

Inhaltsverzeichnis

1 Einleitung.....	1
1.1 Anonymisierungshinweis.....	1
1.2 Einordnung ins betriebliche Umfeld.....	1
1.3 Projektumfeld.....	1
1.4 Abgrenzung eigener Leistungen von Fremdleistungen.....	2
2 Analyse Ist vs. Soll.....	3
2.1 Ist-Zustand vor Einrichtung Schnittstellenanbindungen.....	3
2.2 Soll-Zustand nach Einrichtung Schnittstellenanbindungen.....	4
2.3 Begründete Abweichungen vom ursprünglichen Projektantrag.....	4
3 Migrationskonzept.....	7
3.1 Zieldefinition.....	7
3.2 Abwägung Alternativen zur Erreichung der Ziele.....	7
3.3 Migrationsschritte & zeitlicher Fahrplan.....	8
4 Durchführung Migration.....	10
4.1 Upgrade Serverumgebung.....	10
4.2 Einrichtung KORONA Modul & Schnittstellen Integration.....	12
5 Tests.....	13
5.1 Verwendeter Testkatalog.....	13
5.2 Auftretene Probleme während der Testläufe.....	13
5.2.1 Inkompatibilität KORONA.modul.....	13
5.2.2 X-Frame-Options Header.....	14
5.2.3 Fehlschlag SSL Handshake beim Rechnungsimport.....	14
5.3 Ergebnis Testläufe.....	14
6 Zusammenfassung.....	15
6.1 Fazit.....	15
6.2 Ausblick.....	15
7 Quellenverzeichnis.....	16
8 Glossar.....	17
9 Anlagenverzeichnis.....	20
9.1 Projektantrag.....	21
9.2 Tabellenverzeichnis.....	24
9.3 Quelltextverzeichnis.....	27
9.4 Testprotokolle.....	30
Folgeschritte.....	33
1.Prüfung Apache Protokolldateien auf KORONA.pos Server System.....	33
2.Prüfung Apache Protokolldateien auf Magento Webshop Server.....	33
3.Onlinerecherche bezüglich möglicher Probleme bei der Anzeige externer PHP Skripte via <iframe>.....	34
4.Anpassung Apache Konfiguration auf dem Magento Webshop Server; Auskommentierung der hinterlegten X-Frame-Options.....	34
5.Neustart Apache HTTP Server via service apache2 restart.....	34
6.Wiederholung des Testlaufs.....	34
7.Die geforderten Abnahmekriterien 1-4 wurden jetzt vollständig erfüllt.....	34
8.Aktualisierung Firmen-interne Dokumentation bezüglich der Problematik X-Frame-Options.....	34
Folgeschritte.....	35
Keine Folgeschritte notwendig.....	35
Folgeschritte.....	36
1.Kontrolle Protokolldateien KORONA.pos Server.....	36
2.Rücksprache im Kollegenkreis und gemeinsame Fehlersuche.....	36

3. Erkenntnis, dass der SSL Handshake zwischen SOAP Poller des KORONA.pos Server und dem Magento Webshop nicht erfolgreich durchgeführt werden kann.....	36
4. Verwendung des Mozilla SSL Configuration Generator zur Erstellung einer geänderte Apache SSL Konfiguration.....	36
5. Hinterlegung der geänderten Apache SSL Konfiguration auf dem Magento Webshop Server	37
6. Erneute Durchführung des Testlaufs.....	37
7. Jetzt wurden die Abnahmekriterien 1-3 vollständig erfüllt.....	37
8. Dokumentation der gewonnenen Erkenntnisse im Wiki der Combase AG.....	37
Folgeschritte.....	38
Keine Folgeschritte notwendig.....	38
Folgeschritte.....	40
1. Prüfung des Konfiguration des Zahlungsmoduls im Magento Admin Backend.....	40
2. Rücksprache mit der Projektleiterin bei der Freizeitpark GmbH & Co. KG, Frau Köppke..	40
3. Änderung der hinterlegten Rücksprungadresse im Backend der Sparkassen Internetkasse durch Frau Köppke.....	40
4. Erneute Durchführung des Testablaufs.....	40
5. Jetzt wurden die Abnahmekriterien 1-4 vollständig erfüllt.....	40
6. Dokumentation der gewonnenen Erkenntnisse bezüglich der Hinterlegung der Rücksprung URL des Zahlungsanbieters im Wiki der Combase AG.....	40
9.5 Kundendokumentation.....	41
9.6 Datenblatt KORONA Integration für Magento-basierte Webshops.....	42
KORONA.pos Server Integration.....	42
KORONA.modul für Magento Webshops.....	43
KORONA.resource Integration.....	43

1 Einleitung

1.1 Anonymisierungshinweis

Die in dieser Projektarbeit erwähnten Geschäftspartner bzw. externen Ansprechpartner wurden aus Gründen des Datenschutzes sowie zum Schutz der Geschäftsgeheimnisse der Combase AG anonymisiert.

Lediglich die Bezeichnungen der Produkte der KORONA Softwarefamilie sowie die Namen der unternehmenseigenen Mitarbeiter, welche für den Projektkontext relevant sind, wurden nicht verfremdet.

1.2 Einordnung ins betriebliche Umfeld

Bei der Combase AG handelt es sich um ein im Jahr 1994 in Dresden gegründetes IT Unternehmen. Der Fokus des Unternehmens liegt dabei auf der Entwicklung von Client/Server-basierten Kassen- und Abrechnungslösungen. Weiterhin bietet die Combase AG speziell für die musealen Bereich sowie für Freizeitparks eigene Softwarelösungen für die Themenbereiche Ticketing sowie Einlassmanagement an. Diese Softwarelösungen werden unter den Produktbezeichnungen KORONA.pos Client (Client-basierte Kassensoftware), KORONA.pos Server (Server-basiertes Administrationsbackend), KORONA.entry (Einlassmanagement) bzw. KORONA.resource (Eventmanagement) vertrieben.

Im Jahr 2009 entstand im Zuge einer – durch das Unternehmen betreuten – Diplomarbeit eine eigene Java-basierte Webshop Lösung, welche bereits eine umfassende Integration in die Softwarelandschaft der KORONA Produkte bot und im Verlauf der nächsten drei Jahre stetig weiterentwickelt wurde. Ab Ende des Jahres 2012 wurden Bestrebungen vorangetrieben, die selbst entwickelte KORONA.webshop Software durch eine etablierte und umfangreiche Webshoplösung eines Drittanbieters abzulösen und diese um entsprechende Schnittstellen zur Integration in die KORONA Produktfamilie zu erweitern.

In Folge dieser Bestrebungen wurde ein Team aus Mitarbeitern der Fachbereiche Webentwicklung, Systemintegration sowie Mediengestaltung gebildet, um die Entwicklung dieses neuen Softwareprodukts voranzutreiben. Der Verfasser dieser Arbeit war dabei als Systemintegrator Teil des gebildeten Projektteams. Die Wahl auf die zukünftig zu nutzende Webshopsoftware fiel nach einem internen Evaluationsprozess auf eine der marktführenden E-Commerce Lösungen: die Magento Community Edition Webshop Software.

1.3 Projektumfeld

Ende des Jahres 2014 intensivierten sich die Gespräche zwischen der Combase AG und der Freizeitpark GmbH & Co. KG bezüglich einer möglichen Einführung der KORONA Software-Lösungen

beim Kunden. In Folge dessen wurde vereinbart, dass die Combase AG die notwendige Hard- und Software bereitstellt, sodass zum Saisonstart der Freizeitpark GmbH & Co. KG im April 2015 sowohl der Kassenbetrieb, das Eventmanagement als auch das Einlassmanagement über Lösungen der KORONA Produktfamilie abgewickelt werden können. Darüber hinaus kamen die Geschäftspartner überein, dass möglichst zeitnah nach Saisonstart ebenfalls ein Relaunch des – bereits bestehenden, Magento-basierten – Webshops des Kunden stattfinden sollte. Bezüglich der Überarbeitung des Webshops war die klare Zielstellung, dass für diesen zukünftig entsprechende Schnittstellen zur Integration mit den anderen eingeführten KORONA Produkten zur Verfügung stehen und genutzt werden sollten.

Auf Grund bereits bestehender Geschäftsbeziehungen der Freizeitpark GmbH & Co. KG waren an dem geplanten Webshop Relaunch insgesamt vier verschiedene Unternehmen mit unterschiedlichen Zuständigkeitsbereichen involviert. Die Combase AG war in diesem Prozess damit betraut, die Anbindung des bereits bestehenden Webshops an die Schnittstellen der neu eingeführten KORONA Produkte beim Kunden durchzuführen.

1.4 Abgrenzung eigener Leistungen von Fremdleistungen

Auf Grund der Tatsache, dass insgesamt drei verschiedene Auftragnehmer an dem Teilprojekt Webshop Relaunch beteiligt waren, wurde besonders darauf geachtet eigene, durch den Verfasser im Rahmen dieser Projektarbeit erbrachte Leistungen von anderen Dienstleistungen zu trennen, welche entweder durch einen der weiteren Auftragnehmer oder aber durch andere Mitarbeiter der Combase AG erbracht wurden.

Fremdleistungen, welche für die Umsetzung des Projektes von Bedeutung waren, werden in Rahmen dieser Projektarbeit klar als solche benannt. Wurden Leistungen im Rahmen des Projekts von anderen Mitarbeitern der Combase AG erbracht, wurde dies im Rahmen der Projektdokumentation ebenfalls kenntlich gemacht. Alle anderen Leistungen, welche in der Dokumentation nicht anders gekennzeichnet wurden, wurden durch den Verfasser der Projektarbeit erbracht.

2 Analyse Ist vs. Soll

2.1 Ist-Zustand vor Einrichtung Schnittstellenanbindungen

Die Freizeitpark GmbH & Co. KG betreibt seit März 2011 einen öffentlichen Webshop auf Basis der Magento Webshop Software. Das über diesen Webshop vertriebene Warensortiment umfasste sowohl Eintrittskarten für den Freizeitpark (Tagestickets, Jahrestickets) als auch eine ganze Reihe verschiedener Merchandiseartikel.

Das Hosting des Webshopservers, sowie des für den Emailversand genutzten Mailservers wurde dabei bisher von der Internet Services GmbH übernommen. Die technische Realisierung des Webshops wurde dabei von der Multimedia Agentur GmbH übernommen, welche ebenfalls kleinere Anpassungen und Designänderungen an der Magento Webshopsoftware durchführte. Diese bisherige Aufgabenteilung sollte auch nach Eintritt der Freizeitpark GmbH & Co. KG in Geschäftsbeziehungen mit der Combase AG beibehalten werden.

Aus Sicht des Kunden gab es vier wesentliche Einschränkungen der bisherigen Shoplösung, welche im Zuge des Webshop Relaunchs beseitigt werden sollten:

1. Doppelte Artikelpflege notwendig: bisher mussten alle im Webshop verkaufbaren Artikel manuell in die Shopsoftware eingepflegt werden. Eine automatisierter Abgleich zwischen dem genutzten Warenwirtschaftssystem und dem Webshop fand nicht statt.
2. Nur physischer Versand von Hardtickets möglich: im Onlineshop gab es bisher nur die Möglichkeit gedruckte Hardtickets zu erwerben. Dies bedeutete für den Betreiber einen erheblichen zusätzlichen Aufwand, weil die Tickets manuell gedruckt, verpackt und an den Kunden versendet werden mussten.
3. Kein kurzfristiges Eventticketing möglich: auf Grund der technischen Gegebenheiten war es bisher nicht möglich, über den Webshop Eventtickets mit nur minimaler Vorlaufzeit zu verkaufen. Solche Tickets mussten bisher von der Multimedia Agentur GmbH mit einem mehrtägigen Vorlauf in den Webshop eingepflegt werden.
4. Sporadisches Auftreten von Rundungsdifferenzen: bei der Anwendung spezieller Rabatte im Webshop kam es sporadisch zum Auftreten sogenannter Rundungsdifferenzen, welche zusätzlichen Aufwand für die nachträgliche Korrektur durch die Abteilung Finanzbuchhaltung der Freizeitpark GmbH & Co. KG erforderten.

Aus Sicht des bisherigen technischen Dienstleisters – der Multimedia Agentur GmbH – fehlte es der bisher eingesetzten Webshoplösung an Zukunftssicherheit, da die eingesetzte Magento CE Version 1.6.2.0 das Ende ihres offiziellen Lebenszyklus erreicht hatte. Damit ging einher, dass auf dem Webserver technische veraltete Versionen der PHP Skriptsprache eingesetzt werden mussten, weil

neuere PHP Versionen offiziell von der bis dahin eingesetzten Magento Version nicht unterstützt wurden. Die Verwendung veralteter PHP Versionen stellte ein erhebliches Sicherheitsrisiko dar.

2.2 Soll-Zustand nach Einrichtung Schnittstellenanbindungen

Aus Sicht des Betreibers – der Freizeitpark GmbH & Co. KG – sollten selbstverständlich alle essentiellen, bisher zur Verfügung stehenden Webshop Funktionalitäten auch nach dem Relaunch weiterhin funktionieren.

Mit dem Webshop Relaunch sollten aus Sicht des Kunden vier zentrale Ziele umgesetzt werden:

1. Automatisierte Synchronisation von Artikeln: Artikel sollten nicht mehr manuell in den Webshop eingepflegt werden, sondern über eine technische Schnittstelle zwischen KORONA.pos Server und dem Magento Webshop synchronisiert werden können. Es war dabei besonders wichtig, dass der Kunde selbst diese Synchronisation jederzeit durchführen kann und nicht auf externe Hilfe angewiesen ist.
2. Kurzfristiges Eventticketing: zukünftig sollten kurzfristige Events selbstständig durch die Mitarbeiter der Freizeitpark GmbH ohne externe Hilfe in den Webshop eingepflegt werden können. Hierfür waren durch die Combase AG sowohl die technischen Erfordernisse (Anbindung Schnittstelle zu KORONA.resource) zu schaffen, als auch Schulungsmaterial für den Kunden zu erstellen und dieses innerhalb einer Kurzschulung zu vermitteln.
3. Einführung Print@Home Ticketing: ein Print@Home System zum Selbstaussdruck von Eintrittskarten sollte eingeführt werden. Besonders wichtig war, dass die Generierung des Tickets direkt nach Bezahlung durch den Kunden automatisiert durchgeführt wird, das Ticket anschließend per Email an den Kunden versandt wird und sofort geprüft und auch entwertet werden kann.
4. Behebung Rundungsdifferenzen: es sollte sichergestellt sein, dass es zukünftig bei der Anwendung von Rabatten im Webshop nicht zu Rundungsdifferenzen kommt, welche bisher nachträglich durch die Buchhaltung korrigiert werden mussten.

2.3 Begründete Abweichungen vom ursprünglichen Projektantrag

Nach der initialen Einreichung des Projektantrags bei der IHK Dresden durch den Verfasser fanden weiterführende Gespräche bezüglich der konkreten Ausgestaltung des Webshop Relaunchs zwischen der Combase AG und der Freizeitpark GmbH & Co. KG statt. In Folge dieser Gespräche wurden die folgenden, zum ursprünglichen geplanten Ablauf abweichenden Absprachen getroffen:

Update Magento Software

Ursprünglich wurde zwischen den Vertragspartnern ins Auge gefasst, dass die Aktualisierung der

Magento Webshop Software auf die neueste zur Verfügung stehende Version durch die Combase AG durchgeführt werden soll. Von diesem Plan wurde durch den Kunden Abstand genommen, nachdem sich die Combase AG dem Kunden empfohlen hatte das Update durch den bisherigen technischen Dienstleister – die Multimedia Agentur GmbH – durchführen zu lassen. Diese Änderung war im Sinne der Combase AG, da bereits Ende 2014 eine Richtungsentscheidung getroffen wurde, dass sich die Combase AG auf ihre Kernkompetenzen als Schnittstellenintegrator konzentrieren möchte und die Betreuung der eigentlichen Webshop Software spezialisierten Webagenturen überlassen will.

Übertragung Webshop Inhalte/Datenmigration

Aufbauend auf die Entscheidung, dass die Aktualisierung der Magento Software durch die Multimedia Agentur GmbH durchgeführt wird, wurde mit dem Kunden vereinbart, dass diese Firma ebenfalls die Migration der Bestandsdaten und Inhalte übernimmt. Dies war aus Sicht der Combase AG eine vorteilhafte Lösung, da hierdurch vermieden wurde, dass es zu Unklarheiten bezüglich der Zuständigkeiten im Falle eines Fehlers nach Durchführung der Aktualisierung kommt. Von Seiten der Combase AG wurde vermutet, dass auf Grund des großen Versionssprungs von Magento CE 1.6.2.0 auf Magento CE 1.9.1.0 erheblicher Aufwand zur Fehlerkorrektur in Bezug auf eingepflegte Inhalte (CMS Seiten, statische Blöcke, Email Templates) nötig werden könnte.

Integration Schnittstellen KORONA Produkte

Ursprünglich sollte die Webshopintegration bezüglich der neu beim Kunden eingeführten KORONA Softwarelösungen bereits im März 2015 erfolgen. Dieser Schritt war auf Basis des bestehenden Magento CE 1.6 Webshops des Kunden geplant. Durch das Auftreten neuer Stakeholder (Marketingabteilung der Freizeitpark GmbH & Co. KG) verzögerte sich aber die geplante Einführung der Integrationsschnittstellen. In Folge dieser Verzögerung kam der Kunde mit der Combase AG überein, dass die Anbindung der Schnittstellen zu den anderen KORONA Produkten erst im April 2015 realisiert und auf Basis der aktualisierten Magento CE 1.9.1.0 Umgebung durchgeführt wird. Dadurch fiel die initiale Anbindung der Schnittstellen in den Zeitraum dieser Projektarbeit und wurde durch den Verfasser durchgeführt.

Erstellung Schulungsunterlagen und Durchführung Mitarbeiterschulung

Zum Zeitpunkt der formalen Einreichung des Themas der Projektarbeit bei der IHK war angedacht, dass bereits im März 2015 entsprechende Schulungsunterlagen für die Webshop Integration zu erstellen sind. Die auf Basis dieser Unterlagen durchzuführende Schulung der Mitarbeiter der Freizeitpark GmbH sollte ebenfalls noch im Monat März durchgeführt werden.

Auf der Grund der bereits erwähnten Verzögerung der Anbindung der KORONA Schnittstellen – auf Grund externer, fremdbestimmter Faktoren – wurde übereingekommen, dass die Schulung der Mitarbeiter der Freizeitpark GmbH erst dann stattfinden soll, wenn die Schnittstellenanbindung an die

restlichen eingeführten KORONA Produkte durchgeführt wurde. In Folge dessen fiel die Erstellung der Schulungsunterlagen sowie die Durchführung der Schulung in den Projektzeitraum und wurde durch den Verfasser durchgeführt.

Geänderter Schwerpunkt bezüglich der Kundendokumentation

Im Projektantrag wurde bezüglich der Dokumentation vermerkt, dass eine ausführliche Systemdokumentation sowie ein Datenflussdiagramm im Zuge der Durchführung entstehen sollen. Auf Grund der Tatsache, dass nach Einreichung des Projektthemas die Erstellung von Schulungsunterlagen und die Durchführung der Schulung kurzfristig in den Projektzeitraum verlegt wurden, wurde entschieden den Schwerpunkt der zu erstellenden Dokumentation für diese Projektarbeit auf die Kundendokumentation zu legen.

Diese Entscheidung wurde primär auf Grund der vorgegebenen zeitlichen Rahmenbedingungen – innerhalb welcher eine solche Projektarbeit durchgeführt werden muss – getroffen. Im veranschlagten Zeitrahmen war die Erstellung eines sehr detaillierten Datenflussdiagramms nicht möglich. Diese Art technischer Dokumentation hätte aus Sicht des Kunden auch keinen entscheidenden Mehrwert geboten, weshalb übereingekommen wurde, diese Zeit stattdessen in die Verfeinerung der Kundendokumentation zu investieren.

3 Migrationskonzept

3.1 Zieldefinition

Bevor die eigentliche Migration durchgeführt wurde, wurden zuerst alle zu erreichenden Ziele schriftlich festgehalten und anschließend evaluiert wie und in welcher Reihenfolge diese am besten zu umsetzen sind. Zu den bereits in der Soll-Analyse herausgestellten Zielen und Wünschen des Kunden kamen noch interne – mit der Migration verknüpfte – Ziele der Combase AG hinzu.

Diese internen Ziele umfassten unter anderem den Wunsch, dass zukünftig Neukunden – welche eine Anbindung der KORONA Produkte an Webshopsoftware wünschen – diese Integrationsdienstleistungen nur noch auf Grundlage einer einheitlichen, von der Combase AG zusammengestellten, Softwarelösung auf Basis von Magento CE 1.9 angeboten werden sollen. Mit diesem Schritt sollte sichergestellt sein, dass zukünftig eine möglichst geringe Diversität in Bezug auf die bei unseren Kunden eingesetzten Webshoplösungen herrscht. Ziel dieser Vereinheitlichungsstrategie ist es, möglichst wenig zeitlichen - und damit auch finanziellen – Aufwand für die Pflege und Weiterentwicklung der Codebasis der Integrationsschnittstellen investieren zu müssen. Weiterhin sollte auch der Aufwand für unsere Supportabteilung, um Unterschiede beziehungsweise Probleme, welche nur in ganz bestimmten Versionen der Magento Webshop Software auftreten, zu minimieren.

Die Vereinheitlichung der eingesetzten Softwarebasis sollte über die verwendete Version der Magento Webshop Software hinaus vorangetrieben werden. Zur Erreichung dieses Ziels kam der Verfasser dieser Projektarbeit nach Rücksprache mit dem technischen Experten des Bereichs Serverdienste der Combase AG – Herrn Sebastian Kleint – überein, dass zukünftig neue Webshopsysteme nur noch auf Basis eines aktuellen Ubuntu Server LTS Betriebssystems aufgesetzt werden sollen. Den Ausschlag für das Ubuntu Server LTS Betriebssystem gaben dabei folgende drei Aspekte: der vom Hersteller gewährte Langzeitsupport von fünf Jahren, die gute Verfügbarkeit von aktuellen Softwarepaketen und die Funktionsweise der Paketmanagement Software *apt-get* sowie die einfache Möglichkeit eine Aktualisierung auf neuere Versionen von Ubuntu Server LTS durchführen zu können.¹

3.2 Abwägung Alternativen zur Erreichung der Ziele

Bezüglich der Erreichung der gesteckten Migrationsziele gab es für den Verfasser Projektarbeit relativ wenig Handlungsspielraum. Die Entscheidung, die Integrationsschnittstellen zukünftig nur noch auf Basis eines Magento CE 1.9 Webshops anzubieten, wurde bereits im Vorfeld der Durchführung der Projektarbeit innerhalb des Betriebs getroffen. Die Festlegung für das zukünftig zu verwendende Serverbetriebssystem war ebenfalls bereits im Vorfeld zwischen dem technischen Experten Serverdienste und dem Verfasser getroffen wurden.

¹ Detaillierte Beweggründe für den Umstieg auf Ubuntu Server LTS sind in Tabelle 3: Beweggründe für die Verwendung von Ubuntu Server LTS aufgelistet.

Die einzige, noch zu klärende Frage war die Art der Durchführung der Schulung bezüglich der Webshopintegration. Auf Grund des ambitionierten Zeitplans von einer Woche für die Durchführung der Migration wurde in Rücksprache mit der Projektleiterin Webshop Relaunch der Freizeitpark GmbH & Co. KG – Frau Andrea Köppke – beschlossen, dass die Vermittlung der Schulungsinhalte durch eine telefonische Schulung erfolgen soll. Eine Vor-Ort Schulung wurde auf Grund des begrenzten Teilnehmerkreises – nur Frau Köppke selbst sollte geschult werden – als nicht zielführend erachtet. Weiterhin handelte es sich um eine Aufbauschulung – Herr Dönhoff hatte bereits mehrere Vor-Ort Schulungen in der Kalenderwoche 14 beim Kunden durchgeführt – sodass der Kunde bereits grundsätzlich mit den KORONA Produkten und deren Funktionalitäten vertraut war.

3.3 Migrationsschritte & zeitlicher Fahrplan

Wie bereits unter Punkt 2.3 – Begründete Abweichungen vom ursprünglichen Projektantrag – beschrieben, wurde die Migration der Magento Webshop Software von Version CE 1.6.2.0 auf Version 1.9.1.0 durch einen Mitarbeiter der Multimedia Agentur GmbH bereits im Vorfeld der Durchführung dieser Projektarbeit vollzogen. Um Migrationsprobleme möglichst frühzeitig erkennen und darauf reagieren zu können, wurde in Absprache mit Herrn Sebastian Kleint vereinbart, dass zuerst die Betriebssystemumgebung aktualisiert werden soll. Dies sollte alle zentralen Komponenten des genutzten LAMP-Stacks betreffen.

Die Aktualisierung der Betriebssystemumgebung und aller zentraler Komponenten des LAMP-Stacks war ersten Tag des Projektzeitraums – Montag, den 13.04.2015 – terminiert. Anschließend sollte noch am gleichen Tag das Schulungsmaterial bezüglich der Webshopintegration durch den Verfasser der Projektarbeit erstellt werden und anschließend an Frau Köppke – die verantwortliche Projektleiterin bei der Freizeitpark GmbH und Co. KG – per Email versendet werden. Durch die geplante frühzeitige Übersendung der Schulungsunterlagen sollte sichergestellt werden, dass Frau Köppke sich zumindest auf dem Papier bereits mit den Schulungsinhalten vertraut machen kann, bevor die eigentliche Telefonschulung durchgeführt werden würde.

Für den zweiten Tag des Projektzeitraums wurde die Einrichtung der entsprechenden Integrationsschnittstellen in den Webshop der Freizeitpark GmbH & Co. KG eingeplant. Hierfür wurden zwei Stunden veranschlagt, weil durch den Verfasser lediglich Installations- und Konfigurationsaufgaben durchgeführt werden mussten; es war also keine vollständige Neuentwicklung von Softwarekomponenten zur Umsetzung des Projekt erforderlich. Nach Durchführung der Installation und Konfiguration wurde ein minimaler Funktionstest bezüglich der Schnittstellenfunktionalität eingeplant, um sicherzustellen dass die Schnittstellenanbindung ebenfalls korrekt in Magento CE 1.9 Umgebung ihren Dienst verrichtet.

Da sich der Verfasser im Vorfeld bereits über die wichtigsten Änderungen zwischen – der bisher im Firmenumfeld genutzten – Magento CE 1.7 und der – jetzt erstmals im Zuge des Webshop Relaunchs

beim Kunden Freizeitpark GmbH & Co. KG eingesetzten – Magento CE 1.9 informiert hatte, wurde ein zusätzliches Zeitfenster am zweiten Tag der Migration eingeplant. Dieses Zeitfenster war für die Behebung etwaiger Kompatibilitätsprobleme, welche durch die Verwendung unserer selbst entwickelten Schnittstellenanbindungen in Kombination mit der brandneuen Magento CE 1.9 entstehen könnten, vorgesehen.

Anschließend wurden für den Nachmittag des zweiten Projekttags die Durchführung diverser Schnittstellentests eingeplant. Hierfür war ein lediglich Zeitrahmen von zwei Stunden vorgesehen, da eine Neukonzeption der zu prüfenden Testfälle nicht notwendig war. Stattdessen konnte der Verfasser auf eine Reihe von Standard Testfällen zurückgreifen, welche sich in der Vergangenheit bei der Einrichtung der Schnittstellenanbindungen für Magento CE 1.7 Systeme bewährt hatten.

Der Mittwoch Vormittag vor für die telefonische Vermittlung der Schulungsinhalte eingeplant. Hierfür wurde eine Zeitspanne von einer Stunde für die Durchführung der Schulung veranschlagt. Für den Nachmittag des dritten Tages im Projektzeitraum waren Abnahmetests bezüglich der Schnittstellenintegration durch einen Vertreter des Kunden eingeplant. Hierbei war es besonders wichtig, dass bereits am zweiten Tag im Projektzeitraum bei der Durchführung der eigenen Testläufe alle gängigen Szenarien erfolgreich durchgespielt werden konnten, da ansonsten die Abnahme durch den Kunden gefährdet gewesen wäre, was zwangsläufig zu ungewünschten Verzögerungen geführt hätte.

Für den Vormittag des vierten Tages im Projektzeitraums war die Liveschaltung des – jetzt um die Integration mit den anderen KORONA Produkten erweiterten – Webshops in zwei Stufen geplant. Als erster Schritt sollte durch eine Änderung der DNS Einträge bezüglich der IP Adresse des Webshop Servers dieser wieder aus dem Internet erreichbar sein. Gleichzeitig sollte mit einer Softwarelösung sichergestellt werden, dass nach Änderung der DNS Einträge nur ein sehr begrenzter Kreis von Testern Zugriff auf den neuen Webshop erhalten. Diese sollten dann einige Testbuchungen im Webshop durchführen um sicherzustellen, dass keine Buchungsprobleme auftreten. Dieser Schritt war notwendig, da es einige wenige Testszenarien gab, welche sich nur sinnvoll unter tatsächlichen Livebedingungen durchführen ließen. Zu diesen Szenarien zählt unter anderem der Test bezüglich der korrekten Funktionsweise der Schnittstelle des Zahlungsdienstleisters.

Sollten bei diesen letzten Testläufen keine Probleme mehr auftreten, wurde mit dem Kunden vereinbart, dass anschließend sofort die Freischaltung des Webshops für alle restlichen Kunden erfolgen soll. Auf Basis dieser Zeitplanung konnte im Idealfall der gesamte Donnerstag Nachmittag sowie der darauf folgende fünfte Tag des Projektzeitraums für das Verfassen der Projektdokumentation genutzt werden. Dieser ambitionierte Zeitplan bezüglich der Erstellung der Dokumentation sollte dadurch ermöglicht werden, dass bereits während der Durchführung der anderen Projektphasen möglichst umfassende Notizen und Protokolle angefertigt werden, damit anschließend nur noch eine weitgehende Ausformulierung dieser Mitschriften nötig sein würde.

4 Durchführung Migration

4.1 Upgrade Serverumgebung

Für die Migration der Serverumgebung inklusive des kompletten LAMP-Stacks hielt ich mich an einen, im Kollegenkreis geschätzten – im Internet frei verfügbaren – Migrationsleitfaden.² Die Verbindung zum Server wurde selbst mit dem SecureShell Client SSH hergestellt, wobei mir bereits im Vorfeld der Durchführung der Projektarbeit die zur notwendigen Authentifizierung am Server benötigten Zugangsdaten durch meinen Projektbetreuer mitgeteilt wurden.

Bei der Durchführung des Updates des Betriebssystems kam es vorerst zu keinen unvorhergesehen Problemen. Mergekonflikte, welche durch das Überschreiben alter Konfigurationsdateien durch neuere Versionen des Betriebssystems verursacht wurden, konnten durch eigene Recherche des Verfassers – und in einigen wenigen Zweifelsfällen durch die Rücksprache im Kollegenkreis – aufgelöst werden. Nach der Aktualisierung der Betriebssystemumgebung auf Ubuntu Server 14.04 LTS wurden jetzt folgende Versionen der zentralen LAMP-Stack Komponenten verwendet:

Komponente	Versionsstand vor Update	Versionsstand nach Update
Betriebssystem	Ubuntu Server 12.04 LTS	Ubuntu Server 14.04 LTS
Webserver	Apache HTTP Server 2.2.22	Apache HTTP Server 2.4.7
Skriptsprache	PHP 5.3.10	PHP 5.5.9
Datenbank	MySQL CS 5.5.40	MySQL CS 5.5.41

Abbildung 1: Versionsstand Komponenten LAMP-Stack vor und nach Update des Betriebssystems

Nach der anschließenden Durchführung eines Reboots des Serversystems offenbarte sich ein erstes Problem: via apt-get ließen sich keine neuen Pakete mehr installieren, da der Webserver die Hostnamen der verschiedenen Paketrepositories nicht mehr auflösen konnte. Hierbei lag nahe, dass die gesamte DNS Auflösung nicht mehr zuverlässig funktionierte, was durch einen Aufruf diverser Hostnames mit Hilfe des Kommandozeilen-basierten Browsers **w3m** verifiziert wurde. Eine anschließende Prüfung der Datei /etc/network/interfaces bestätigte, dass in dieser der Eintrag für den zu verwendeten DNS Server komplett fehlte.

Es ist zu vermuten, dass im Vorfeld der Durchführung der Projektarbeit die IP Adresse des DNS Servers manuell in die Datei /etc/resolv.conf eingetragen wurde. Änderungen, welche direkt in dieser Datei durchgeführt werden, sind nicht persistent gehen bei einer Aktualisierung des Betriebssystems sowie einem Neustart des Servers verloren. Auf Grund dessen kontaktierte der Verfasser den Ansprechpartner unseres Webhosting Providers – Herrn Thompson – bezüglich des zu verwendenden DNS Servers. Nachdem der von Herrn Thompson genannte DNS Server persistent in der Datei

² Siehe Migrationsleitfaden Ubuntu Server LTS 12.04 auf Ubuntu Server LTS 14.04

/etc/network/interfaces hinterlegt wurde, war auch eine erneute Durchführung der Paketaktualisierung via apt-get problemlos möglich.

Nachdem die Betriebssystemumgebung migriert war, wurde anschließend eine Migration der Apache HTTP Webserver Konfiguration durchgeführt. Bei der Recherche im Vorfeld der Durchführung des Projekts zeichnete sich bereits ab, dass eine Aktualisierung des Apache HTTP Servers eine Aktualisierung einer ganzen Reihe von Konfigurationsdateien erfordern würde, damit der Webserver anschließend wieder problemlos durchstarten kann. Bei der Aktualisierung der Apache Konfiguration wurde wieder auf einen im Kollegenkreis empfohlenen Migrationsleitfaden zurückgegriffen.³

Es wurden dabei die gängigen, im Migrationsleitfaden empfohlenen Anpassungen an den Konfigurationsdateien vorgenommen:

1. Die Konfigurationsdateien der einzelnen VHOSTs wurden explizit mit der .conf Dateiendung versehen, damit die Konfiguration korrekt angezogen wird.
2. Die Verzeichnis Einträge in den VHOST Konfigurationsdateien wurden um den Eintrag ***Require all granted*** erweitert.
3. Die Einträge bezüglich zu setzender Optionen wurde mit einem expliziten + bzw. - als Präfix versehen, da eine Mischung von Optionen mit Präfix und Optionen ohne explizites Präfix unter Apache 2.4 nicht mehr gültig ist.

```
001-webshop-prod.1.diff
--- 001-webshop-prod
+++ 001-webshop-prod.conf
@@ -9,7 +9,8 @@
-      Options -Indexes FollowSymLinks MultiViews
+      Require all granted
+      Options -Indexes +FollowSymLinks +MultiViews
```

Abbildung 2: Zusammenfassung durchgeführte Änderungen Apache Konfiguration

Nachdem diese Änderungen durchgeführt wurden, wurde ein Konfigurationsselbsttest mit Hilfe des Kommandozeilenwerkzeugs apachectl initiiert. Nachdem das Werkzeug keine Meldungen mehr bezüglich etwaiger Konfigurationsfehler zurücklieferte, wurde der Apache Webserver neu gestartet. Anschließend ließ sich bereits per Browser das (Kunden) Frontend des Webshops im Webbrowser erfolgreich anzeigen.

Die Prüfung der Protokolldateien der Skriptsprache PHP sowie der genutzten relationalen Datenbanksoftware MySQL CS 5.5.41 lieferten keine Auffälligkeiten, welche Grund zur Besorgnis gegeben hätten. Folglich konnte als nächster Migrationsschritt die notwendigen Vorkehrungen getroffen werden, um die Anbindung an die Schnittstellen zu den anderen KORONA Produkten auf dem Webserver einzurichten. Dazu wurde unter anderem des Kommandozeilenprogramms ***php --info***

³ Siehe Migrationsleitfaden Apache HTTP Server 2.2 auf Apache HTTP Server 2.4

kontrolliert, ob alle notwendigen Erweiterungen für PHP bereits korrekt installiert und geladen waren. Beim Abgleich zwischen den installierten Erweiterungen auf dem Server und der von uns im Firmen-internen Wiki gepflegten Referenzliste fiel auf, dass auf dem Webserver die Verschlüsselungserweiterung *mccrypt* nicht korrekt eingebunden war. Eine Überprüfung via `apt-get` ergab, dass die Erweiterung selbst korrekt installiert war, aber aus unerfindlichen Gründen nicht korrekt angezogen wurde. Das Problem wurde durch ein `sudo php5enmod mccrypt` und ein anschließenden Apache Neustart via `sudo service apache2 restart` gelöst.

Beim anschließenden Login in das Magento Admin Backend wurde der Verfasser via eingebauten Backend Notification Service auf die Verfügbarkeit des kritischen Sicherheitsupdates SUPEE-5344 hingewiesen.⁴ Auf Grund der Tragweite der Sicherheitslücke, welche zur Ausführung von beliebigen PHP Code auf dem Webserver genutzt werden konnte, wurde der entsprechende Sicherheitspatch direkt durch den Verfasser eingespielt. Hierfür wurde das übliche Vorgehen zur Installation eines SUPEE angewandt: Herunterladen des Bash Skriptes von der Herstellerseite, Übertragung auf den Zielsystem via `scp` Kommandozeilentool, sowie das anschließende Deployment des Patches durch Ausführung des Shell Skriptes.

4.2 Einrichtung KORONA Modul & Schnittstellen Integration

Nachdem die korrekte Funktionsweise aller benötigten PHP Erweiterungen auf dem Webserver überprüft wurde und der dringende Sicherheitspatch installiert wurde, konnte nun die Einrichtung der Schnittstellenanbindung durchgeführt werden. Es wurde hierfür das Firmen-intern dokumentierte Standardvorgehen zur Einrichtung der Anbindung angewandt.

Die Einrichtung der Anbindung gliederte sich dabei in folgende durch den Verfasser durchgeführte Konfigurationsschritte:

1. Einrichtung der Artikelsynchronisation durch Installation eines Magento CSV Import Moduls sowie Durchführung diverser Konfigurationsanpassungen im KORONA.pos Server Backend
2. Einrichtung des Rechnungsimports für Webshopumsätze im KORONA.pos Server durch Anpassungen an KORONA.pos Server Konfigurationsdateien
3. Installation KORONA.modul für Magento Webshops inklusive Anpassung der verwendeten Templates und Einbindung eines Javascript-basierten Kalendercontrols
4. Durchführung Endkonfiguration KORONA.modul im Magento Admin Backend

Bei der Durchführung der Einrichtung kam es zu keinerlei erwähnenswerten Problemen.

4 Siehe Bekanntmachung Patch SUPEE-5344 für Magento Sicherheitslücke

5 Tests

5.1 Verwendeter Testkatalog

Die zu prüfenden Testfälle bezüglich der korrekten Funktionsweise der Schnittstellen mussten nicht im Zuge dieser Projektarbeit neu konzeptioniert werden. Stattdessen konnte der Verfasser auf einen bereits bewährten Katalog verschiedener Schnittstellentests zurückgreifen und musste diese Abläufe nur noch durchführen und die Ergebnisse dokumentieren. Im dafür verwendeten Testkatalog waren bereits allen wichtigen Aspekte festgelegt: die zu testenden Komponenten, durchzuführende Vorbereitungsschritte vor dem Testlauf, die Testschritte selbst, sowie die zu prüfenden Abnahmekriterien.

Details bezüglich der durchgeführten Testläufe können auf Grund des Umfang des Testkatalogs hier nicht in aller Ausführlichkeit dargestellt werden. Die Ergebnisse sind jedoch sehr ausführlich im angehängten Testprotokoll dokumentiert.⁵ Exemplarisch für auftretene Probleme während der Testläufe sollen aber im folgende drei Störungen und deren Behebung dokumentiert werden.

5.2 Auftretene Probleme während der Testläufe

5.2.1 Inkompatibilität KORONA.modul

Bei der Durchführung des Tests für die korrekte Funktionsweise des von der Combase AG entwickelten KORONA.modul stellte sich heraus, dass in Verbindung mit einer Magento 1.9.1.0 Installation das Modul weder eine Bestellbestätigung per Email an den Kunden versendet, noch die Ticket PDF Datei korrekt generiert. Nachdem der Verfasser einige Nachforschungen bezüglich mit Magento Version 1.9.1.0 eingeführten Änderungen durchgeführt hatte, war schnell klar, dass eine Anpassung am KORONA.modul erforderlich sein würde.

Das Modul verwendet eine angepasste Order.php Datei, welche um Integrationsfunktionen den Produkten KORONA.entry, KORONA.resource sowie zur Generierung von PDF Dateien erweitert wurde. Zur Behebung der Inkompatibilität wurde zuerst ein grafischer Abgleich zwischen der originalen Order.php aus Magento 1.7.0.2 und der Order.php aus Magento 1.9.1.0 durch den Verfasser durchgeführt. Anschließend wurden die festgestellten Änderungen auf die Version der Order.php des KORONA.modul übertragen. Anschließend konnte der Testlauf bezüglich der korrekten Funktionsweise des KORONA.modul erfolgreich durchgeführt werden. Die durchgeführten Änderungen am KORONA.modul wurden dokumentiert.⁶

5 Siehe Anhang 9.4 Testprotokolle

6 Siehe Anhang Order.php.diff im Quelltextverzeichnis

5.2.2 X-Frame-Options Header

Beim Test der Artikelsynchronisation stellte sich heraus, dass eine Anpassung der Apache Konfiguration in Bezug auf die X-Frame-Options notwendig war, damit das Formular zur Synchronisation korrekt im KORONA.pos Server Backend angezeigt werden konnte.

```
001-webshop-prod.2.diff
--- 001-webshop-prod.backup
+++ 001-webshop-prod.conf
@@ -33,7 +33,7 @@
-         Header always set X-Frame-Options DENY
+         # Header always set X-Frame-Options DENY
```

Abbildung 3: Notwendige Änderungen Apache Konfiguration für Artikelsynchronisation

5.2.3 Fehlschlag SSL Handshake beim Rechnungsimport

Der Versuch Rechnungen des Webshops in den KORONA.pos Server zu importieren scheiterte initial am einem nicht zustande gekommenen SSL Handshake. In Folge dessen mussten die SSL Konfiguration des Webshop Servers angepasst werden, wobei hier nach einem Hinweis aus dem Kollegenkreis auf den Mozilla SSL Configuration Generator zurückgegriffen wurde.⁷

```
001-webshop-prod.3.diff
--- 000-webshop-prod.old
+++ 000-webshop-prod.conf
@@ -26,8 +26,10 @@
-         SSLCipherSuite AES256+EECDH:AES256+EDH
+         # SSLCipherSuite AES256+EECDH:AES256+EDH
+         SSLProtocol All -SSLv2 -SSLv3
+         # Fuer Java 7 Kompatibilität! nicht abändern
+         SSLCipherSuite         ECDHE-RSA-AES128-GCM-SHA256:ECDHE-ECDSA-AES128-GCM-
SHA256:ECDHE-RSA-AES256-GCM-SHA384:ECDHE-ECDSA-AES256-GCM-SHA384:DHE-RSA-AES128-GCM-SHA256:DHE-
DSS-AES128-GCM-SHA256:kEDH+AESGCM:ECDHE-RSA-AES128-SHA256:ECDHE-ECDSA-AES128-SHA256:ECDHE-RSA-
AES128-SHA:ECDHE-ECDSA-AES128-SHA:ECDHE-RSA-AES256-SHA384:ECDHE-ECDSA-AES256-SHA384:ECDHE-RSA-
AES256-SHA:ECDHE-ECDSA-AES256-SHA:DHE-RSA-AES128-SHA256:DHE-RSA-AES128-SHA:DHE-DSS-AES128-
SHA256:DHE-RSA-AES256-SHA256:DHE-DSS-AES256-SHA:DHE-RSA-AES256-SHA:AES128-GCM-SHA256:AES256-GCM-
SHA384:AES128-SHA256:AES256-SHA256:AES128-SHA:AES256-SHA:AES:CAMELLIA:DES-CBC3-SHA:!aNULL:!
eNULL:!EXPORT:!DES:!RC4:!MD5:!PSK:!aECDH:!EDH-DSS-DES-CBC3-SHA:!EDH-RSA-DES-CBC3-SHA:!KRB5-DES-
CBC3-SHA
```

5.3 Ergebnis Testläufe

Nach Behebung der drei geschilderten Probleme konnten alle Punkte des Testkatalogs erfolgreich absolviert werden. Auch beim – durch Frau Köppke durchgeführten – Abnahmetest fielen keine Unstimmigkeiten mehr auf. Damit konnte der Webshop Relaunch inklusive Integration der Schnittstellen zu den anderen KORONA Produkten erfolgreich abgeschlossen werden. Ein kleinerer Konfigurationsfehler – bezüglich der zu verwendenden Rücksprung URL der Zahlungsschnittstelle Sparkassen Internetkasse – konnte durch Frau Köppke selbstständig behoben werden.⁸

⁷ Siehe Hinweis bezüglich Mozilla SSL Configuration Generator im Quellenverzeichnis

⁸ Siehe Testprotokoll 9.4.6 Zahlungsschnittstelle Sparkassen Internetkasse im Anhang

6 Zusammenfassung

6.1 Fazit

Das gewünschte Migrationsziel konnte im anvisierten Zeitrahmen erreicht werden. Aus Sicht des Kunden wurden die zentralen Ziele erreicht:

1. Die automatisierte Durchführung der Artikelsynchronisation wurde eingerichtet.
2. Durch die Einführung der KORONA Systeme und der Schulung der Mitarbeiter der Freizeitpark GmbH & Co. KG kann jetzt auch sehr kurzfristig Events geplant und abverkauft werden.
3. Eine Print@Home Ticketing Lösung inklusive Ticketvalidierung durch KORONA.entry wurde eingeführt
4. Bezüglich der in der Vergangenheit sporadisch auftretenden Rundungsdifferenzen konnte zum Zeitpunkt des Abgabe dieser Projektarbeit noch keine abschließende Aussage getroffen werden.

Eine konkrete Schätzung des durch die Integration der KORONA Schnittstellen beim Kunden geschaffenen Mehrwerts ist zum Zeitpunkt des Verfassens dieser Projektarbeit noch nicht durchführbar. Als problematisch erwies sich bei der Durchführung des Projektes die unklaren Systemgrenzen auf Grund der Vielzahl involvierter Akteure. Hier muss nachgelagert noch eine klarere Verantwortlichkeitshierarchie entwickelt werden, welcher Akteur im Falle von auftretenden Problemen im Zusammenhang mit dem Webshop Aufgaben des 1st und 2nd Level Supports erfüllen muss. Weiterhin gab es im Laufe der Durchführung des Projekts interne Abstimmungsschwierigkeiten zwischen zwei involvierten Mitarbeitern der Freizeitpark GmbH, was aber glücklicherweise nicht die erfolgreiche Durchführung der Migration gefährdete.

6.2 Ausblick

Nachdem die Migration beim Kunden erfolgreich durchgeführt wurde. Trat dieser kurzfristig an die Combase AG heran und bat um Lösungsvorschläge bezüglich zwei konkreter Punkte. Erstens soll eine einfachere Möglichkeit zur Zuordnung zwischen geleisteten Zahlungen und den dazugehörigen Bestellungen im Webshop geschaffen werden. Weiterhin erkundigte man sich bezüglich der Möglichkeiten bezüglich optischer Anpassungen des im KORONA.modul enthaltenen Veranstaltungskalenders. Im Zuge dieser zwei Punkte ergeben sich möglicherweise nachgelagerte kleinere Folgeaufträge für die Combase AG.

7 Quellenverzeichnis

- Ubuntu Server LTS Support Roadmap
<https://wiki.ubuntu.com/LTS>
- Migrationsleitfaden Ubuntu Server LTS 12.04 auf Ubuntu Server LTS 14.04
<http://ubuntuserverguide.com/2014/06/how-to-upgrade-ubuntu-server-12-04-to-ubuntu-server-14-04-lts.html>
- Migrationsleitfaden Apache HTTP Server 2.2 auf Apache HTTP Server 2.4
<https://www.linode.com/docs/security/upgrading/updating-virtual-host-settings-from-apache-2-2-to-apache-2-4>
- Bekanntmachung Patch SUPEE-5344 für Magento Sicherheitslücke
<https://rackspeed.de/blog/kritische-magento-sicherheitsluecke-patch-supee-5344/>
- Release Notes zur Veröffentlichung von Magento 1.9.1.0
http://merch.docs.magento.com/ce/user_guide/content/magento/release-notes-ce-1.9.1.html
- The X-Frame Options Response Header Guide
<https://developer.mozilla.org/en-US/docs/Web/HTTP/X-Frame-Options>
- Mozilla SSL Configuration Generator
<https://mozilla.github.io/server-side-tls/ssl-config-generator/>

8 Glossar

Apache HTTP Server

Quelloffener HTTP Webserver der Apache Software Foundation, welcher einen zentralen Bestandteil des typischen LAMP-Stacks darstellt.

Advanced Packaging Tool (APT)

Paketverwaltungssystem, welches im Umfeld des Debian/GNU Linux Projekts entstanden und inzwischen auch von einer ganzen Reihe anderer Linuxdistributionen genutzt wird. Die Hauptbestandteile des Pakets sind die beiden Kommandozeilenprogramme apt-get sowie apt-cache.

End of Life

Erreichtes Ende des offiziellen Lebenszyklus eines Software- oder Hardwareproduktes. In Bezug auf Softwareprodukte geht das Erreichen des End of Life Status üblicherweise damit einher, dass keine Aktualisierung für die Software mehr durch den Hersteller bereitgestellt werden. Dies kann bei sicherheitsrelevanter Software auf Seiten der Nutzer einen sehr hohen Migrationsdruck erzeugen, da selbst schwere Sicherheitslücken nach dem Erreichen des End of Life üblicherweise nicht mehr korrigiert werden.

KORONA.entry

Java-basierte Client-/Serversoftware der Combase AG für die Anwendungsfelder Einlassmanagement, Besucherlenkung sowie Zutrittskontrolle.

KORONA.pos Server

Java-basierte Serversoftware der Combase AG für die Anwendungsfelder Artikelverwaltung, Warenwirtschaft sowie Umsatzauswertung.

KORONA.pos Client

Java-basierte Kassensoftware der Combase AG. Arbeitet als Client-Gegenstück mit dem KORONA.pos Server zusammen.

KORONA.webshop Integration

Sammelbegriff für eine ganze Reihe von Integrationsschnittstellen, welche Verbindungen zwischen den Produkten der KORONA Software Familie mit der Magento Webshop Software schaffen.

LAMP-Stack

Zusammenstellung von Software bestehend aus einem GNU/Linux Betriebssystem, einem Apache HTTP Server, einer MySQL Datenbank und der Programmiersprache PHP. Der klassische LAMP-Stack stellt dabei eine äußerst gängige Softwareumgebung für das Hosting von Websoftware dar.

Magento Community Edition (CE) / Magento Enterprise Edition (EE)

PHP-basierte Software für den Betrieb von Onlineshops. Der Hersteller bietet die Software dabei in zwei verschiedenen, unterschiedlich lizenzierten Varianten an: die kostenfreie Magento Community Edition (Magento CE) sowie die kostenpflichtige Magento Enterprise Edition (Magento EE).

MagentoConnect

Online Marktplatz für den Vertrieb und Erwerb von kostenlosen sowie kostenpflichtigen Erweiterungen für die Magento Webshop Software.

Meld

Quelloffenes grafisches Softwaretool zum rekursiven Abgleich von Dateien und Verzeichnissen. Kann als Hilfsmittel für die Erstellung sowie für die Anwendung von Patches/Diffs verwendet werden.

MySQL Community Server (CS)

Relationales Datenbanksystem auf Basis der Sprache SQL. Typischer Bestandteil eines LAMP-Stacks.

PHP

Populäre Skriptsprache zur Erstellung dynamischer Webseiten sowie komplexer Webanwendungen. PHP ist Teil des klassischen LAMP-Stacks.

Print@Home Ticketing

Technische Lösung für den Onlineverkauf von Tickets, welche dem Kunden nach Bezahlung digital übersendet werden und die anschließend vom Kunden selbst ausdrucken werden können. Die Tickets weisen als Schutz gegen Betrugsversuche üblicherweise überprüfbare Sicherheitsmerkmale wie Barcodes oder auch QR-Codes auf.

Rundungsdifferenzen

Minimale Abweichungen der errechneten Posten- bzw. Zahlungssumme, welche durch Ungenauigkeiten bei der Anwendung von Rundungsregeln entstehen können.

SOAP

Netzwerkprotokoll zum Austausch von Daten zwischen verschiedenen Systemen.

SOAPUI

Grafisches Werkzeug für Softwaretests, welches besonders gut für die Analyse und Fehlersuche in Verbindung mit SOAP-basierten Webservices geeignet ist.

SSL Handshake

Prozess des Verbindungsaufbaus zwischen Client und Server bei SSL-verschlüsselten Verbindungen. Konnte der Handshake erfolgreich durchgeführt werden, können Client und Server anschließend Daten verschlüsselt miteinander austauschen.

Stakeholder

Person oder Gruppe, welche ein berechtigtes Interesse am Verlauf oder am Ergebnis eines Prozesses oder Projektes hat und versucht diese Interessen durchzusetzen.

Subversion/SVN

Quelloffene Softwarelösung zur Versionsverwaltung. Subversion zählt – im Vergleich zu moderneren verteilten Versionsverwaltungssystemen wie git oder Mercurial – zu den klassischen Systemen mit zentraler Client-/Serverarchitektur.

SUPEE

Bezeichnung für offizielle, vom Hersteller herausgegebene Sicherheitsupdates für die Magento Webshop Software. Diese werden für die kostenlose Community Edition in der Regel als Bash Skript auf der Magento Website angeboten, während der Hersteller für kostenpflichtige Enterprise Edition Updatepakete bereitstellt, welche komfortabel direkt aus dem administrativen Backend eines Magento Webshops eingespielt werden können.

Ubuntu Server LTS

Servervariante des Debian-basierten Ubuntu Linux Betriebssystem. LTS Versionen zeichnen sich dabei durch sehr umfangreichen offiziellen Supportzeitraum von fünf Jahren aus, bevor die Software offiziell ihr End of Life erreicht hat.

9 Anlagenverzeichnis

9.1 Projektantrag

Ausbildungsbetrieb / Praktikumsbetrieb

Combase AG, Behringstraße 45, 01159 Dresden

Projektbezeichnung

Migration der Magento Webshopumgebung inklusive Portierung von Erweiterungen, die zur Integration in die KORONA Softwareumgebung notwendig sind.

Projektbeschreibung

Die derzeit eingesetzte Magento Systemumgebung hat bereits das Ende seines offiziellen Lebenszyklus erreicht. Auch die in dieser Installation genutzten Systemkomponenten (Linux Distribution, Apache Webserver, MySQL Datenbank, PHP Laufzeitumgebung) sind nicht mehr auf einem aktuellen Stand. Insgesamt stellt dies ein großes Sicherheitsrisiko für Produktivumgebungen dar. Dies gilt insbesondere im Hinblick auf die öffentliche Zugänglichkeit solch eines Gesamtsystems. Ziel ist es die Systemkomponenten sowie den Magento Webshop auf einen aktuellen Softwarestand zu migrieren. Gleichzeitig ist es erforderlich, dass die von uns entwickelten Module zur Anbindung an die KORONA Softwareumgebung weiterhin nach der Migration voll funktionsfähig sind. Dabei sind auch kundenspezifische Inhalte zu portieren.

Projektumfeld

Combase AG, Abteilung IT Services

Projektphasen inklusive Zeitplanung

1. Definition:

- Analyse IST-Zustand 2h
- Analyse SOLL-Zustand 2h

2. Planung:

- Erstellung Migrationskonzept 6h

3. Durchführung:

- Upgrade Systemumgebung 1h
- Upgrade Magento 1h
- Upgrade Module für KORONA Integration 2h

- Datenmigration 4h
- Integrationstest 1h
- Regressionstest 2h

4. Abschluss:

- 4.1 Fazit 1h
- 4.2 Probleme 1h
- 4.3 Ausblick 1h

5. Dokumentation 11h

Dokumentation zur Projektarbeit

- Systemdokumentation
- Datenflussdiagramm
- Modulbeschreibung KORONA Integration
- Testprotokoll

Bearbeitungsdauer von

Montag, 13. April 2015

Bearbeitungsdauer bis

Freitag, 17. April 2015

Präsentationsmittel

- Overheadprojektor
- Flipchart

Andere Präsentationsmittel

- Laptop
- Beamer

Themenbetreuer

Herr Sebastian Kleint, IT Services, Combase AG

Auflagen

Laut Richtlinien der IHK ist eine Produktdokumentation und eine Kundendokumentation zu erstellen.

9.2 Tabellenverzeichnis

Tabelle 1: Beteiligte Unternehmen Webshop Relaunch & zugeteilte Aufgabengabe

Unternehmen	Aufgabenbereich
Freizeitpark GmbH & Co. KG	<ul style="list-style-type: none">• Auftraggeber Gesamtprojekt inklusive Teilprojekt Webshop Relaunch
Combase AG	<ul style="list-style-type: none">• Aktualisierung Webserverumgebung auf aktuelle Ubuntu LTS Version• Anbindung Webshopsoftware an bestehende Schnittstellen der KORONA Softwareprodukte schaffen• Schulung Mitarbeiter Freizeitpark GmbH & Co. KG im Umgang mit den geschaffenen Webshop Anbindungen zu den anderen eingeführten KORONA Produkten
Multimedia Agentur GmbH	<ul style="list-style-type: none">• Kleinere Anpassungen an der Webshopsoftware• Kosmetische Designanpassungen/Customizing
Internet Services GmbH	<ul style="list-style-type: none">• Hosting Webshop Server• Hosting Mailserver

Tabelle 2: Ansprechpartner der beteiligten Unternehmen & formale Rollen im Projekt

Unternehmen	Ansprechpartner	Rolle
Combase AG	Herr Thomas Dönhoff	Projektleiter Gesamtprojekt Einführung KORONA Produkte bei der Freizeitpark GmbH & Co. KG
Combase AG	Herr Sebastian Kleint	Technischer Experte Serverdienste
Combase AG	Herr Martin Vohland	Technischer Experte Webshopintegration
Freizeitpark GmbH & Co. KG	Frau Andrea Köppke	Projektleiterin Webshop Relaunch
Freizeitpark GmbH & Co. KG	Herr Sascha Wachsmann	Technischer Leiter IT Services der Freizeitpark GmbH & Co. KG
Multimedia Agentur GmbH	Herr Thomas Gold	Webentwickler
Internet Services GmbH	Herr Homer Thompson	IT Hosting Consultant

Tabelle 3: Beweggründe für die Verwendung von Ubuntu Server LTS

Aspekt	Erklärung
Langzeitsupport	Bei den Ubuntu Server LTS Versionen garantiert der Hersteller Canonical einen Langzeitsupport für das Betriebssystem von insgesamt fünf Jahren. Dieser lange garantierte Supportzeitraum ermöglicht der Combase AG zusätzliche Planungssicherheit.
Paketverfügbarkeit, Aktualität, Bedienbarkeit Paketmanager	In den Vergangenheit wurde in der Combase AG auf Grund bereits vorhandener technischer Expertise vorwiegend auf Linux-Betriebssysteme auf Basis der <i>SLES</i> bzw. <i>OpenSuSE</i> Familie gesetzt. In den letzten Jahren wurden aber eine Reihe von Mängeln offenbar, welche unter den Systemintegratoren den Wunsch nährten zukünftig ein alternatives Betriebssystem einzusetzen. Besonders die Aspekt des Paketmanagements führte auf den SuSE Systemen immer wieder zu Problemen. Oftmals waren die Pakete veraltet, neuere Pakete standen nicht zur Verfügung und die Bedienung des SuSE Paketmanagers <i>zypper</i> wurde gegenüber Ubuntu's <i>apt-get</i> als veraltet und nicht mehr zeitgemäß empfunden.
Einfache Aktualisierung des Betriebssystems	Bereits in der Vergangenheit in der Combase AG durchgeführte Testläufe bezüglich der Aktualisierung der unternehmensintern verwendeten Ubuntu LTS Server zeigten, dass selbst ein Upgrade von einer Ubuntu LTS Version auf die darauffolgende Ubuntu LTS Version mit moderatem Aufwand durchführbar ist. Im Sinne der Zukunftssicherheit gab uns dieser Aspekt die nötige Sicherheit, dass auch zukünftig aktuelle und stabile Systempakete für das Betriebssystem zur Verfügung stehen werden und sich der zeitliche Aufwand für die Durchführung der Aktualisierung in einem vertretbaren Rahmen bewegen sollte.

9.3 Quelltextverzeichnis

Order.php.diff

```
--- ./shop_prod/app/code/local/Mage/Sales/Model/Order.php.backup
+++ ./shop_prod/app/code/local/Mage/Sales/Model/Order.php
@@ -310,7 +309,17 @@
+   const EMAIL_EVENT_NAME_NEW_ORDER      = 'new_order';
+   const EMAIL_EVENT_NAME_UPDATE_ORDER   = 'update_order';
@@ -348,15 +357,16 @@
-   const ACTION_FLAG_COMMENT             = 'comment';
+   const ACTION_FLAG_COMMENT              = 'comment';
+   const ACTION_FLAG_PRODUCTS_PERMISSION_DENIED= 'product_permission_denied';
@@ -534,6 +544,9 @@
+   if (!$this->_canVoidOrder()) {
+       return false;
+   }
@@ -572,18 +584,25 @@
+   return $this->_canVoidOrder() ? $this->getPayment()->canVoid($this->getPayment()) : false;
+ }
+
+ /**
+  * Check whether order could be canceled by states and flags
+  *
+  * @return bool
+  */
+ protected function _canVoidOrder()
+ {
+     if ($this->canUnhold() || $this->isPaymentReview()) {
+         return false;
+     }
-     $state = $this->getState();
-     if ($this->isCanceled() || $state === self::STATE_COMPLETE || $state === self::STATE_CLOSED) {
-         return false;
-     }
-     return $this->getPayment()->canVoid(new Varien_Object);
+     return true;
@@ -781,6 +800,10 @@
+     if ($this->getActionFlag(self::ACTION_FLAG_REORDER) === false) {
+         return false;
+     }
+
@@ -808,18 +831,14 @@
-         $product = Mage::getModel('catalog/product')
-             ->setStoreId($this->getStoreId())
-             ->load($productId);
+         $product = Mage::getModel('catalog/product')
+             ->setStoreId($this->getStoreId())
+             ->load($productId);
+     }
+     if (!$product->getId() || (!$ignoreSalable && !$product->isSalable())) {
```

```

        return false;
    }
}
-
-
-     if ($this->getActionFlag(self::ACTION_FLAG_REORDER) === false) {
-         return false;
-     }
@@ -1151,7 +1170,7 @@
-     if ($this->canCancel()) {
+     if ($this->canCancel() || $this->isPaymentReview()) {
@@ -1245,22 +1264,27 @@
-     * Send email with order data
+     * Queue email with new order data
+     *
+     * @param bool $forceMode if true then email will be sent regardless of the fact that it was already
sent previously
    *
    * @return Mage_Sales_Model_Order
+    * @throws Exception
    */
-    public function sendNewOrderEmail()
+    public function queueNewOrderEmail($forceMode = false)
@@ -1530,7 +1572,15 @@
-     $mailer->send();
+
+     /** @var $emailQueue Mage_Core_Model_Email_Queue */
+     $emailQueue = Mage::getModel('core/email_queue');
+     $emailQueue->setEntityId($this->getId())
+         ->setEntityType(self::ENTITY)
+         ->setEventType(self::EMAIL_EVENT_NAME_NEW_ORDER)
+         ->setIsForceCheck(!$forceMode);
+
+     $mailer->setQueue($emailQueue)->send();
@@ -1551,13 +1601,26 @@
-     * Send email with order update information
+     * Send email with order data
+     *
+     * @return Mage_Sales_Model_Order
+     */
+    public function sendNewOrderEmail()
+    {
+        $this->queueNewOrderEmail(true);
+        return $this;
+    }
+
+    /**
+     * Queue email with order update information
+     *
+     * @param boolean $notifyCustomer
+     * @param string $comment

```

```

+     * @param bool $forceMode if true then email will be sent regardless of the fact that it was already
sent previously
+     *
+     * @return Mage_Sales_Model_Order
+     */
-     public function sendOrderUpdateEmail($notifyCustomer = true, $comment = '')
+     public function queueOrderUpdateEmail($notifyCustomer = true, $comment = '', $forceMode = false)
@@ -1614,11 +1679,32 @@
-         $mailer->send();
+
+         /** @var $emailQueue Mage_Core_Model_Email_Queue */
+         $emailQueue = Mage::getModel('core/email_queue');
+         $emailQueue->setEntityId($this->getId())
+         ->setEntityType(self::ENTITY)
+         ->setEventType(self::EMAIL_EVENT_NAME_UPDATE_ORDER)
+         ->setIsForceCheck(!$forceMode);
+         $mailer->setQueue($emailQueue)->send();
+
+         return $this;
+     }
+
+ /**
+  * Send email with order update information
+  *
+  * @param bool $notifyCustomer
+  * @param string $comment
+  *
+  * @return Mage_Sales_Model_Order
+  */
+ public function sendOrderUpdateEmail($notifyCustomer = true, $comment = '')
+ {
+     $this->queueOrderUpdateEmail($notifyCustomer, $comment, true);
+     return $this;
+ }

```

9.4 Testprotokolle

9.4.1 KORONA.modul Test

Getestete Komponenten/Teilkomponenten

- Kalendercontrol KORONA.modul
- KORONA.entry Anbindung
- PDF Generierung inklusive Erzeugung Barcode
- Email Versand an den Kunden

Durchgeführte Vorbereitungsschritte

1. KORONA.modul auf Kundenserver nach Installationsanleitung installiert
2. Kalendercontrol nach Anleitung in die genutzten Templates des Magento Themes eingebettet und auf gewünschter Seite im Shopsystem eingebunden
3. Grundkonfiguration KORONA.modul im Magento Backend durchgeführt (Hinterlegung KORONA.resource Webservice URL, KORONA.entry Webservice URL, Kassen ID Verkaufsstelle Webshop, Reservierungszeit)
4. Testevent im KORONA.resource erstellt und für die Verkaufsstelle Webshop verfügbar gemacht

Testablauf

1. Kalendercontrol zur Auswahl des gewünschten Termins nutzen
2. Beliebige Anzahl Tickets für das gewählte Event in den Warenkorb legen
3. Warenkorb zur Kasse bringen und vollständig durchbuchen
4. Genutztes Emailkonto abrufen und Eingang Ticket Email überprüfen

Abnahmekriterien

1. Das Kalendercontrol wird korrekt im Header der Shopseite angezeigt
2. Bei der Auswahl eines Datums mit Hilfe des Kalendercontrols wird man anschließend korrekt zur Liste der Events für das gewählte Datum weitergeleitet
3. Bei der Auswahl eines Datums in der Vergangenheit wird man unter Berücksichtigung der im Magento Backend konfigurierten Vorlaufzeit auf den richtigen Folgetag weitergeleitet
4. Für das gewählte Events wird die Zahl der noch zur Verfügung stehenden Plätze korrekt angezeigt

5. Es wird genau die ausgewählte Anzahl an Tickets in den Warenkorb gelegt
6. Die noch verbleibende Reservierungszeit wird dynamisch im Warenkorb angezeigt
7. Die Tickets werden nach Ablauf der Reservierungszeit wieder automatisch aus dem Warenkorb entfernt
8. Beim Gang zur Kasse unterscheidet sich die zu zahlende Summe nicht von der Summe der Posten im Warenkorb (unter Abzug etwaiger Versandgebühren und nach Verrechnung von Rabatten)
9. Der Kunde erhält eine Bestellbestätigung per Email nachdem erfolgreich bezahlt wurde
10. Das Aussehen der Bestellbestätigung entspricht dem im Magento Admin Backend definierten Layout des jeweiligen Email Templates
11. Die Bestellbestätigung enthält das gewünschte PDF Ticket als Anhang
12. Das Aussehen des PDF Tickets entspricht dem im Magento Backend am jeweiligen Artikel hinterlegten Layout
13. Die PDF Datei enthält die korrekte Anzahl an Tickets sowie unterschiedliche Barcodes für jedes Ticket

Testergebnisse

1. Die Abnahmekriterien 1-8 wurden vollständig erfüllt
2. Das Abnahmekriterium Nummer 9 wurde nicht erfüllt; der Kunde erhielt keine Bestellbestätigung per Email
3. Die Abnahmekriterien 10-13 waren zu diesem Zeitpunkt nicht überprüfbar, da die Überprüfung voraussetzte, dass dem Kunden zuvor eine Bestellbestätigung per Email samt PDF Ticket zugesandt wurde

Folgeschritte

1. Analyse der Protokolldateien im Magento Verzeichnis
2. Onlinerecherche bezüglich Änderungen in der Handhabung versendeter Emails unter Magento 1.9
3. Abgleich bezüglich etwaiger Änderungen an der Datei Order.php zwischen Magento 1.7.0.2 und Magento 1.9.1.0 via ***diff -Naur Order.php.old Order.php.new***
4. Durchführung eines Mergevorgangs mit Hilfe des grafischen Werkzeugs Meld
5. Generierung eines Patches für Anpassungen an der Datei Order.php

6. Übertragung des generierten Patches auf die angepasste Order.php Datei des KORONA Moduls mit Hilfe von Meld
7. Anschließend Wiederholung des KORONA.modul Tests
8. Alle geforderten Abnahmekriterien wurden nach der Durchführung der Anpassungen an der Order.php Datei des KORONA Moduls erfüllt
9. Überprüfung der durchgeführten Änderungen an der Order.php durch einen Webentwickler der Combase AG
10. Einstellen der durchgeführten Änderungen unter Versionskontrolle nach positiver Überprüfung durch den Webentwickler

9.4.2 KORONA.pos Server Artikelsynchronisation

Getestete Komponente/Teilkomponente

- KORONA.pos Server Artikelsynchronisation

Durchgeführte Vorbereitungsschritte

1. Magento Erweiterung für automatisierten Import von CSV Dateien installiert
2. Notwendige Konfigurationsschritte (Anlage Webshop Kasse, Zuordnung eines eigenen Sortiments, Hinterlegung Navigationsknoten) im KORONA.pos Server Backend durchgeführt
3. Synchronisationsskript auf dem Webshop Server eingespielt und passende Konfiguration (Hostname KORONA.pos Server, Authentifizierungsdaten für Anmeldung) hinterlegt

Testablauf

1. Artikelsynchronisation über angelegten Navigationsknoten im KORONA.pos Server aufrufen
2. Artikelsynchronisation initiieren

Abnahmekriterien

1. Das Formular zur Artikelsynchronisation wird im KORONA.pos Server Backend angezeigt
2. Nach Durchführung der Artikelsynchronisation erhält der Nutzer eine korrekte Rückmeldung bezüglich des Erfolg/Misserfolges des Synchronisationsvorgangs
3. Neue Artikel, welche erstmals synchronisiert wurden, wurden korrekt im Magento Backend angelegt
4. An bestehende Artikel, welche nur aktualisiert wurden, gingen keine Artikeleigenschaften (Langtext Artikelbeschreibung, hinterlegte Artikelbilder) verloren

Testergebnis

1. Abnahmekriterium 1 wurde nicht erfüllt: das Formular zur Artikelsynchronisation wurde nicht im KORONA.pos Server Backend angezeigt
2. Die Abnahmekriterien 2-4 waren auf Grund dieser Tatsache vorerst nicht überprüfbar

Folgeschritte

1. Prüfung Apache Protokolldateien auf KORONA.pos Server System
2. Prüfung Apache Protokolldateien auf Magento Webshop Server

3. Onlinerecherche bezüglich möglicher Probleme bei der Anzeige externer PHP Skripte via `<iframe>`
4. Anpassung Apache Konfiguration auf dem Magento Webshop Server; Auskommentierung der hinterlegten X-Frame-Options
5. Neustart Apache HTTP Server via ***service apache2 restart***
6. Wiederholung des Testlaufs
7. Die geforderten Abnahmekriterien 1-4 wurden jetzt vollständig erfüllt
8. Aktualisierung Firmen-interne Dokumentation bezüglich der Problematik X-Frame-Options

9.4.3 KORONA.entry Integrationstest

Getestete Komponente/Teilkomponente

- KORONA.entry Integration des KORONA Moduls

Durchgeführte Vorbereitungsschritte

1. KORONA.modul Test erfolgreich bestanden

Testablauf

1. Ticketgültigkeit via KORONA.entry Web Frontend überprüfen
2. Hinterlegte Organisationseinheiten für das Ticket via direkter Abfrage des Datensatzes in der Firebird Datenbank überprüfen

Abnahmekriterien

1. Das Ticket ist laut KORONA.entry Web Frontend genau für den im Webshop vom Kunden gebuchten Eventzeitraum gültig
2. Es gibt keine inhaltlichen Widersprüche zu den im KORONA.pos Server hinterlegten Ticketdaten des Artikels
3. In der KORONA.entry Datenbank wurde das Ticket für alle Organisationseinheiten hinterlegt, welche laut Ticketdaten dem Artikel im KORONA.pos Server zugeordnet wurden

Testergebnis

1. Die Abnahmekriterien 1-3 wurden vollständig erfüllt

Folgeschritte

Keine Folgeschritte notwendig.

9.4.4 KORONA.pos Server Rechnungsimport

Getestete Komponente/Teilkomponente

1. SOAP Poller Komponente des KORONA.pos Server

Durchgeführte Vorbereitungsschritte

1. Im Magento Backend SOAP/XML-RPC Benutzer angelegt und ausreichende Rechte zugewiesen
2. Verbindungstest gegen den Magento SOAP Webservice mit Hilfe von SOAPUI durchgeführt
3. Konfiguration SOAP Poller in KORONA.pos Server application.properties Datei vorgenommen

Testablauf

1. Testbuchung im Webshop durchführen
2. Den korrekten Import der Buchung aus dem Webshop im KORONA.pos Server Backend kontrollieren

Abnahmekriterien

1. Die Buchung aus dem Webshop ist als Rechnung im KORONA.pos Server Backend aufrufbar
2. Die Preise der einzelnen Posten und die Gesamtsumme des Belegs stimmen zwischen Webshop und KORONA.pos Server überein
3. Bestandsänderungen für Artikel mit Bestandsführung im KORONA.pos Server wurden korrekt verbucht

Testergebnis

1. Das Abnahmekriterium Nummer 1 wurde nicht erfüllt; die Buchung aus dem Webshop war nicht im KORONA.pos Server Backend auffindbar
2. Die Abnahmekriterien 2-3 waren deshalb vorerst nicht überprüfbar

Folgeschritte

1. Kontrolle Protokolldateien KORONA.pos Server
2. Rücksprache im Kollegenkreis und gemeinsame Fehlersuche
3. Erkenntnis, dass der SSL Handshake zwischen SOAP Poller des KORONA.pos Server und dem Magento Webshop nicht erfolgreich durchgeführt werden kann
4. Verwendung des Mozilla SSL Configuration Generator zur Erstellung einer geänderte Apache

SSL Konfiguration

5. Hinterlegung der geänderten Apache SSL Konfiguration auf dem Magento Webshop Server
6. Erneute Durchführung des Testlaufs
7. Jetzt wurden die Abnahmekriterien 1-3 vollständig erfüllt
8. Dokumentation der gewonnenen Erkenntnisse im Wiki der Combase AG

9.4.5 KORONA.resource Integrationstest

Getestete Komponente/Teilkomponente

1. KORONA Magento Bridge
2. Belegimportschnittstelle KORONA.resource

Durchgeführte Vorbereitungsschritte

1. KORONA.modul Test erfolgreich bestanden
2. Notwendige Konfigurationseinstellungen an der KORONA Magento vorgenommen

Testablauf

1. Testbuchung im Webshop durchführen
2. Den korrekten Import der Buchung aus dem Webshop im KORONA.resource Backend kontrollieren

Abnahmekriterien

1. Im KORONA.resource ist für die Buchung aus dem Webshop ein neuer Vorgang entstanden
2. Am Vorgang sind die korrekten Kundendatensätze aus dem Webshop hinterlegt
3. Die Posten am Vorgang entsprechen den im Webshop gebuchten Events
4. Aktivierte Tickets sind mit korrekter Barcodenummer am Vorgang einsehbar

Testergebnis

1. Die Abnahmekriterien 1-4 wurden vollständig erfüllt

Folgeschritte

Keine Folgeschritte notwendig.

9.4.6 Zahlungsschnittstelle Sparkassen Internetkasse

Getestete Komponente/Teilkomponente

1. Magento Modul für Anbindung Zahlungsschnittstelle Sparkassen Internetkasse

Durchgeführte Vorbereitungsschritte

1. KORONA.modul Test erfolgreich bestanden
2. Magento Modul für Anbindung Sparkassen Internetkasse wurde bereits installiert und konfiguriert
3. Umstellung der DNS Einträge für die öffentliche Domain des Webshops war bereits erfolgt, sodass eine Rückmeldung der Zahlungsschnittstelle beim Webshop erfolgen konnte
4. Ein spezieller Testartikel wurde bereits in den Webshop eingepflegt und dieser wurde nur einer begrenzten Kundengruppe von Testern zugänglich gemacht

Testablauf

1. Kauf des zuvor angelegten Artikels im Webshop durchführen
2. Nach Durchführung des Kaufs im Magento Admin Backend Bestellstatus überprüfen

Abnahmekriterien

1. Keine Abweichungen zwischen der Summe des Warenkorbs beim Gang zur Kasse und dem geforderten Betrag beim Zahlungsanbieter
2. Der Rücksprung vom Zahlungsdienstleister zum Webshop funktioniert
3. Die Bestellung weist den Status **complete** im Magento Admin Backend aus
4. An der Bestellung sind entsprechende Transaktionsinformationen hinterlegt, welche eine nachträgliche Zuordnung der geleisteten Zahlung zu einer Bestellung ermöglichen

Testergebnis

1. Abnahmekriterium Nummer 1 wurde erfüllt
2. Die Abnahmekriterien 2-3 wurden nicht erfüllt; es erfolgte weder ein korrekter Rücksprung zum Webshop, noch wies die entsprechende Bestellung im Magento Admin Backend anschließend den korrekten Bestellstatus aus
3. Abnahmekriterium Nummer 4 war auf Grund dessen zu diesem Zeitpunkt nicht überprüfbar

Folgeschritte

1. Prüfung des Konfiguration des Zahlungsmoduls im Magento Admin Backend
2. Rücksprache mit der Projektleiterin bei der Freizeitpark GmbH & Co. KG, Frau Köppke
3. Änderung der hinterlegten Rücksprungadresse im Backend der Sparkassen Internetkasse durch Frau Köppke
4. Erneute Durchführung des Testablaufs
5. Jetzt wurden die Abnahmekriterien 1-4 vollständig erfüllt
6. Dokumentation der gewonnenen Erkenntnisse bezüglich der Hinterlegung der Rücksprung URL des Zahlungsanbieters im Wiki der Combase AG

9.5 Kundendokumentation

- Deckblatt Kundendokumentation ist die letzte Seite der regulären Projektarbeit
- Kundendokumentation als eigenständiges Dokument im Anhang der Projektarbeit
- Kundendokumentation erhält eigenes Inhaltsverzeichnis; Seitennummerierung beginnt für Kundendokumentation wieder von vorn!
- Anhänge der Kundendoku sind als eigene Anhänge an die Kundendokumentation anzufügen

9.6 Datenblatt KORONA Integration für Magento-basierte Webshops

Die Combase AG bietet eine ganze Reihe von Integrationsschnittstellen, welche die Produkte der KORONA Softwarefamilie mit Magento-basierten Webshops ideal verknüpfen können.

Unsere Magento Integration umfasst dabei folgende Funktionalitäten:

KORONA.pos Server Integration

1. Artikelsynchronisation

- Komfortabler Import von Artikeln aus KORONA.pos in Magento Shopsysteme.
- Kein fehleranfälliger, manueller Übertrag von Artikeln mehr notwendig.
- Keine Einschränkung Ihrer kreativen Möglichkeiten. Ausführliche Artikelinformationen (Artikelbeschreibungen, Artikelbilder) können weiterhin im Magento Admin Backend gepflegt werden.
- Bei Bedarf kann der Artikelsynchronisationsprozess auch vollständig automatisiert werden.

2. Automatischer Rechnungsimport

- Webshopverkäufe werden automatisch in KORONA.pos Server importiert.
- Auswertung Ihrer Webshopverkäufe mit Hilfe der umfangreichen KORONA.pos Server Berichte.
- Nutzen Sie die umfangreichen Analysemöglichkeiten des KORONA.pos Server um Ihre E-Commerce Business noch weiter zu optimieren.

3. Verbuchung von Bestandsänderungen

- Automatische Verbuchung von Abverkäufen bestandsgeführter Artikel im KORONA.pos Server.
- Keine manuelle Nachbuchung von Webshopverkäufen zum Ziel der Bestandserfassung mehr notwendig.

4. Automatischer Kundenimport

- Automatisierter Import von Kundendaten aus Magento Webshopssystemen.
- Zuordnung der Webshopkunden zu abgetrennten Kundengruppen möglich.
- Möglichkeit zum anschließenden Abgleich der gesammelten Kundendaten aus dem KORONA.pos Server einem externen Customer Relationship Management (CRM) System.

KORONA.modul für Magento Webshops

1. Veranstaltungskalender

- Ihre Webshopkunden buchen die von Ihnen angebotenen Events mit Hilfe eines intuitiv zu bedienenden, grafischen Veranstaltungskalenders.
- Anzeige der noch verfügbaren Plätze für Ihre Events in Echtzeit.

2. Print@Home Ticketing

- Hinterlegung beliebiger Ticketlayouts und Zuordnung dieser zu Ihren Webshopartikeln.
- Umfangreiche Möglichkeiten zur Ticketgeschaltung via HTML/CSS. Weiterhin besteht die Möglichkeit zur Einbettung von PHP Programmcode zur Erzeugung hochdynamischer Ticketlayouts.
- Print@Home Tickets werden direkt nach Bezahlung durch dem Kunden dynamisch generiert und per Email an diesen versandt. Kein manueller Versand von Hardtickets mehr notwendig.
- Schutz vor Ticketbetrug durch KORONA.entry Integration. Tickets werden bei der Generierung im KORONA.entry System registriert und durch einen Barcode geschützt.

KORONA.resource Integration

1. Flexibles Kontingentmanagement

- Komfortable Steuerung des zur Verfügung stehenden Ticketkontingents für Ihre Veranstaltungen über das KORONA.resource Backend.
- Möglichkeit zur Hinterlegung unterschiedlicher Ticketkontingente für unterschiedliche Verkaufsstellen. Sie bestimmen, wie viele Tickets im Webshop zur Verfügung stehen sollen und welches Restkontingent für die Abendkasse zurückgehalten werden soll.
- Sie möchten für Ihr Event mehrere aufeinanderfolgende Vorverkaufsphasen durchführen und auch die preislichen Konditionen in jeder einzelnen Phase variieren? Auch dies lässt sich über die umfangreiche und flexible Verkaufsstellenkonfiguration von KORONA.resource umsetzen.

2. Automatischer Import von Vorgängen

- Event-bezogene Abverkäufe aus dem Webshop werden automatisiert in KORONA.resource importiert und sind dort unter einer Vorgangsnummer abrufbar.
- Im KORONA.resource Backend stehen Ihnen selbstverständlich die bekannten, umfangreichen Möglichkeiten zur Nachbearbeitung sowie Auswertung zur Verfügung.