

Inhaltsverzeichnis

Tabellenverzeichnis	IV
Abbildungsverzeichnis	IV
Abkürzungsverzeichnis	V
1 Refactoring im Preisbock	6
1.1 Allgemeines zum Preisbock.....	6
1.2 Allgemeines zu Refactoring	6
1.3 Notwendigkeit eines Refactorings beim Preisbock	7
2 Optimierung der Datenbank	8
2.1 Normalisierung der Datenbank	8
2.2 Probleme bei der Datenportierung.....	12
3 Mögliche neue Systeme für den Preisbock.....	13
3.1 Magento.....	13
3.1.1 Allgemeines.....	13
3.1.2 Vorteile	14
3.1.3 Nachteile	14
3.1.4 Chancen	15
3.1.5 Risiken.....	15
3.1.6 Vorgehen bei Migration auf System.....	16
3.2 dotIndustry	17
3.2.1 Allgemeines.....	17
3.2.2 Vorteile	18
3.2.3 Nachteile	18
3.2.4 Chancen	18
3.2.5 Vorgehen bei Portierung auf dieses System	19
3.3 Intershop Enfinity Suite 6.....	20
3.3.1 Allgemeines.....	20
3.3.2 Vorteile	20
3.3.3 Chancen	21
3.3.4 Nachteile	21

3.3.5 Risiken.....	21
3.3.6 Vorgehen bei Migration zur Plattform	22
3.4 Zusammenfassung der Migrationsmöglichkeiten.....	23
4 Finale Systementscheidung	26
4.1 Entscheidung für eine Plattform.....	26
4.2 Mögliche Teilung des Preisbocks in Plugins.....	26
Anlagenverzeichnis.....	XXVIII
Quellen	XXIX
Ehrenwörtliche Erklärung.....	XXX

Tabellenverzeichnis

Tabelle 1: gelöschte Tabellen in optimierter Datenstuktur	10
Tabelle 2: Umwandlung zu neuen Nutzer- und Produkttabellen.....	10
Tabelle 3: gelöschte Attribute in optimierter Datenstuktur.....	11

Abbildungsverzeichnis

Abbildung 1: momentane Relation zwischen Kundentabellen.....	8
Abbildung 2: momentane Relation zwischenProdukttabellen.....	8

Abkürzungsverzeichnis

Abkürzung	Langschreibweise
API	A pplication P rogramming I nterface
AJAX	A synchronous J ava S cript a nd X ML
DBMS	D atabase M anagement S ystem
MySQL	M y S tructured Q uery L anguage
PB	P reis b ock
PHP	P ersonal H ome P age T ools
WP	W ord p ress

1 Refactoring im Preisbock

1.1 Allgemeines zum Preisbock

„Preisbock.de ist eines der beliebtesten Live-Shopping-Portale Deutschlands, das täglich ein stark reduziertes Produkt zum spontanen Kauf als »Bock des Tages« anbietet. Dieses Angebot gilt bis zu einer Woche, so lange der Vorrat reicht. [Beim Preisbock können] User auf der jeweiligen Produktseite über das Angebot des Tages diskutieren und Kommentare abgeben.“¹

Die Webapplikation Preisbock ist in ihrem momentanen Stand der Entwicklung auf die openSource-Blogsoftware Wordpress² aufgebaut. Diese wird durch mehrere Plugins, welche zusammen den Preisbock bilden, erweitert und gelangt final zu der Funktionalität von www.preisbock.de.

1.2 Allgemeines zu Refactoring

„Refaktorisieren ist der Prozess, ein Softwaresystem so zu verändern, dass das externe Verhalten nicht verändert wird, der Code aber eine bessere interne Struktur erhält. Es ist ein diszipliniertes Vorgehen, um Code zu bereinigen, das die Wahrscheinlichkeit, dabei Fehler einzuführen, minimiert.“³

Beim Refactoring wird folglich das Design von Code verbessert und optimiert, nachdem dieser geschrieben wurde. Somit wird eine Weiterentwicklung des Programms erleichtert.

¹ Übernommen aus Pressematerial Preisbock [Pre08]

² Weitere Informationen über Wordpress unter <http://wordpress-deutschland.org>

³ Definition aus [MFo00] deutsche Übersetzung, Seite xviii

1.3 Notwendigkeit eines Refactorings beim Preisbock

Die internen Strukturen des Preisbocks haben momentan einen erhöhten Optimierungsbedarf, da vor allem die Datenbankstrukturen eine effektive Weiterentwicklung behindern. Dieser Zustand ist entstanden, da der Preisbock an die openSource-Anwendung Wordpress entwickelt wurde, die nicht als Onlineshopsystem, sondern als Blogsoftware konzipiert wurde. Im frühen Anfangsstadium der Entwicklung des Preisbocks wurde zuerst die Anwendung Preisbock als eigenständige Applikation neben dem Wordpress geplant, final aber als ein Wordpress-Plugin, also als ein zusätzlicher Bestandteil des Wordpress, umgesetzt. Weitere Informationen über die Entwicklung und den internen Aufbau des Preisbocks können der zweiten Projektarbeit des Autors entnommen werden. Durch diese Kombination der zwei Systeme blieben die Datenbankstrukturen des Wordpress dabei weitgehend bestehen, sodass das Preisbocksystem um an vollständige Informationen über Produkte oder Kunden zu gelangen, intern mehrere Produkt- und Kunden-IDs auswerten muss. Zudem sind die Bestandsdaten in sich nicht vollständig konsistent. Eine weitere Problematik resultiert aus der Verknüpfung einer openSource-Anwendung mit einer Eigenentwicklung, da openSource-Anwendungen regelmäßig auf dem neuesten Stand gehalten werden müssen, eigene Entwicklungen dabei aber behindert werden oder eine erneute Portierungen nötig wird.

Diese Faktoren machen eine Weiterentwicklung und somit ein weiteres schnelles Wachstum des Preisbock immer schwieriger. Ein Refactoring ist somit notwendig geworden.

Da der Preisbock momentan nur als Erweiterung einer Blogsoftware besteht und kein professionelles Shopsystem zur Basis hat, ist es empfehlenswert ein Refactoring gleich zu nutzen, damit der Preisbock in ein neues, leistungsfähigeres System migriert werden kann. Der Aufwand einer notwendigen Datenbankneustrukturierung in Kombination mit der Anpassung und Optimierung interner Strukturen ist nur unwesentlich geringer als die Migration in ein neues Shopsystem. Die momentane Komplexität des Preisbocks hat die Komplexität eines durchschnittlichen Wordpress-Plugins überschritten. Den Entwicklern des Preisbocks ist kein anderes Plugin bekannt, was auch nur ansatzweise diesen Umfang erreicht.

2 Optimierung der Datenbank

2.1 Normalisierung der Datenbank

Die Wordpressvariante des Preisbocks wird bisher so entwickelt, dass die Datenbestände des Wordpress nicht oder nur geringfügig verändert und die ursprünglichen Datenbanktabellen des Wordpress nicht erweitert werden. Für dieses Vorgehen wurde sich entschieden, damit die Datenbestände des Preisbocks möglichst unabhängig von der Struktur der Datenbestände des Wordpress bleiben. Zudem sollten eine Aktualisierungen des Wordpress und somit auch eine eventuelle Änderung in der Struktur der Datenbanktabellen des Wordpress zu möglichst wenigen Änderungen an der Preisbockapplikation führen. Da aber trotzdem hauptsächlich für die Produkttexte und Logindaten von Nutzern Inhalte und Funktionalitäten von Wordpress genutzt werden, kommt es zu der Situation, dass zwischen Tabellen des Wordpress und gleichbedeutenden Tabellen in den Datenbeständen des momentanen Preisbocks nicht normalisierte Beziehungen vorliegen, die das System an dieser Stelle ausbremsen.

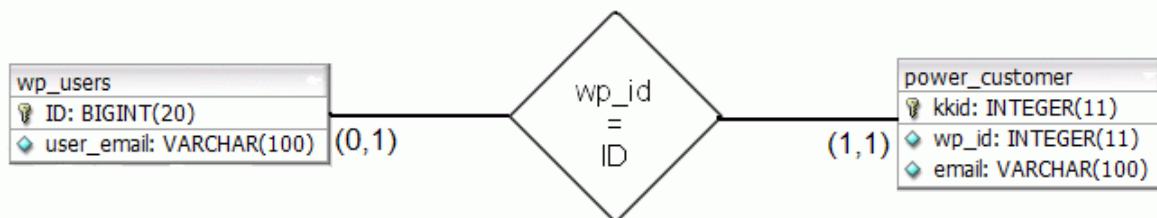


Abbildung 1: momentane Relation zwischen Kundentabellen

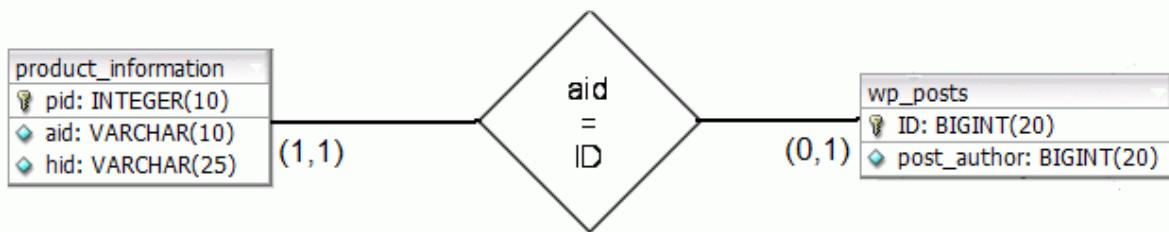


Abbildung 2: momentane Relation zwischen Produkttabellen

Dieses erschwert den störungsfreien Zugriff auf die zentralen Datenbestände bspw. von Produkt und Nutzer und damit die weitere Entwicklung des Preisbocks.

Weiterhin treten mit zunehmenden Entwicklungsstadium und ansteigenden Nutzerzahlen immer öfter Performanceprobleme auf. Dies geschieht vor allem bei einer Verknüpfung und Bearbeitung der Nutzer- mit den Produkttabellen und/oder den Verkäufen. Ein Beispiel hierfür wäre eine Abfrage, wann und für welches Produkt sich Nutzer extra registriert haben, damit sie jenes Produkt käuflich erwerben können. Statt einer Verknüpfung von drei Entitäten müssen hierfür beim Wordpress-Preisbock momentan fünf Entitäten verknüpft werden. Primäres Ziel muss es also sein, die Produkt- und Nutzertabellen zu jeweils einer Tabelle zusammenzuziehen, einheitliche Nutzer- und Produktidentifikatoren zu schaffen und die gesamte Datenbank in eine dritte Normalform zu überführen. „Die dritte Normalform analysiert funktionale Abhängigkeiten zwischen Nichtschlüsselattributen und formt sie in Schlüsselabhängigkeiten um. Alle Nichtschlüsselattribute sind gemäß zweiter Normalform voll funktional von einem Schlüssel abhängig.“⁴ Das heißt, dass alle Informationen eindeutig in der Datenbank abgelegt sind und sich die Entitäten in klaren Relationen zueinander befinden.

Der erste Schritt des Refactorings ist folglich eine komplette Übersicht über die nach dem Refactoring angestrebten Datenstrukturen zu bekommen. Eine optimierte Datenbankstruktur für den Preisbock ist im Anhang zu finden.

Bei der Optimierung der Datenbank wurden sämtliche Tabellen einer Notwendigkeitsprüfung unterzogen. Dabei stellt sich heraus, dass folgende Tabellen in der optimierten Datenstruktur des Preisbocks nicht mehr benötigt werden:

Tabellenname	Logischer Inhalt
en2de	Sehr alte Übersetzungstabelle für Labels (nicht aktiv)
power_current	aktuelle Nutzer auf der Plattform (nicht aktiv)
power_dyn_pay_config	Labelverwaltung für Versandmethoden
power_session	aktuelle Nutzer auf der Plattform (nicht aktiv)
power_trac	Tracking der Seitenzugriffe (nicht aktiv)
wp_forum_groups	Verschiedene Forumgruppen
wp_social	Links der Social-Bookmark-Dienste

⁴ entnommen aus [TKu07] Seite 94

wp_links	Linkverwaltung zum Wordpress zu anderen Blogs
wp_comments	Kommentare des Wordpress (ersetzt durch Forum)
wp_link2cat	Mapping von Kategorien und Links zu anderen Blogs

Tabelle 1: gelöschte Tabellen in optimierter Datenstruktur

Weiterhin werden die Produktkategorien wieder in die Struktur des Wordpress vor der Version 2.5 überführt, da deren Funktionalität für den Preisbock vollkommen ausreicht. In diesen Kategorietabellen sollen auch nicht die zugehörigen Wochentage des Produktes, sondern reelle Produktkategorien gespeichert werden. Der Wochentag kann anhand des Produktdatums bestimmt werden. Zudem werden neue Produkt- und Usertabellen geschaffen, in die momentan verteilte Tabellen migriert werden:

Aktuelle Tabellen	Zusammengefasste Tabelle
wp_users wp_forum_profil und power_customer	customer
processbar_start product_newsletter product_variation_name wp_posts product_informations	product

Tabelle 2: Umwandlung zu neuen Nutzer- und Produkttabellen

Zusätzlich wird eine neue Tabelle für die Verwaltung von Dateianlagen geschaffen, die auch die Aufgaben der Tabelle mapping_imagesize übernimmt. Eine zusätzliche Tabelle für Affiliatestatistiken wird bei der Normalisierung der Affiliate-tabellen nötig. Zusätzlich wurden die Meta-Tabellen entfernt.

Im nächsten Schritt wurden alle Attribute aus den Entitäten entfernt, die nicht mehr benötigt werden:

Tabelle	Attribut	Beschreibung
mail_remind	nicemail	Mailtemplate (immer gleich)
mailconfig	result	Statusflag (nicht genutzt)
power_allocate	tid	Timestamp
power_allocate	cart	Flag Warenkorb vorhanden
power_allocate	variation	Flag Variation vorhanden
power_allocate	b_nr	Hausnummer
product_bookmarks_options	EXTERNAL1	Externe ID
product_information	paypalinfo	Mehr Zahlungsinformationen
product_information	kreditcardinfo	Mehr Zahlungsinformationen
product_information	nachnahme_info	Mehr Zahlungsinformationen
product_information	payment_info	Mehr Zahlungsinformationen
product_information	trim	Immer leer
product_information	show_vendor	Flag Händler anzeigen
stats_user	SESSIONID	Session-ID des Nutzers
wp_forum_forums	parent_id	Gruppe des Forums
wp_forum_profil	forum_id	Zuweisung Nutzer->Forum
wp_forum_status	picurl	Bildpfad zu Avatar
wp_product2parcelservice	shipping_costs_austria	Lieferkosten nach Österreich
wp_ratings	rating_posttitle	Produkttitel
wp_ratings	rating_username	Username

Tabelle 3: gelöschte Attribute in optimierter Datenstruktur

Zudem werden jeweils die Felder für die Produkt- und Nutzer-IDs zusammengefügt, da es nach neuer Struktur nur noch jeweils eine Entität gibt. Eine Vereinfachung der Tabelle zur Verwaltung der Einstellungen ist auch sinnvoll. Final wurde bei der Optimierung ein Augenmerk auf die richtige Definition von Datentypen, NULL- und Default-Werten gelegt.

Eine Umbenennung / Vereinfachung der Tabellennamen ist optional.

2.2 Probleme bei der Datenportierung

Im Wordpress-Preisbock sind Attribute vorhanden, welche Daten beherbergen, die bei einer Normalisierung in mehrere Tupel oder Attribute getrennt werden müssen. So müssen die Attribute power_allocate.rechnung_name und power_customer.rechnung_name in einen Vor- und einen Zunamen getrennt werden. Weiterhin müssen die Adressinformationen des Kunden, die in der Tabelle power_customer gespeichert sind (13 Stück⁵), komplett in der Relation der Kundenadressen (power_customer_addresses) abgebildet werden. Zudem sind die Spalten b_land in den Tabellen power_customer und power_allocate in sich inkonsistent. In verschiedenen Versionen des Wordpress-Preisbocks wurde dort das Land ausgeschrieben, ein Länderindex als Fremdschlüssel zur Tabelle power_countries oder 0 bzw. nichts eingetragen. Das Attribut further_infos in der Tabelle power_allocate weist bei alten Bestellungen noch verschiedenste zusätzliche Bestellinformationen auf. Bei diesen Bestellungen ist meist auch das order_date (Bestelldatum) nicht gesetzt, sondern nur ein sich bei einer Aktualisierung des Tupels automatisch geänderter Timestamp tid. Die Tabelle wp_posts muss aufgetrennt werden in statische Seiten, Informationen zu Produkten und Produktanlagen. Zudem sind in dieser Tabelle in der Spalte post_content verschiedenste Informationen zum Produkt im Fließtext gespeichert. Die Spalte tva_ka.vids sollte in extra Tupel getrennt werden. Zudem kann wp_rating.rating_timestamp sowie power_customer.birthday in ein normales Datum umgewandelt werden.

Zusammenfassend ist es nötig, die vorhandenen kombinierten Strukturen aus Wordpress und Preisbock zu einer optimierten Struktur zugunsten der besseren Verarbeitung der Daten im Preisbock zu modifizieren. Durch die Zusammenlegung der momentan verteilten Nutzerinformationen sowie der Produktinformationen kommt es zu der Situation, dass jeweils nur noch eine eindeutige Nutzer- bzw. Produkt-ID vorhanden ist. Durch weitere Optimierungen im Sinne von Normalisierungen wird das Verarbeiten der Informationen erleichtert.

⁵ b_company, b_address, b_nr, b_city, b_postcode, b_land, rechnung_name, rechnung_lastname, rechnung_firstname, rechnung_company, rechnung_street, rechnung_city, rechnung_postcode

3 Mögliche neue Systeme für den Preisbock

3.1 Magento

3.1.1 Allgemeines

Magento ist ein noch relativ junges openSource-Shopsystem. Entwickelt wird es hauptsächlich von der Firma Varien aus den USA⁶, welche plant damit in den kommenden Jahren zum Marktführer bei den Onlineshopsystemen zu werden. Das System ist eine komplett objektorientierte PHP-Anwendung mit Zugriff auf eine MySQL-Datenbank, die sich als sehr flexibel darstellt. Die Entwickler haben sich dazu entschlossen, das Zend-Framework⁷ für PHP zu nutzen. Die Flexibilität wird durch eine enorm leistungsfähige Pluginschnittstelle erweitert. Die meisten verfügbaren Plugins befinden sich allerdings noch im Beta-Stadium. Die neueste Version 1.1.1 wurde Ende Juli 2008 als Stable veröffentlicht.

In der Anwendung wurden viele wichtige Eigenschaften wie Suchmaschinenoptimierung, Suchfunktionen, Produktbewertungen, Wunschzettel und ein Newslettertool integriert. Der Anwendung ist es möglich mehrere Shops in einem System zu verwalten. Zudem ist der komplette internationale Einsatz mit mehreren Sprachen und Währungen ohne weiteres möglich. Weiterhin ist es möglich viele Anpassungen vorzunehmen ohne den Kern der Anwendung verändern zu müssen. Dadurch bleibt das System selbst bei einer Individualisierung durch den Nutzer sehr lange updatefähig.

Durch die ständige Weiterentwicklung von Magento sowie einer ständig wachsenden Entwicklergemeinschaft, ist davon auszugehen, dass weitere Funktionalitäten folgen werden. In naher Zukunft wird das System vollständig auf den europäischen Markt angepasst worden sein, um in dieser Region weitere Verbreitung zu erlangen. Alle in einem Webshop benötigten Elemente vom Produkt über Variationen bis hin zu grundlegenden Statistiken sind in Magento bereits in der jetzigen Version integriert.

⁶ Firmen Webseite unter <http://www.varien.com/>

⁷ Einführung sowie Dokumentation und Beispiele unter <http://framework.zend.com/>

3.1.2 Vorteile

Magento ist genau wie der aktuelle Preisbock in der Programmiersprache PHP verfasst. Zudem ist es möglich mit wenigen Änderungen am Quelltext und mit Hilfe des Magento-Plugins „Lazzymonks Wordpress Integration“ ein Wordpress, welches ja die Grundlage des momentanen Preisbocks bildet, in ein Magento-System zu integrieren. Dieses Plugin bildet dann quasi ein Interface zwischen den beiden Systemen. Das benötigte Plugin befindet sich leider wie so viele noch im Beta-Stadium. Jedoch könnte sich hieraus eine Möglichkeit ergeben, den jetzigen Preisbock schrittweise auf das neue System zu portieren, da beide Systeme parallel betrieben und gegenseitig auf Funktionalitäten zurückgreifen können.

3.1.3 Nachteile

Durch den openSource-Charakter von Magento ist der Quelltext jeder Person frei zugänglich. Dies betrifft natürlich auch mögliche Sicherheitslücken.

Zudem ist der Standardkaufvorgang AJAX-basiert und müsste für den Preisbock komplett umgebaut werden. Da der Preisbock auf diesem System wie schon beim Wordpress eine Erweiterung eines openSource-Projektes ist, muss eine neue Pluginsprache erlernt und professionell angewandt werden. Trotz der wesentlich besseren Erweiterbarkeit der Datenstrukturen des Magento, ist es möglich, dass sich Elemente im neuen Preisbocks schon wie beim Wordpress nicht optimal in die Struktur integrieren können, damit ein Update immer möglich bleibt. Weiterhin läuft das Magento trotz erhöhter Performance in der aktuellen Version sehr langsam, wie anhand einer Testinstallation auf einem Entwicklungsrechner beobachtet werden konnte.

3.1.4 Chancen

Bisher gibt es wenige professionelle Firmen, die sich auf die Anpassungen von Magento spezialisiert haben. Hier könnte eine neue Dienstleistung für Kunden entstehen. Zudem kann das im Magento eingesetzte Zend-Framework die Arbeit bei der Entwicklung erleichtern. Das System ist zur Verwaltung mehrerer Shops ausgelegt, es ist also flexibel genug um evtl. zukünftig einen separaten Shop oder auch verschiedene Preisböcke für verschiedene Produktgruppen zu integrieren. Die Entwicklung durch die Firma Varien ist durch ihre komplette Objektorientierung und der Anwendung von Standards und Quasistandards sehr professionell. Sie wird durch Entwicklungen der openSource-Gemeinde erweitert, wobei die systeminterne Pluginschnittstelle von Magento zum Einsatz kommt, die Plugins anhand eines Pluginkeys⁸ in das Magento einfügt. Somit können auch Funktionalitäten entstehen, die der neue Preisbock im günstigen Fall nutzen kann.

3.1.5 Risiken

Bei Magento besteht die Möglichkeit, dass wie schon bei der Wordpressversion durch das PHP Ungenauigkeiten bei der Entwicklung auftreten können, da die Programmiersprache PHP Datentypen nur unvollständig unterstützt und daher Fehler in der Implementierung leicht toleriert. Zudem kann es passieren, dass angehängte Strukturen nicht gut durchgeplant bzw. suboptimal implementiert sind und somit in diesem Bereich die Entwicklungsarbeit behindern.

Magento ist in Betracht zu seiner Größe, Entwicklungszeit und Komplexität ein relativ junges System. Das System muss regelmäßig auf dem neuesten Stand gehalten werden, da Magento openSource ist und die internen Strukturen somit jedem bekannt sind. Durch den Einsatz von PHP und vielen Datenbanktabellen erscheint das System sehr Performance schluckend, wie anhand einer Testinstallation beobachtet werden konnte.

⁸ erhältlich im Magento-connect-Bereich unter <http://www.magentocommerce.com/magento-connect>

3.1.6 Vorgehen bei Migration auf System

Bei der Portierung erfolgt für Kunden, Produkte und Bestellungen doppelte Datenhaltung in beiden Systemen bis zum erfolgreichen Abschluss der Portierung. Somit können jederzeit beide Systeme über ihre eigenen Funktionalitäten auf die Daten zugreifen und ein evtl. notweniges Rollback ist möglich. Der Datenbankserver wird diese Mehrbelastung verkraften, da während des momentanen Spitzenbetriebes noch reichlich Ressourcen als Puffer vorhanden sind.

Für die Portierung der Bestandsdaten kann teilweise die Importfunktionalität des Magento genutzt werden. Durch die ständige redundante Datenhaltung können die Systeme gut schrittweise portiert werden.

1. Prototypen werden hergestellt, hauptsächlich für Vereinigung Magento mit Wordpress (vor allem URL-Rewriting) sowie Umgestaltung des AJAX-basierten Kaufvorganges. Zudem soll geprüft werden, ob die Objekte des Magento auch für die Wordpress-Variante des Preisbock genutzt werden können.
2. Kundendaten ins Magento portieren
Dabei werden der Kundenstamm (customer_entity) und die Kundenadressen (customer_address_entity) in die Datenbestände des Magentos integriert und einer Nutzergruppe „Käufer“ zugeordnet. Dabei soll geprüft werden. Kundendaten werden im Magento verwaltet, restliches Backend, Frontend und Kaufvorgang verbleiben in der Wordpressvariante.
3. Die Produktinformationen (product entity mit product entity gallery und inventory stock) werden ins Magento portiert. Datenhaltung erfolgt redundant, genutzt werden weiterhin das Backend und Frontend von Wordpress, auch zum Produkt einstellen.
4. Mit Hilfe des Wordpress-Plugins, einiger Modifikationen an beiden Systemen und einem angepassten URL-Rewriting mittels .htaccess-Datei wird das momentane Preisbocksystem mit dem Magento gekreuzt.

Dies wird benötigt, damit in diesem Schritt die Templates der Front über das Magento abgewickelt werden können. Dies betrifft die Startseite (Kommentare müssen aus Wordpressforum geladen werden), Passwort vergessen, Empfehlen, Produktrating, Impressum, Über Preisbock, Presse sowie restliche statische

Seiten. Der Bereich „Mein Bock“ sowie Händler, Kaufvorgang und CustomerService verbleiben im Wordpress, da Bestellungen, Widgets und Bockmarks noch nicht migriert wurden.

5. Der Kaufvorgang wird nach Magento portiert. Dabei ist darauf zu achten, dass nur ein Produkt gleichzeitig gekauft werden kann. Beim Checkout erfolgt weiterhin eine redundante Datenhaltung mit dem Wordpress, da zwar nun Produkte im Magento angelegt werden können, die Bestellungen aber weiterhin über das Preisbock-Backend verwaltet werden. Damit dies möglich ist, müssen die Lieferanten sowie Paketdienste und Zahlungsmethoden auch über das Magento-System verwaltet werden.
6. Im nächsten Schritt wird der Checkout in das Magento-System portiert (sales_order.*). Dabei müssen die Schnittstellen des Preisbocks nach außen mit portiert werden. Dies betrifft hier vor allem den Export zur TVA, Kundenstornierungen, die über das Magento realisiert werden müssen, weitere Statusverwaltung wie bezahlt und Retoure sowie die Paypalschnittstelle.
7. Folgend wird das Forum nach Magento portiert. Dabei ist es möglich, dass schon ein gutes Plugin vorhanden ist. Zu ergänzen ist dieses sicherlich mit dem Userstatus, Avatar und Signatur.
8. Im vorletzten Schritt werden die Werbemittel zum Produkt und zum Preisbock auf das Magento-System portiert. Dies betrifft die Widgets, Newsletter, Froogle, sowie das Bockmarking.
9. Zuletzt werden alle fehlenden Funktionalitäten in das Magento migriert. Die Schnittstellen zum Wordpress werden entfernt und das System durch eine finale Optimierung vom Wordpress-System getrennt.

3.2 dotIndustry

3.2.1 Allgemeines

Die webbasierte Anwendung dotIndustry ist eine Eigenentwicklung der dotSource GmbH ist. Sie ist unterteilt in einen Kern mit den grundlegenden Funktionalitäten,

welcher durch weitere Komponenten, wie beispielsweise einem von der dotSource GmbH entwickeltes Vertragsmanagement, erweitert wird. Auch die Webanwendung Newskraft basiert auf diesem Kern.

Bei Einsatz dieser Plattform würde eine langfristige Bindung des Preisbocks an Produkte und Dienstleistungen der dotSource GmbH erfolgen.

3.2.2 Vorteile

Die Einarbeitung der Preisbock-Entwickler in dieses System wird aufgrund dessen, dass es eine Eigenentwicklung der dotSource GmbH ist und aufgrund der Tatsache, dass sich dotIndustry als Basis der Webanwendung Newskraft etabliert hat, relativ gering ausfallen. Bei auftretenden Problemen können viele aktive Entwickler konsultiert werden. Ein Wissensaustausch kann leicht statt finden.

Die Portierung kann schrittweise und individuell erfolgen, da nur wenige Anpassungen an bestehende Systeme gemacht werden müssen.

3.2.3 Nachteile

Da dotIndustry bisher noch nie als Onlineshop integriert wurde, müssen die meisten Funktionalitäten wie die komplette Produkt- und Kaufumgebung sowie die Verwaltung der Bestellungen und einige Statistiken komplett neu implementiert werden. Dabei kann mitunter auf vorher geprüfte Klassenstrukturen des aktuellen Preisbocks zurückgegriffen werden.

3.2.4 Chancen

Da dotIndustry eine Eigenentwicklung der dotSource GmbH ist, kann das Shopsystem des migrierten Preisbocks modular aufgebaut werden. Somit können für den Preisbock zusätzlich geschaffene Module an Drittakten verkauft oder in anderen Anwendungen eingesetzt werden. Somit sind Synergieeffekte möglich.

Der Entwicklung des Preisbocks auf dieser Plattform kommt das momentane Refactoring der Newskraft-Plattform entgegen. Eine weitere Chance für die dotSource wäre, dass nur noch Java als Programmiersprache verwendet wird und somit die Kompetenzen mehr gebündelt werden können.

3.2.5 Vorgehen bei Portierung auf dieses System

Aufgrund der Tatsache, dass bei der Portierung auf die dotIndustry-Plattform keinerlei Shopkomponenten vorhanden sind, sind kaum Vorkehrungen zur doppelten Datenhaltung zu treffen.

1. Zuerst werden die Nutzerdaten aus dem Wordpress-Preisbock in die dotIndustry-Tabellen user, useraddress, userdata und userroupassignment portiert. Weiterhin werden die Tags vom Bockmarking in die Tabelle tag eingetragen.
2. Folgend werden grundlegende Klassen für Produkt und Preisbocknutzer im dotIndustry implementiert. Die Daten werden dabei aus der Datenbankstruktur des aktuellen Preisbocks geholt. Hierbei ist es möglich, dass über diese Klassen vorerst nur eine Lesefunktionalität implementiert und die Schreibfunktionalität später mit der optimierten Datenstruktur des aktuellen Preisbocks im dotIndustry implementiert wird.
3. Danach wird das Frontend des Preisbocks weitgehend auf die dotIndustry-Plattform portiert. Dies betrifft die Startseite, statische Seiten sowie evtl. Feed und Newsletter. Alle Elemente, die Änderungen an den Bestellungen tätigen (Customerservice und Checkout) verbleiben sowie das Backend in der Wordpress-Version des Preisbocks. Das gute Jobframework von Newskraft kann bei Jobs, die nichts an den Kaufvorgängen ändern, genutzt werden.
4. Im Anschluss daran werden die Produktinformationen in das dotIndustry eingeführt. Dabei kann eine Teilung in allgemeine und preisbockspezifische Produktinformationen zur besseren Verwendung als separat zu vertreibendes Modul betrieben werden. Eine redundante Produktdatenhaltung in den Preisbockversionen ist für die Bestellungsverwaltung im Wordpress-Preisbock nötig. Die Bearbeitung des Produktes sowie eine Verwaltung der Händler,

Lieferdienste und Zahlungsmöglichkeiten kann über ein Backend im dotIndustry erfolgen.

5. Darauf wird der Checkout und die Bestellverwaltung komplett in das dotIndustry portiert. Die Bestellverwaltung erfolgt dann über das Backoffice im dotIndustry. Kunden müssen im Customerservice des dotIndustry die Möglichkeit bekommen, selbst Stornierungen zu tätigen. Der Export zur TVA sowie die Paypalanbindung müssen bei diesem Schritt portiert werden.
6. Als vorletzten Schritt werden die Community-Features und Werbemittel auf die neue Version angepasst. Dazu zählen unter anderem das Forum, das Bookmarking sowie die Widgets.
7. Schließlich wird die Portierung des Preisbocks auf die dotIndustry-Plattform finalisiert, indem Schnittstellen zum Wordpress entfernt werden und der Preisbock an das dotIndustry optimiert wird.

3.3 Intershop Enfinity Suite 6

3.3.1 Allgemeines

Die Intershop Enfinity Suite 6 ist eine der größten kommerziellen Shopsysteme für den Online-Handel. Es ist dahingehend konzipiert, dass international operierende Händler auch in einer Mehrseiten-Funktionalität das System einsetzen können. So tun dies bisher viele große Onlineshops wie beispielsweise OTTO oder QUELLE. Die Geschäftsprozesse vom Verkauf an den Kunden bis hin zur Lieferantenbindung werden unterstützt.

Die Plattform ist in der Programmiersprache Java verfasst und setzt auf Oracle als DBMS. Sie bietet genug Flexibilität um den individuellen Bedürfnissen der Betreiber gerecht zu werden.

3.3.2 Vorteile

Das System ist wohl eines der leistungsfähigsten und hat sich schon bei vielen großen Händlern international bewährt. Der Performance ist sehr zufrieden stellend.

Der Einsatz dieses Systems würde die Kooperation zwischen der dotSource GmbH und Intershop bestätigen.

3.3.3 Chancen

Durch die Nähe zur Vertreiber- und Entwicklungsfirma Intershop, die sich im gleichen Gebäude wie die dotSource GmbH und die Preisbock GmbH befindet, sowie die guten Kontakte zwischen den Firmen sind bei möglicherweise auftretenden Problemen sicher hilfreiche Tipps erhältlich. Eine Weiterentwicklung des Systems erfolgt im großen Rahmen durch professionelle Softwareentwickler. Zudem ist es möglich, dass durch die Entwicklung an der Enfinity Suite ein Angebot an Consultants für Intershop entsteht. Final würde die dotSource GmbH nur noch Java in den Primären Entwicklungszweigen einsetzen und könnte sich voll auf diese Sprache konzentrieren können.

3.3.4 Nachteile

Es wird viele Probleme bei der Migration des Wordpress-Preisbocks in diese Plattform geben. Neben der ohnehin schon schwierigen Portierung der inkonsistenten Datenbestände des aktuellen Preisbocks auf eine neue Basisplattform, muss nun auch ein Wechsel des DBMS vom MySQL nach Oracle stattfinden. Dies erfolgt in Kombination mit einem Wechsel der Programmiersprache von PHP nach Java, sowie einer enorm steigenden Komplexität. Zudem entfallen Lizenzkosten für die Nutzung der Plattform.

3.3.5 Risiken

Bei der Migration in dieses System würde eine Abhängigkeit von einem Fremdanbieter entstehen. Zudem ist die Software für große, international operierende Shops konzipiert und Preisbock entspricht somit nicht der primären

Zielgruppe. Es ist wahrscheinlich, dass bei der Portierung Details aufgrund der hohen Komplexität des Systems übersehen werden.

3.3.6 Vorgehen bei Migration zur Plattform

Die Problematik bei der Migration auf dieses System ist die hohe Komplexität der Plattform in Kombination mit einem unterschiedlichen Datenbank-Management-System und einer anderen Programmiersprache. Eine Variante wäre, mit Hilfe eines Abstraction-Layers die Workflows der Intershop-Anwendung anzustarten und zurück- oder hingelieferte Daten für die Wordpressvariante des Preisbocks aufzubereiten (bspw. als XML). Diese Variante ist jedoch mit viel Aufwand verbunden und lässt die Performance stark leiden. Somit wird eine Variante favorisiert, bei der zuerst die Funktionalität und danach die Daten portiert werden.

1. In den PHP-Preisbock muss eine weitere Datenbankklasse implementiert werden, die den Zugriff auf eine Oracle-Datenbank realisiert.
2. Im ersten Schritt erfolgt die Migration der Nutzerdaten in die Datenstruktur der Intershop Enfinity Suite. Nachdem werden die Nutzerdaten im Wordpress-Preisbock aus dem Datenbestand der Enfinity gelesen. Templates, Backend und Checkout erfolgen weiter über die Wordpress-Plattform.
3. Als nächsten Schritt werden die Produktdaten in die neue Plattform migriert. Danach kann das Produkt über die Intershop-Plattform angelegt werden. Die Bestellungen sowie ihre Verwaltung und der Customerservice erfolgen weiter im Wordpress-Preisbock. Damit dem Produkt Händler zugewiesen werden können, muss deren Verwaltung im neuen System erfolgen.
4. Da nach diesem Schritt alle für eine Anzeige des Produktes notwenigen Informationen migriert wurden, können nun die Templates des Frontends über die neue Plattform realisiert werden.
5. Anschließend wird der Checkout in die neue Plattform migriert. Dabei müssen die Backendprozesse zur Bestellverwaltung sowie der Export zur TVA mitgezogen werden. Auch die Migration des Customerservice mit Stornierungsmöglichkeit für die Kunden erfolgt in diesem Schritt.

Die Werbemittel sowie das Forum und das Bocking verbleiben bis hier im Wordpress-Preisbock.

6. Nachdem alle zu den Bestellungen gehörigen Prozesse in die Intershop-Plattform migriert wurden, können nun das Forum, das Bocking, Werbemittel sowie alle noch evtl. verbleibenden Komponenten umgestellt werden.
7. Final werden alle Schnittstellen zum Wordpress gelöst und die neue Version des Preisbocks an die Enfinity Suite optimiert.

3.4 Zusammenfassung der Migrationsmöglichkeiten

Die Migration auf ein Magento-System ist am einfachsten zu realisieren, da dieses System wie schon die jetzige Version des Preisbocks auf der Programmiersprache PHP und dem DBMS MySQL basiert. Da dieses System einen kompletten Onlineshop auf openSource-Basis darstellt, sind die meisten vom Preisbock benötigen Features schon vorhanden. Trotz und wegen der leistungsfähigen Pluginschnittstelle dieser Plattform, kann es passieren, dass zu dem System hinzugefügte Strukturen wie schon in der Wordpress-Version des Preisbocks im Verlauf der Entwicklung Probleme bereiten. Zudem läuft das System auf einem Entwicklungsrechner trotz Performancezuwachs zur Version 1.1.1 langsam. Dennoch birgt die Migration des Preisbocks in dieses System die wenigsten Gefahren aufgrund seiner größten Ähnlichkeiten zum jetzigen System.

Die Plattform dotIndustry als hausinterne Entwicklung der dotSource GmbH ist die flexibelste Lösung für einen hybriden Betrieb während der Migration. Dies entsteht, da in diesem System bisher keinerlei Shopkomponenten enthalten sind. Somit müssen komplett neue Komponenten geschaffen werden, was wiederum mit einem erhöhten Aufwand verbunden ist. Jedoch muss bei diesem System nicht zuerst das System eines Fremdanbieters verstanden werden, damit dieses dann erweitert werden kann. Die hausinternen Entwickler, welche momentan schon andere Projekte auf dieser Plattform entwickeln, könnten mit Rat und Tat zur Seite stehen und den Einstieg in das System wesentlich erleichtern. Zudem können keine Inkompatibilitäten auftreten. Als interessanter Nebeneffekt ist zu betrachten, dass

Shopmodule so aufgebaut werden können, dass sie das momentan von der dotSource GmbH vertriebene System erweitern und an weitere Kunden verkauft werden können. Somit ist es möglich, dass ein Teil der Entwicklungskosten wieder zurück gewonnen werden kann. Zudem ist die Entwicklungsabteilung des Preisbocks sehr der Programmiersprache Java aufgeschlossen.

Die meisten Probleme werden bei der Migration in das Shopsystem Enfinity Suite der Firma Intershop entstehen. Dieses ist am unähnlichsten mit dem momentanen Preisbocksystem. Zudem Punkt, dass sich die Entwickler des Preisbocks größtenteils in eine neue Programmiersprache einarbeiten müssen, ändert sich das DBMS von MySQL auf Oracle. Durch die enorme Komplexität des Systems, sind auch die internen Datenbestände völlig anders aufgebaut. Ein hybrider Betrieb stellt sich als äußerst schwierig da, da Daten zwischen zwei verschiedenen Systemen (bspw. mittels XML oder eines zusätzlichen Abstraktionslayers) übergeben werden müssen. Bei dem Einsatz dieses Systems wäre eine Weiterentwicklung durch professionelle Programmierer seitens der Firma Intershop garantiert. Weiterhin würde dies die Kooperation zwischen Intershop und der dotSource bestätigen. Die Webapplikation Preisbock hätte zukünftig keine systembedingten Wachstumsprobleme mehr, wenn auch das System an sich für den momentanen Preisbock überdimensioniert groß ist.

Eine objektive Empfehlung geht an das System Magento, da dieses System dem Preisbock in seiner bisherigen Form am ähnlichsten ist. Zudem sind schon viele wichtige Shop-Komponenten im System verankert. Das Risiko bei der Portierung ist moderat, da es auch möglich ist über die Komponenten des Preisbocks auf die Komponenten des Magento zuzugreifen.

Eine subjektive Empfehlung geht allerdings an das System dotIndustry, da hier die Portierung wohl am unproblematischsten von Statten geht. Dies geschieht dadurch, dass noch kein Shopsystem besteht, auf das eine Migration stattzufinden hat. Außerdem können Inhouse-Entwickler bei der schnellen Einarbeitung in das System behilflich sein. Zudem bietet die Entwicklung an diesem System die Möglichkeit durch den Verkauf von Shopkomponenten an Dritte, zusätzliche Einnahmen zu erzielen. Weiterhin besteht nicht wie bei Magento die Gefahr, dass dem System

zugefügte Komponenten sich nicht einwandfrei in das System integrieren, welches bei der Wordpress-Variante des Preisbocks oft zu Fehlern führte.

Das Intershop-System kann als neue Plattform momentan nicht empfohlen werden, da es aufgrund seiner internen Strukturen und seiner Komplexität zu weit vom momentanen Preisbock abweicht und auftretende Fehler bei der Migration fast garantiert sind.

4 Finale Systementscheidung

4.1 Entscheidung für eine Plattform

Auf der Grundlage dieser Ausarbeitung sowie wirtschaftlicher Überlegungen ist die Führung der dotSource GmbH zu dem Entschluss gekommen, Magento als neue Plattform für den Preisbock einzusetzen. Entscheidend waren für die Wahl dieser Plattform, dass Magento auf den gleichen Technologien, nämlich der Programmiersprache PHP und dem Datenbankmanagementsystem MySQL aufbaut. Somit entfällt die lange Einarbeitungszeit in andere Technologien. Die Migration auf dieses System wird dadurch einfacher vonstatten gehen. Das Risiko ist moderat. Gleichzeitig ist eine schrittweise Migration in die neue Plattform möglich, während alte Elemente des Preisbocks weiter gewartet und angepasst werden können. Durch die technologischen Gemeinsamkeiten der alten und der zukünftigen Plattform des Preisbocks ist es möglich, im Preisbock entwickelte Komponenten mit wenigen Modifikationen in das Magento-System zu integrieren. Dadurch entfällt Entwicklungsarbeit, wodurch wiederum die Migration auf die neue Plattform kostengünstiger wird.

Der Einsatz von Magento korreliert auch mit der Firmenstrategie der dotSource GmbH, die sich zukünftig als Social-Commerce-Agentur etablieren möchte. Die für Preisbock als Magento-Plugins entwickelten Komponenten können zusammen mit einem Einsatz von Magento dem Kunden angeboten werden.

4.2 Mögliche Teilung des Preisbocks in Plugins

Damit Elemente des Preisbocks modular an Dritte weitergegeben werden können, ist es nötig bei der Migration ins Magento den Preisbock in mehrere Plugins zu teilen. Bei der Entwicklung ist dabei darauf zu achten, dass das Nichtvorhandensein eines anderen Plugins abgefangen wird. Eine mögliche Pluginstruktur sieht wie folgt aus:

- Export zur TVA – Anbindung des Lagers
- Paypalschnittstelle – für eingehende Paypalzahlungen

- Wirecard – Zahlungsverarbeitung
 - Risktool – Abschätzung von Risiken bei Kreditkartenzahlung
 - Kreditkartenunterstützung
 - EI LAs
- Forum mit erweiterten Nutzereinstellungen (Avatar, Signatur)
- Mailtemplates, die über das Backoffice verwaltbar sind
- Diverse Statistiken / Nutzertracking
- Affiliateverwaltung
- Erweiterte Gutscheinverwaltung
- Statusmails (eingehende Bestellung, letztes verfügbares Produkt gekauft, ...)
- Direkte Lieferantenbewerbung über das Frontend
- Kaufvorgang ohne AJAX
- Kundenbindung (Zahlungs- und Loginerinnerungen, eingeloggt bleiben, Produktbewertung ohne Kommentar)
- Widgets – interaktive Werbemittel
- Bookmarking – Produktwünsche einfach als Bookmarks verwalten
- Erweiterte Bildfunktionalitäten (versch. Größen, Teaser)
- Preisbock (Produkte in Sidebar, Produktarchiv)
 - Liveshopping (Erster Käufer, Fakelkommentare, Verfügbarkeitsbalken, Countdown)

Anlagenverzeichnis

Anlage	Inhalt
Anlage 1	SWOT-Analyse potenzieller neuer Basissysteme
Anlage 2	Normalisierte Datenbankstruktur des Preisbocks
Anlage 3	Vergleich möglicher neuer Basissysteme anhand der Features

Quellen

Kürzel	Quelle
[PWi06]	Peter Winkler: „M+T Computerlexikon“, Markt und Technik Verlag, 2000
[TKu07]	Thomas Kudraß, „Taschenbuch Datenbanken“, Hanser-Verlag, 2007
[MFo00]	Martin Fowler: „Refactoring“, Addison-Wesley-Verlag, 2000
[QPb08]	Quelltext und Datenbank des Preisbocks der dotSource GmbH Stand 29.07.2008 – Revision 2933
[QMa08]	Quelltext des Magento 1.1.2, http://www.magentocommerce.com/getmagento/1.1.2/magento-1.1.2.zip , Abruf 06.08.2008
[Mag08]	http://www.magentocommerce.com/wiki/magento_database_diagram Abruf 31.07.2008
[ERI08]	ERM der Intershop Infinity Suite 6
[Pre08]	Pressematerial Preisbock, http://www.preisbock.de/wp-content/uploads/2008/05/factsheet_preisbock.pdf , Abruf 10.08.2008

Ehrenwörtliche Erklärung

Ich erkläre hiermit ehrenwörtlich, dass ich meine Praxisarbeit mit dem Thema

„Refactoring der Webapplikation Preisbock in Kombination mit der
Suche nach einer neuen Basisplattform“

ohne fremde Hilfe angefertigt habe,

dass ich die Übernahme wörtlicher Zitate aus der Literatur sowie die Verwendung
der Gedanken anderer Autoren an den entsprechenden Stellen innerhalb der
Arbeit gekennzeichnet habe und dass ich meine Praxisarbeit bei keiner anderen
Prüfung vorgelegt habe.

Ich bin mir bewusst, dass eine falsche Erklärung rechtliche Folgen haben wird.

Ort, Datum

Unterschrift

5 Anlagen

SWOT-Analyse potenzieller neuer Basissysteme

	Stärken	Schwächen	Chancen	Risiken
Intershop 6	<ul style="list-style-type: none"> • erprobt bei riesigen Shops • gute Informationen im Turm erhältlich • Bestätigung der Kooperation dotSource - Intershop 	<ul style="list-style-type: none"> • Garantierte Probleme bei der Portierung vom MySQL zu Oracle, wenn die Anwendung kein MySQL unterstützt • Lizenzkosten 	<ul style="list-style-type: none"> • Nur noch Java als Entwicklungssprache (PB-Entwickler javafreundlich) • Internationalisierung möglich • Weiterentwicklung durch professionelle Programmierer im großen Rahmen 	<ul style="list-style-type: none"> • von einem Fremdanbieter Abhängig • oversized für jetzigen Preisbock
Magento 1.1	<ul style="list-style-type: none"> • einfachere Protierung durch PHP • Magento lässt sich mit einem Wordpress kreuzen (Plugins als Interfaces) 	<ul style="list-style-type: none"> • openSource - Preisbock als Plugin wie bei Wordpress nötig • Standardkaufvorgang AJAX-basiert • neue Pluginsprache muss erlernt werden 	<ul style="list-style-type: none"> • bisher wenige Anbieter, die Support für Magento anbieten (dürften wir das?) • Zend-Framework kann Arbeit erleichtern • zur Verwaltung mehrerer Shops ausgelegt • Weiterentwicklung durch openSource-Gemeinde • Internationalisierung 	<ul style="list-style-type: none"> • Angehängte Strukturen könnten wieder wie bei WP außer Kontrolle geraten • noch relativ jung - viele Plugins sowie die Version 1.1 sind noch im Betastadium • ziemlich Performanceschlucke

			möglich	<ul style="list-style-type: none"> nd • muss durch openSource regelmäßig auf dem neusten Stand gehalten werden • PHP lässt viele Schlußigkeiten leicht zu
dotIndustry	<ul style="list-style-type: none"> • keine Einarbeitung in ein neues System, da inhouse einfach Erfahrungen weitergegeben werden können 	<ul style="list-style-type: none"> • komplette Produkt- und Kaufumgebung sowie einige Statistiken müssen neu geschrieben oder portiert werden 	<ul style="list-style-type: none"> • durch komplett Eigenentwicklung können zusätzliche Module an Kunden verkauft oder bei anderen Anwendungen eingesetzt werden (Synergieeffekte) • Nur noch Java als Entwicklungssprache (PB-Entwickler javafreundlich) • gerade vollführtes Refactoring 	

TABELLE	SPALTENNAME	DATENTYP	NULL	KEY	STANDARDWERT	BESONDERES
mail_log	mail_log_id	int unsigned	NO	PRIMARY		auto_increment
	mail_to	varchar(50)	YES			
	mail_template	varchar(50)	YES			
	mail_subject	varchar(100)	NO			
	mail_text	text	NO			
	timecode	datetime	NO		CURRENT_TIMESTAMP	
	status	varchar(20)	YES			
remind_payment	remind_payment_id	int unsigned	NO	PRIMARY		auto_increment
	allocate_id	int unsigned	NO			
	timecode	datetime	YES		CURRENT_TIMESTAMP	
mailconfig	mailconfig_id	int unsigned	NO	PRIMARY		auto_increment
	mail_nicename	varchar(30)	NO	UNIQUE		
	replay	varchar(50)	YES		support@preisbock.de	
	mailto	varchar(50)	YES			
	subject	varchar(100)	NO			
	message	text	NO			
paypal_transaction	paypal_transaction_id	int unsigned	NO	PRIMARY		auto_increment
	paypal_id	varchar(20)	NO	UNIQUE		
	buy_time	datetime	YES			
	buy_email	varchar(50)	YES			
	buy_firstname	varchar(50)	YES			
	buy_lastname	varchar(50)	YES			
	buy_reason	varchar(100)	YES			
	buy_articlenumber	varchar(50)	YES			
	buy_reference_no	varchar(10)	YES			
	buy_price	decimal(8,2)	YES			
	payer_status	varchar(20)	YES			
	payment_status	varchar(20)	YES		99	
	payment_type	varchar(15)	YES			
	payment_pendingreason	varchar(50)	YES			
affiliate	affiliate_id	int unsigned	NO	PRIMARY		auto_increment
	customer_id	int unsigned	NO		0	
	lastaction	timestamp	YES		CURRENT_TIMESTAMP	
	lastpaytime	timestamp	YES		0000-00-00 00:00:00	
	click	int(10)	NO		0	
	cooperate_perclick	decimal(8,2)	NO		0.00	
	reg	int(10)	NO		0	
	newsletter_reg	int(10)	NO		0	

	cooperate_perleed	decimal(8,2)	NO	0.00		
	cooperate_persale	double(8,2) unsigned	NO	0.00		
	cooperate_percent	tinyint(3) unsigned	NO		0	
	cooperate_absolut	double(8,2) unsigned	NO	0.00		
allocate	allocate_id	int unsigned	NO	PRIMARY		auto_increment
	customer_id	int unsigned	NO			
	product_id	int unsigned	NO			
	order_date	datetime	NO			
	reference_no	int(6) unsigned zerofill	NO		7115	
	affiliate_id	int unsigned	YES			
	booked	int(10) unsigned	NO		0	
	shipping_cost	decimal(8,2)	NO	0.00		
	shipping_mode	varchar(30)	NO			
	cashed_voucher_id	int(10) unsigned	YES			
	firstname	varchar(60)	NO			
	lastname	varchar(60)	NO			
	billing_company	varchar(60)	YES			
	billing_address	varchar(60)	NO			
	billing_city	varchar(20)	NO			
	billing_postcode	varchar(7)	NO			
	billing_country	int unsigned	NO			
	rechnung_firstname	varchar(60)	NO			
	rechnung_lastname	varchar(60)	NO			
	rechnung_company	varchar(60)	YES			
	rechnung_address	varchar(60)	NO			
	rechnung_city	varchar(20)	NO			
	rechnung_postcode	varchar(7)	NO			
	rechnung_country	int unsigned	NO			
	email	varchar(100)	NO			
	tel	varchar(16)	YES			
	comment	longtext	NO			
	sum_price	decimal(8,2)	NO	0.00		
	sent	enum(0,1)	NO		0	
	cancelation	enum(0,1)	NO		0	
	cancelation_reason	varchar(500)	YES			
	packstueck_nr	varchar(20)	YES			
	sent_at	datetime	YES	0000-00-00 00:00:00		
	cancel_at	datetime	YES	0000-00-00 00:00:00		
	pay_at	datetime	YES	0000-00-00 00:00:00		
	edit_at	datetime	YES	0000-00-00 00:00:00		
	edit_by	varchar(20)	YES			
	export_at	datetime	YES	0000-00-00 00:00:00		

	payback_at	datetime	YES	0000-00-00 00:00:00	
	booked_at	datetime	YES	0000-00-00 00:00:00	
	retoure_at	datetime	YES	0000-00-00 00:00:00	
	sent_at_edit	varchar(20)	YES		
	cancel_at_edit	varchar(20)	YES		
	pay_at_edit	varchar(20)	YES		
	export_at_edit	varchar(20)	YES		
	payback_at_edit	varchar(20)	YES		
	booked_at_edit	varchar(20)	YES		
	retoure_at_edit	varchar(20)	YES		
cart	cart_id	int unsigned	NO	PRIMARY	auto_increment
	allocate_id	int unsigned	NO		
	variation_id	int unsigned	NO		
	single_price	decimal(8,2)	NO		
	amount	tinyint(3) unsigned	NO		
country	country_id	int unsigned	NO	PRIMARY	auto_increment
	country_name	varchar(255)	NO		
customer	customer_id	int unsigned	NO	PRIMARY	auto_increment
	shipping_address_id	int unsigned	YES		
	anrede	varchar(4)	YES		
	titel	varchar(20)	YES		
	birthday	date	YES		
	lastname	varchar(60)	YES		
	firstname	varchar(60)	YES		
	email	varchar(100)	NO		
	tel	varchar(16)	YES		
	bank_name	varchar(255)	YES		
	bank_account_owner	varchar(255)	YES		
	bank_account	varchar(30)	YES		
	bank_code	varchar(10)	YES		
	tax_number	varchar(10)	YES		
	edit_at	timestamp	YES	CURRENT_TIMESTAMP	
	edit_from	int unsigned	YES		
	referrer_affiliate_id	int unsigned	YES		
	verify_age	enum(0,1)	NO		0
	last_login	datetime	YES		
	login	varchar(255)	NO		
	pass	CHAR(32)	NO		
	nicename	varchar(255)	NO		
	registerdate	datetime	NO		

	activationkey	varchar(255)	YES			
	status	enum('passive','active','locked')	NO			
	avatar	varchar(1024)	YES			
	signatur	TEXT	YES			
	display_name	varchar(255)	NO			
	forum_status_id	int unsigned	NO		1	
	privileges	TEXT	NO			
address	address_id	int unsigned	NO	PRIMARY		auto_increment
	customer_id	int unsigned	NO			
	address_title	varchar(30)	NO			
	firstname	varchar(60)	NO			
	lastname	varchar(60)	NO			
	company	varchar(60)	YES			
	street	varchar(60)	NO			
	city	varchar(30)	NO			
	postcode	varchar(7)	NO			
	country	int unsigned	NO			
	is_shipping_address	tinyint(1)	NO		0	
	is_billing_address	tinyint(1)	NO		0	
merchant	mercant_id	int unsigned	NO	PRIMARY		auto_increment
	lastname	varchar(30)	NO			
	firstname	varchar(30)	NO			
	address	varchar(100)	NO			
	postcode	varchar(7)	NO			
	city	varchar(20)	NO			
	country_id	int unsigned	NO		1	
	internet	varchar(255)	YES			
	email	varchar(100)	NO			
	tel	varchar(30)	YES			
	fax	varchar(16)	YES			
	mobil	varchar(16)	YES			
	companynname	varchar(40)	NO			
	taxNumber	varchar(15)	NO			
	tac	text	NO			
	logo	varchar(255)	YES			
	bank	longtext	NO			
	depositor	varchar(75)	NO			
	account_number	varchar(15)	NO			
	bank_code	varchar(20)	NO			
	bank_name	varchar(75)	NO			
	reason	varchar(100)	YES			

	IBAN	varchar(50)	NO			
	SWIFT	varchar(50)	NO			
	checkout	enum(0,1)	NO		0	
	kredit	enum(0,1)	NO		0	
	amex	enum(0,1)	NO		0	
	mastercard	enum(0,1)	NO		0	
	visa	enum(0,1)	NO		0	
	paypal	enum(0,1)	NO		0	
	paypal_email	varchar(100)	YES			
	delivery	enum(0,1)	NO		0	
	prepayment	enum(0,1)	NO		0	
	debit	enum(0,1)	NO		0	
	verified	enum(0,1)	NO		0	
	comment	text	YES			
	service_from	time	YES			
	service_to	time	YES			
	service_tel	varchar(255)	YES			
	service_email	varchar(255)	YES			
	nicename	varchar(255)	NO		unique	
wish	wish_id	int(10) unsigned	NO	PRIMARY		auto_increment
	description	varchar(255)	NO			
	date	datetime	NO		CURRENT_TIMESTAMP	
wish2tag	wish2tag_id	int unsigned	NO	PRIMARY		
	wish_id	int unsigned	NO			
	tag_id	int unsigned	NO			
tag	tag_id	int unsigned	NO	PRIMARY		auto_increment
	label	varchar(255)	NO			
	nicename	varchar(255)	NO			
bockmark	bockmark_id	int unsigned	NO	PRIMARY		auto_increment
	customer_id	int unsigned	NO		0	
	title	varchar(255)	NO			
	bockmark_url_id	int unsigned	NO			
	description	longtext	YES			
	public	enum(0,1)	NO		1	
	price	decimal(8,2)	NO	0.00		
	shipping_costs	decimal(8,2)	NO	0.00		
	picture_url	varchar(255)	YES			
	date	datetime	NO			

bookmark2bookmark_option	bookmark2bookmark_option_id	int unsigned	NO	PRIMARY		
	bookmark_id	int unsigned	NO			
	bookmark_option_id	int unsigned	NO			
bookmark2tag	bookmark2tag_id	int unsigned	NO	PRIMARY		
	bookmark_id	int unsigned	NO			
	tag_id	int unsigned	NO			
bookmark_options	bookmark_option_id	int unsigned	NO	PRIMARY		auto_increment
	nicename	varchar(255)	NO			
	label	varchar(255)	NO			
product	product_id	int unsigned	NO	PRIMARY		auto_increment
	tva_id	int unsigned	YES			
	mercant_id	int unsigned	NO			
	price	decimal(8,2)	NO			
	upv	decimal(8,2)	YES			
	orig_stock	int unsigned	NO			
	disposable	int(11)	NO		0	
	tax	tinyint(4)	NO		19	
	checkout_url	varchar(255)	NO			
	long_title	text	NO			
	cooperate_persale	double(8,2) unsigned	NO	0.00		
	cooperate_percent	double(4,2) unsigned	NO	0.00		
	cooperate_absolut	double(8,2) unsigned	NO	0.00		
	forced_register	enum(0,1)	NO		0	
	show_progressbar	enum(0,1)	NO		1	
	maxorderamount	mediumint(9)	NO		5	
	show_textlink	enum(0,1)	NO		0	
	show_vendor	enum(0,1)	NO		1	
	link_logo	enum(0,1)	NO		1	
	today_nobock	enum(0,1)	NO		0	
	shipping_days	mediumint(9)	YES			
	further_mailinfos	text	YES			
	nicename	varchar(255)	NO			
	author	int unsigned	NO			
	publish_time	datetime	NO			
	content_technical	TEXT	YES			
	content_product	TEXT	YES			
	product_picture	int unsigned	NO			
	title	varchar(255)	NO			
	nicename	varchar(255)	NO			
	status	enum('draft','publish')	NO			

	newsletter_text	TEXT	NO			
	newsletter_html	TEXT	NO			
	newsletter_subject	varchar(255)	NO			
	variation_name	varchar(50)	YES			
	tva_lager	int signed	YES			
	progressbar1time	int(11)	YES			
	progressbar2time	int(11)	YES			
	progressbar2percent	tinyint(4)	YES			
bockmark_url	bockmark_url_id	bigint(20)	NO	PRIMARY		auto_increment
	url	varchar(255)	NO			
variation	variation_id	int unsigned	NO	PRIMARY		auto_increment
	product_id	int unsigned	NO			
	name	varchar(50)	NO			
	price	float(8,2)	NO			
	disposable	int(10)	NO			
remind_activate	remind_activate_id	int unsigned	NO	PRIMARY		auto_increment
	customer_id	int unsigned	NO			
	remind_date	datetime	NO			
remind_login	remind_login_id	int unsigned	NO	PRIMARY		auto_increment
	customer_id	int unsigned	NO			
	remind_date	datetime	NO			
save_mailform	IP	varchar(15)	NO	PRIMARY		
	UTC	int(10) unsigned	NO		0	
	count	tinyint(4)	NO		0	
stats_feed	stats_feed_id	bigint(20) unsigned	NO	PRIMARY		auto_increment
	product_id	int unsigned	NO			
	time	datetime	NO			
stats_friends	stats_friend_id	int unsigned	NO	PRIMARY		auto_increment
	efrom	varchar(40)	NO			
	time	datetime	NO			
	eto	varchar(40)	NO			
	message	text	YES			
stats_newsletter	stats_newsletter_id	int unsigned	NO	PRIMARY		auto_increment
	product_id	int unsigned	NO			
	customer_id	int unsigned	NO			

	time	datetime	NO			
stats_newsletter_archive	id	int(10)	NO	PRIMARY		auto_increment
	date	date	YES	UNIQUE		
	h09	varchar(4)	NO			
	h10	varchar(4)	NO			
	h11	varchar(4)	NO			
	h12	varchar(4)	NO			
	h13	varchar(4)	NO			
	h14	varchar(4)	NO			
	h15	varchar(4)	NO			
	h16	varchar(4)	NO			
	h17	varchar(4)	NO			
	h18	varchar(4)	NO			
	h19	varchar(4)	NO			
	h20	varchar(4)	NO			
	h21	varchar(4)	NO			
	h22	varchar(4)	NO			
	h23	varchar(4)	NO			
	h00	varchar(4)	NO			
	h01	varchar(4)	NO			
	h02	varchar(4)	NO			
	h03	varchar(4)	NO			
	h04	varchar(4)	NO			
	h05	varchar(4)	NO			
	h06	varchar(4)	NO			
	h07	varchar(4)	NO			
	h08	varchar(4)	NO			
stats_newsletter_archive_buy	id	int(10)	NO	PRIMARY		auto_increment
	product_id	int unsigned	NO			
	date	datetime	NO			
	customer_id	int unsigned	NO			
stats_referer	stats_referer_id	bigint(20)	NO	PRIMARY		auto_increment
	referer	longtext	NO			
	reg_user	bigint(10)	NO		0	
	login_user	bigint(10)	NO		0	
	come_user	bigint(20)	NO		0	
stats_user	stats_user_id	bigint(20)	NO	PRIMARY		auto_increment
	customer_id	int unsigned	YES		NULL	
	referer	longtext	NO			

	date	datetime	NO		CURRENT_TIMESTAMP	
stats_vendorlink	stats_vendorlink_id	int(10) unsigned	NO	PRIMARY		auto_increment
	customer_id	int unsigned	YES		NULL	
	mercant_id	int unsigned	NO			
	timecode	datetime	NO			
tva_article	tva_article_id	int(10)	NO	PRIMARY		auto_increment
	uhrzeit	datetime	NO			
	lfndnr	int unsigned	NO	UNIQUE		
	product_id	int unsigned	NO			
tva_exports	tva_exports_id	int(10)	NO	PRIMARY		auto_increment
	tva_article	int(10)	NO			
	uhrzeit	datetime	NO			
	lfndnr	int unsigned	NO			
	product_id	int unsigned	NO			
	allocate_id	int unsigned	NO			
voucher	voucher_id	int unsigned	NO	PRIMARY		auto_increment
	code	varchar(8)	NO	UNIQUE		
	value	float(6,2)	NO			
	create_date	date	NO			
	end_date	date	NO			
	voucher_campaign_id	int(11) unsigned	YES			
	universal	enum(0,1)	NO		0	
	cashed	enum(0,1)	YES			
voucher_campaign	voucher_campaign_id	int unsigned	NO	PRIMARY		auto_increment
	name	varchar(30)	NO			
	voucher_count	int(11)	NO			
	voucher_remain	int(11)	NO			
	voucher_cashed	int(11)	NO			
	beginn_date	date	NO			
	end_date	date	NO			
	value	float(6,2)	NO			
	note	varchar(1024)	YES			
	active	enum(0,1)	NO		0	
	universal	enum(0,1)	NO		0	
voucher_sendto_user	voucher_sendto_user_id	int(10) unsigned	NO	PRIMARY		auto_increment
	customer_id	int(10) unsigned	NO			
	voucher_id	varchar(8)	YES			

	time_to_send	date	NO			
	reference_no	int(6) unsigned zerofill	NO			
voucher_transaction	voucher_transaction_id	int unsigned	NO	PRIMARY		auto_increment
	voucher_id	int unsigned	NO			
	product_id	int unsigned	NO			
	customer_id	int unsigned	NO			
	cashed_date	datetime	NO			
category	category_id	int unsigned	NO	PRIMARY		auto increment
	category_name	varchar(255)	NO			
	nicename	varchar(255)	NO			
product2category	product2category_id	int unsigned	NO	PRIMARY		
	category_id	int unsigned	NO			
	product_id	int unsigned	NO			
forum_forum	forum_forum_id	int unsigned	NO	PRIMARY		auto_increment
	name	varchar(255)	NO			
	nicename	varchar(255)	NO			
	description	varchar(255)	NO			
	views	int unsigned	NO		0	
	sort	int unsigned	NO			
	introduction	text	YES	NULL		
forum_product2thread	forum_product2thread_id	int(10) unsigned	NO	PRIMARY		
	product_id	int(10) unsigned	NO			
	forum_thread	int(10) unsigned	NO			
forum_post	forum_post_id	int unsigned	NO	PRIMARY		auto_increment
	text	longtext	YES			
	forum_thread_id	int unsigned	NO			
	date	datetime	NO	CURRENT_TIMESTAMP		
	customer_id	int unsigned	YES	NULL		
	subject	varchar(255)	NO			
	views	int unsigned	NO	0		
	approved	enum(0,1)	NO		0	
	top	enum(0,1)	NO		0	
	IP	varchar(15)	YES	NULL		
forum_status	forum_status_id	int unsigned	NO	PRIMARY		auto_increment
	name	text	NO			
	minposts	int unsigned	NO			

	numberpic	int(1)	YES			
	comment	text	NO			
forum_thread	forum_thread_id	int unsigned	NO	PRIMARY		auto_increment
	forum_forum_id	int unsigned	NO			
	views	int(11)	NO			
	subject	varchar(255)	NO			
	nicename	varchar(255)	NO			
	date	datetime	NO		0000-00-00 00:00:00	
	status	varchar(20)	NO		open	
	starter	int unsigned	NO			
newsletter_only	newsletter_only_id	int unsigned	NO	PRIMARY		auto_increment
	email	varchar(100)	NO	UNIQUE		
	regist	date	NO		0000-00-00	
	unregist	date	YES			
	IP	varchar(15)	NO			
	validate_hash	varchar(20)	YES			
	lastletter	int(10) unsigned	NO			0
	type	enum('text','html')	NO		text	
stats_newsletter_send	stats_newsletter_send_id	int(10) unsigned	NO	PRIMARY		auto_increment
	product_id	int(10) unsigned	NO			
	date	datetime	NO		0000-00-00 00:00:00	
	recipient	int(10) unsigned	NO			0
	duration	float(8,4)	NO		0.0000	
option	option_id	int unsigned	NO	PRIMARY		auto_increment
	option_name	varchar(64)	NO	UNIQUE		
	option_value	longtext	NO			
	option_description	tinytext	NO			
parcelservice	parcelservice_id	int(10) unsigned	NO	PRIMARY		auto_increment
	parcel_name	varchar(255)	NO			
	nicename	varchar(255)	NO			
	delivery	double(8,2)	NO		0.00	
attachment	attachment_id	int unsigned	NO	PRIMARY		auto increment
	author	int unsigned	NO			
	timecode	datetime	NO			
	nicename	VARCHAR(255)	NO			
	path	VARCHAR(255)	NO			
	mime_type	VARCHAR(255)	NO			

	width	tinyint unsigned	YES			
	height	tinyint unsigned	YES			
product2parcelservice	product_id	int(10) unsigned	NO	PRIMARY		
	parcel_id	mediumint(9)	NO	PRIMARY		
	shipping_costs	decimal(6,2)	NO		0.00	
	country_id	int unsigned	NO			1
	delivery	decimal(8,2)	NO			
rating	rating_id	int unsigned	NO	PRIMARY		auto_increment
	rating_product	int unsigned	NO			
	rating_rating	int(2)	NO			
	rating_timestamp	datetime	NO			
	rating_ip	varchar(15)	NO			
	rating_host	varchar(200)	NO			
	rating_customer	int unsigned	YES		NULL	
vendor2parcelservice	vendor2parcelservice_id	int unsigned	NO	PRIMARY		auto_increment
	mercant_id	varchar(25)	NO			
	parcel_id	int unsigned	NO			
	standard	enum(0,1)	NO			0
	default_charge	decimal(8,2)	NO		0.00	
	country_id	int unsigned	NO		0.00	
	default_delivery	decimal(8,2)	NO			
product2attachment	product2attachment_id	int unsigned	NO	PRIMARY		
	product_id	int unsigned	NO			
	attachment_id	int unsigned	NO			
pages	page_id	int unsigned	NO	PRIMARY		auto_increment
	author	int unsigned	NO			
	timecode	datetime	NO		CURRENT_TIMESTAMP	
	content	TEXT	YES			
	title	VARCHAR(255)	YES			
	nicename	VARCHAR(255)	YES			
	status	enum(draft, publish)	NO			

Preisbock-Feature	Wicht.	P	Intershop Suite 6	E S	Magento 1.1	E S	dotIndustry	E S
Basic Features			Summe:	71		70		48
DBMS	hoch	3	Oracle	1 3	MySQL	2 6	MySQL	2 6
Suchmaschinenfreundliche URLs	hoch	3		2 6		2 6		2 6
automatisches Erstellen versch. Bildgrößen	hoch	3		1 3	Größen anpassen	1 3	Größen anpassen	1 3
Gutscheinssystem	mittel	2		2 4		2 4		0 0
Kaufvorgang	hoch	3		2 6	Standard AJAXbasiert	1 3		0 0
Kreditkartenzahlung	mittel	2	via Saferpay	1 2	alternative Anbieter	2 4		0 0
Mailtemplates für Text- und HTML-Mails	hoch	3		2 6		2 6		2 6
Nutzergruppen (User und Admins)	hoch	3		3 9		3 9		3 9
Nutzerlogin	hoch	3		2 6	Benutzername ist E-Mail	1 3		2 6
Paypalanbindung	mittel	2		2 4		2 4		0 0
Spezifische Platzhalter in Mailtemplates	hoch	3		0 0		0 0		2 6
Templates	hoch	3		2 6		2 6		2 6
versch. Adressen pro Benutzer	mittel	2		2 4		2 4		0 0
versch. Bezahlungssysteme (nachn., vork., kredit.)	hoch	3		2 6		2 6		0 0
volle HTTPS-Unterstützung	hoch	3		2 6		2 6		0 0

Features Frontend - Allgemein	Wicht.	P	Summe:	41	49	30
Ausverkauft- Verpasstlabel	hoch	3		0 0	0 0	0 0
Bigimage	mittel	2		0 0	Bildzoom	1 2
Double-opt-in bei Nutzerregistrierung	hoch	3		2 6		2 6
Eingeloggt bleiben	niedrig	1		0 0		0 0
Empfehlungsfunktionalität	mittel	2		2 4		2 4
Fake Kommentare	niedrig	1		0 0		0 0
First Sukker (mit Fake)	mittel	2		0 0		0 0
Forum	hoch	3		0 0	muss es als Plugin geben	1 3
Jeden Tag ein neues Produkt (Produktrotation)	hoch	3		1 3	Start-, Endzeit festlegbar	1 3
Kommentar bei Bestellung	mittel	2		2 4		0 0
letzten x Produkte (Sidebar)	hoch	3		0 0		0 0
Lieferantenbewerbung	hoch	3		0 0		0 0
Nutzerregistrierung	hoch	3		2 6		2 6
Nutzerstatus im Forum (z.B. Kleiner Bock)	mittel	2		0 0		0 0
Passwort vergessen	hoch	3	nach Sicherheitsfrage	1 3		2 6
Produktarchiv	hoch	3		0 0		0 0
Produktbewertung	mittel	2	nur mit Kommentar	1 2	nur mit Kommentar	1 2
Rabatt-Teaser	hoch	3		2 6		2 6

RSS-Feed	mittel	2	0 0	2 4	2 4	2 4
Variationen im Kaufvorgang	hoch	3	2 6	2 6	2 6	0 0
Verfügbarkeitsbalken (gefakt)	hoch	3	0 0	0 0	0 0	0 0
Wiederrufsb. und AGB des Händlers in Bestellmail	hoch	3	0 0	0 0	0 0	0 0
Wunschliste	niedrig	1	im Kundenbereich	1 1	im Kundenbereich	1 1

Features Frontend - Mein Bock u.ä.		Summe:	24	28	12
Adressen ändern	hoch	3	2 6	2 6	nur eine Adresse 1 3
Bookmarkfunktionalität	niedrig	1	0 0	0 0	0 0
erweitertes Userprofil(Profilbild, Signatur)	mittel	2	0 0	0 0	0 0
Kunden können selbst stornieren	mittel	2	0 0	0 0	0 0
Kundenbereich	hoch	3	2 6	2 6	1 3
nicht aktivierte User werden an Login gehindert	hoch	3	0 0	0 0	0 0
Passwort ändern	hoch	3	2 6	2 6	2 6
Preisbock-Widget-Unterstützung	hoch	3	0 0	0 0	0 0
Übersicht über Bestellungen mit Status	hoch	3	2 6	2 6	0 0
Wahl zwischen HTML-und Textnewsletter	mittel	2	0 0	Einstellungssache	2 4

Features Backend		Summe:	111	117	16
Angabe einer UVP	hoch	3	2 6	2 6	0 0
begrenzte Stückmenge	hoch	3	2 6	2 6	0 0
Produkt kann ausverkauft gesetzt werden	hoch	3	2 6	2 6	0 0
Browser Backoffice	mittel	2	nur IE	1 2	3 6
flexible MwSt bzw MwSt-Gruppen	niedrig	1	2 2	2 2	0 0
Newsletter zu jedem Produkt (TXT/HTML)	mittel	2	eCircle-messenger	1 2	0 0
Kunde erstellen / bearbeiten	niedrig	1	2 2	2 2	2 2
Kunde sperren	mittel	2	2 4	nur admin sperrbar	nicht sperrbar 1 2
Kundenverwaltung (neu/bearbeiten)	hoch	3	2 6	2 6	0 0
Lieferantenverwaltung (neu/bearbeiten)	hoch	3	2 6	2 6	0 0
Lieferung und Versandkosten in versch. Länder	hoch	3	3 9	3 9	0 0
Max Bestellmenge pro Kunde	hoch	3	2 6	2 6	0 0
Paketdienste	hoch	3	2 6	2 6	0 0
Produkt anlegen/bearbeiten	hoch	3	3 9	3 9	0 0
Produkt kopieren	hoch	3	2 6	2 6	0 0
Produkt nach 24h automatisch nicht mehr verfügbar	mittel	2	0 0	Produktendzeit einstellbar	2 4
Produkt zu bestimmter Zeit online stellen	hoch	3	2 6	2 6	0 0

Übersicht über Verkäufe

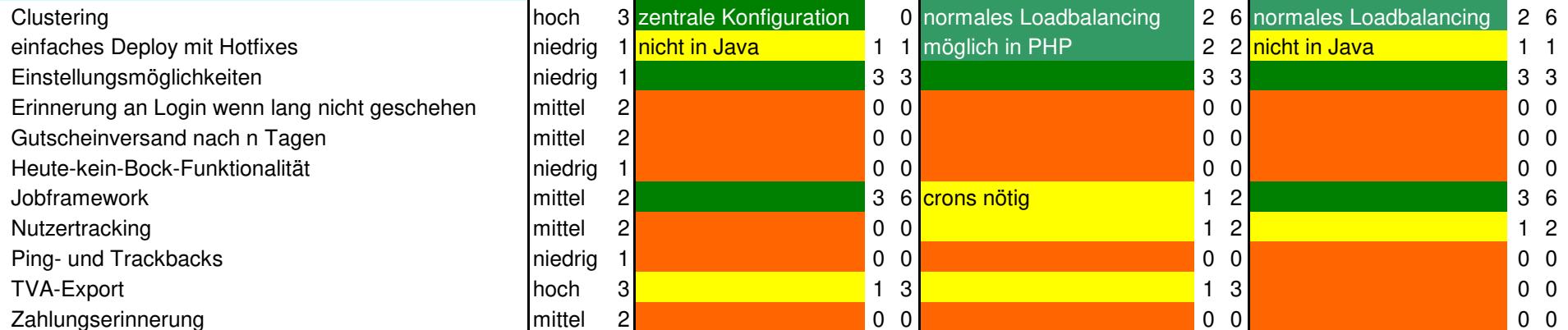
Unterstützung von Variationen

Verkaufsstatistiken (siehe unten)

Vorschaufunktion für Produkt



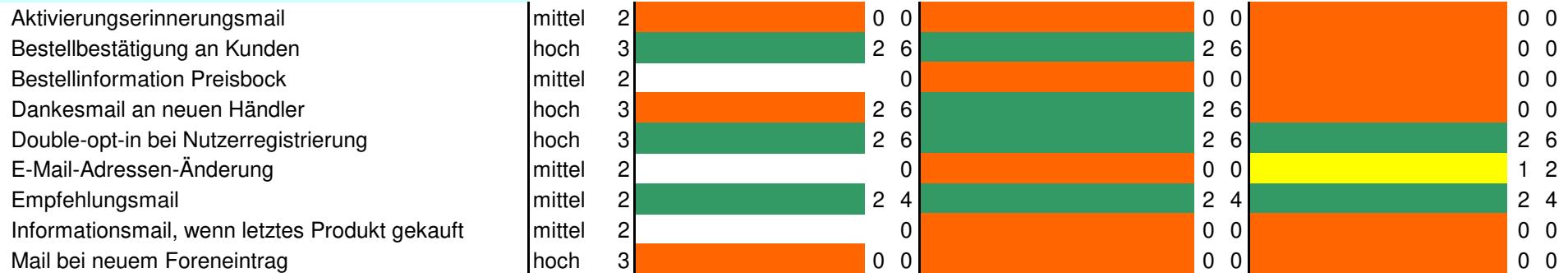
diverse Features



zusätzlich zum Preisbock



Mails



Mail mit neuem Passwort	hoch	3	2	6	2	6	2	6	2	6
Mails mit Gutscheincode	mittel	2	0	0	0	0	0	0	0	0
Newsletter in Text und HTML	hoch	3	0	0	2	6	1	3		
Stornierung im Customerservicecenter	mittel	2	0	0	0	0	0	0	0	0

Bestellstatus	Summe:									
Bezahlt	hoch	3	New							
Exportiert	mittel	2	Confirmed							
Versendet	hoch	3	Rejected							
Storniert	hoch	3	Partially Shipped							
Retoure	mittel	2	Shipped							
Zurückerstattet	mittel	2	Open							
Gebucht	mittel	2	Completed							
										nichts vorhanden

Statistiken / Tracking	Summe:			0	12	0				
Benutzer - Statistik - Käufer	hoch	3	0	0	2	6	0	0	0	0
Klicks im Feed und was wurde dadurch verkauft	mittel	2	0	0	0	0	0	0	0	0
Übersicht für's Controlling	hoch	3	0	0	2	6	0	0	0	0
Von welchem Affiliate haben sich vieviel Registriert	hoch	3	0	0	0	0	0	0	0	0
Von welchem Affiliate haben wieviel gekauft.	hoch	3	0	0	0	0	0	0	0	0
Von welchem Affiliate sind wieviel wann gekommen	niedrig	1	0	0	0	0	0	0	0	0
Wer hat im NL geklickt und gekauft	mittel	2	0	0	0	0	0	0	0	0
Wer hat wann welchen Eintrag im NL geklickt	niedrig	1	0	0	0	0	0	0	0	0
Wer kam vom Affiliate und hat gekauft	hoch	3	0	0	0	0	0	0	0	0
Wieviele haben auf Händlerlink geklickt	niedrig	1	0	0	0	0	0	0	0	0

Gesamtsumme: 300 342 145

Legende:

3 - besser
2 - vorhanden
1 - mit Anpassung
0 - nicht vorhanden