cases sind grafisch im UML-Modell hinterlegt.

Analyse – Datenbank

Die Datenbank ist vollständig im UML-Modell beschrieben.

Entwicklung / Test

Die Entwicklung ist vollständig abgeschlossen. Alle Features wurden implementiert. Ebenfalls abgeschlossen ist der Test. Weitere Arbeiten sind nicht notwendig.

Dokumentation

Die Dokumentation ist vollständig. Weitere Arbeiten sind nicht notwendig.

6.2.2 Hinweise zur Weiterentwicklung

Analyse – Use Cases

Sinnvoll für eine Modellbasierende Weiterentwicklung (MDA, MDSD) wäre es, wenn der jetzige Inhalt des Analysedokuments in das Modell überführt wird. Dazu müssten über Stereotypen Tagged Values eingeführt werden.

Eine Dokumentation von Workflows als Aktivitätsdiagramm ist wahrscheinlich nicht sinnvoll. Gegenüber der Textnotation ergeben sich aufgrund des ausgefeilten MVC-Ansatzes von CakePHP keine Vorteile mehr ergeben.

Geschätzer Aufwand: 5 Personentage für eine Übernahme der Text aus dem Dokument in das Modell.

Analyse – Datenbank

Sinnvoll wäre es, eine geeignete UML-Präsentation für Beschreibungen des Modell-Anteils im MVC-Pattern zu finden. Insbesondere hinsichtlich erlaubter/nicht erlaubter Werte. Lösungsansatz wären zum Beispiel OCL oder ein Tagged Value über einen speziellen Stereotypen. Alle Modellbeschreibungen im Sourcecode sowie die Datenbankstruktur ließe sich dann per MDA generieren.

Geschätzter Aufwand: 15 Personentage für die Beschreibung (5 Tage) im Modell sowie Erzeugung eines Generators (10 Tage) in OpenArchitectureWare

6.3 Mögliche Weiterentwicklungen

6.3.1 MDA / MDSD

Während Analyse und Entwicklung wurde auf die Möglichkeit eines zukünftigen MDA/MDSD Ansatzes geachtet.

Basierend auf den vorliegenden Erfahrungen sowie der konsequenten Nutzung des MVC-Patterns wird die Eignung wie folgt eingeschätzt:

Views:	Views sollten sich nahezu vollständig per MDA erzeugen lassen. Ausgenommen sind davon allerdings Sonderviews für Druck-Dokumente (PDF) sowie Statistiken (im wesentlichen XML-basierend)
Model:	Datenmodelle sollten sich vollständig per MDA generieren lassen
Kalkulation	Für Berechnungs-Algorithmen ist ein MDA-Ansatz unrealistisch. Bedingt durch die Komplexität der Berechnungsverfahren würde eine UML-Modellierung unverhältnismäßig aufwendig werden.
Controller	Eine Generierung von Controllern ist denkbar. Dabei muss bei der Generierung auf eine Vererbbarkeit von generierten Controllern geachtet werden. Gelegentlich enthalten Controller ein hohes Maß an Komplexität. Alternativ ist es auch möglich die Komplexität in eigene fachliche Klassen auszulagern und per Composit-Pattern zu nutzen.

Insgesamt sollte es möglich sein, ca. 80% des gesamten Sourcecodes mit vertretbarem Aufwand generativ zu erzeugen. Als vertretbar wird dabei ein maximal gleich hoher Aufwand zu einer manuellen Entwicklung angesehen.

6.3.2 Customer Relationship Management (CRM)

Im Rahmen der Abschlusspräsentation wurde das Konzept einer Customer Relationship Management Software vorgestellt. Gemäß der informellen Vereinbarung befindet sich eine Leistungsbeschreibung in Arbeit und wird parallel zum vorliegenden Endbericht überreicht werden.

Der Aufwand wird im Augenblick mit ca. 1000 Personentagen geschätzt.

6.3.3 Internationalisierung

Hinsichtlich der möglichen Eröffnung internationaler Kontaktbüros ist eine Mehrsprachigkeit der Software sinnvoll. Die Möglichkeit die gesamte GUI mehrsprachig zu machen ist bereits vorgesehen. Diskutiert werden müssten Aufwände für mehrsprachige Datenbankinhalte, etwa Namen/Beschreibungen von Reports.

Aufwand für die Mehrsprachigkeit der GUI: 5 Personentage für einen Dolmetscher, 3 Personentage für fachlichen Support des Dolmetschers

Aufwand für eine Mehrsprachigkeit der Datenbank: 400 Personentage. Es ist ein kompletter Umbau der Datenbank notwendig.

7 Zusammenfassung

Mit dem vorliegenden Bericht wurde die Softwarelösung im Einzelnen dargestellt. Gemäß Helmut Balzert ist unter der Softwaretechnik eine „zielorientierte Bereitstellung und systematische Verwendung von Prinzipien, Methoden und Werkzeugen für die arbeitsteilige, ingenieurmäßige Entwicklung und Anwendung von umfangreichen Softwaresystemen“ zu verstehen. Nach dieser Interpretation umfasst die Softwaretechnik eine Vielzahl von Teilgebieten, wie das Entwickeln der Software sowie das Betreiben der Softwaretechnik. Im Anbetracht dieser Tatsache ist es daher erforderlich, dass Fehler frühzeitig entdeckt werden, idealerweise noch während der Entwicklungsphase um Kostensenkungspotentiale optimal ausnutzen zu können.

Gemäß der DIN 69901 Norm ist ein Projekt „ein Vorhaben, bei dem innerhalb einer definierten Zeitspanne ein definiertes Ziel erreicht werden soll, und sich dadurch auszeichnet, dass es im Wesentlichen ein einmaliges Vorhaben ist“. Im Zusammenhang mit dieser Norm, hatte das Projektmanagement die herausfordernde Aufgabe nicht nur die einzelnen Projektphasen, sondern auch die Projektinhalte klar und verständlich zu definieren. Abgestimmte Tätigkeiten mit einem Anfangs- und Endtermin mussten festgesetzt werden und die vereinbarte Leistung innerhalb des Zeitrahmens erbracht werden, um Abweichungen oder möglichke Konflikte frühzeitig zu vermeiden.

Konflikte innerhalb der Projektphase können das gesamte Projekt verzögern. Daher ist es absolut erforderlich gewesen, dass sowohl auf der Kundenseite als auch auf der Entwicklerseite eine gemeinsame Sprache gesprochen wird. Um diesen vorzubeugen haben wir nicht nur Meilensteine festgelegt, sondern auch die Unified Modelling Language (UML) verwendet. Eine übersichtliche und vereinheitlichte Darstellung von Ergebnissen der Analyse- und Designphase wurde auf diesem Wege ermöglicht.

Die Vereinheitlichung von Standards war ein wesentlicher Bestandteil des Entwicklungsprozesses und trug maßgeblich zum Erfolg der Rent-A-Jet AG Software bei. Auch innerhalb der Softwarearchitektur hat sich ThinkLogics bewusst für die Vereinheitlichung von Webstandards entschieden, um eine Plattformunabhängigkeit zu ermöglichen.

Im Rahmen der Softwareerstellung spielte für ThinkLogics nicht nur die Vereinheitlichung von Standards eine Rolle, sondern auch die Reduzierung von komplexen Anwendungsfällen. Dies beginnt bereits damit, dass eine leicht erlernbare Scriptsprache für ThinkLogics im Vordergrund stand. Aufgrund dieser Überlegungen entschied sich ThinkLogics für PHP.

Die Vereinheitlichung von Standards sowie die Anwendung einer einfachen Scriptsprache trug maßgeblich zum Erfolg bei. Weiterhin ist zu berücksichtigen, dass durch die leichte Erlernbarkeit der Sprache mit geringem Fachwissen die Softwarefunktionalitäten entsprechend kostengünstig erweitert werden können.

Mit der Software wurde eine Lösung entwickelt, die individuell auf die Bedürfnisse der Rent-A-Jet AG konzipiert wurde. Eine leichte Erweiterbarkeit der Softwarefunktionalität ermöglicht Rent-A-Jet AG künftige Entwicklungsprojekte schneller voranzutreiben. Dies bedeutet im Endeffekt, dass die Rent-A-Jet AG nicht nur eine individualisierte Softwareanwendung erhalten hat, sondern in einem erheblich kürzeren Zeitumfang agieren kann und somit schneller am Markt handelt kann als seine Konkurrenten.

8 Anhang

8.1 Autorenverzeichnis

Timm Vollmer	Kapitel 4, Handbuch
Andrey Behrens	Kapitel 3 (außer 3.5), 6
Frederick Geist	Kapitel 2, 3.5
Daniel Böhm	Kapitel 1, 5, 7, Testbericht
Sonstiges	Das Kapitel 8 (Anhang) wurde gemeinschaftlich erstellt.

8.2 Abbildungsverzeichnis

8.3 Quellenverzeichnis

http://www.apachefriends.org/de/xampp-windows.html#628

http://de.wikipedia.org/wiki/Vorgehensmodell_(Software)

http://www.projekthandbuch.de/

http://de.wikipedia.org/wiki/Php

http://de.wikipedia.org/wiki/Ajax_(Programmierung)

http://de.wikipedia.org/wiki/Javascript

http://globalspecials.sun.com/store/mysql/DisplayHomePage/

http://de.wikipedia.org/wiki/Mysql

http://jquery.com/

http://www2.informatik.hu-berlin.de/~hs/Lehre/2004-WS_SWQS/20041103_Testen.pdf

ftp://ftp.fernuni-hagen.de/pub/fachb/inf/pri3/papers/winter/diss.pdf

http://cakephp.org/

http://de.wikipedia.org/wiki/CakePHP

http://de.wikipedia.org/wiki/Framework

8.4 Mit geltende Unterlagen

Zur Vermeidung von Redundanzen wird an verschiedenen Stellen auf weitere Dokumente verwiesen. Hier eine vollständige Aufstellung aller mitzuliefernden Dokumente.

- **UML-Modell**

  UML-Modell als Grundlage der Analyse in der Datei MagicDrawUmlModell.mdzip. Die Datei ist eine Datei für Magic Draw in der Version 16 und benötigt mindestens die Standard-Version von Magic Draw

- **Anforderungsanalyse 1**

  Enthält die Analyse des Lastenhefts als Grundlage des Angebots

- **Angebot**

  Enthalten ist das Angebot, wie es bereits vorliegt.

- **Anforderungsanalyse 2**

  Enthält die vollständige Analyse als Grundlage von Fachkonzept und Entwicklung.

- **Mathematisches**

  Modell In der Datei Analyse/berechnungen.mw (Maple12-Format) bzw. Analyse/berechnungen.pdf (Ausgabe der Maple-Datei als PDF) wurden die verwendeten mathematischen Modelle zur Preis- und Entfernungsberechnung dokumentiert.

- **Handbuch**

  Anleitung zur Installation, Bedienung und Wartung der Rent-A-Jet Softwareanwendung

- Leistungsbeschreibung der Firma Rent-A-Jet als Grundlage des Angebots

- **Endbericht**

  Vorliegendes Dokument in elektronischer Form

- **Software.zip**

  Das komplette Sourcecode-Verzeichnis, einschließlich einer getesteten CakePHP-Referenzversion

8.5 Abkürzungsverzeichnis

bzw.	beziehungsweise
CRM	Customer Relationship Management – Kundenbeziehungsmanagement. Ausrichtung eines Unternehmens auf Kkunden und systematische Gestaltung von Kundenbeziehungsprozessen
CSS	Cascading Style Sheets – deklarative Stylesheet- Sprache für strukturierte Dokumente
DOM	Document Object Methode – Spezifikation einer Schnittstelle für Zugriff auf HTML- oder XML Dokumente
GUI	Graphical User Interface – grafische Benutzeroberfläche
HTML	Hypetext Markup Language – Auszeichnungssprache zur Darstellung von Internetseiten
LAN	Local Area Network - Rechnernetz
OCL	Object Constrains Language. Eine standardisierte Sprache zur Beschreibung von Objektabhängigkeiten und Nebenbedingungen.
PHP	Hypertext Precprocessor – Scriptsprache zur Erstelllung von dynamischen Internetseiten
SQL	Structured Query Language. Bei SQL handelt es sich um eine Abfragesprache für relationale Datenbanken. Im Rahmen vorliegender Software wird SQL zur Erzeugung von Statistiken eingesetzt.
UML	Unified Modelling Language. Eine grafische Notation von geschäftlichen und IT-prozessen. Im Rahmen dieses Projektes für die Analyse eingesetzt.
URL	Uniform Resource Locator – Ort einer Ressource in Computernetzwerken
XAMPP	Zusammenstellung freier Software, welche den Webserver Apache, die Datenbank MySQL und die Scriptsprache PHP beinhaltet
XML	Extended Markup Language – Auszeichnungssprache zur Darstellung hierarchich strukturierter Daten in Form von Textdaten
z.B.	zum Beispiel
PDF	Portable Document Format – Weit verbreitetes Format für Documente
CSV	Character Separated Value – Dateiformat welches den Aufbau von Textdateien beschreibt.
SE-Book	Software Engineering Buch. Beschreibt das T-Systems-Vorgehensmodell für Softwareentwicklungen. Das Modell basiert auf dem V-Models.
MDA	ModelDrivenArchitecture. Softwareentwicklungsmethode, bei der die gesamte Architektur eines Computersystems in Form von UML-Modellen und -Objekten vorliegt. Aus diesem UML-Modell wird im Rahmen Sourcecode generiert. Im Idealfall könnte ein Informationssystem vollständig aus dem Modell generiert werden.
MDSD	Eng verwandt mit MDA. Allerdings werden lediglich Programm- und Klassenstrukturen generiert. Die Logik der Software wird weiterhin manuell geschrieben.