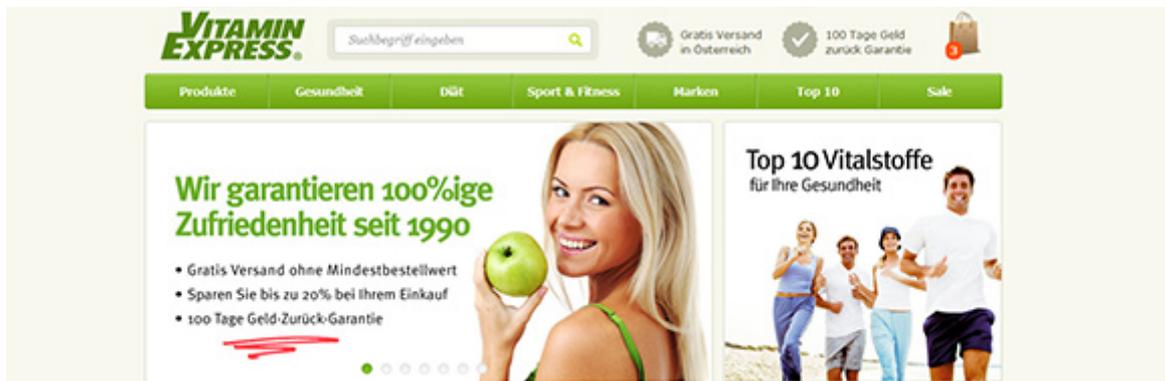


☰ Zeige Kategorien

Case Study: VitaminExpress Webshop mit Magento



🕒 13.03.2014 von [Matthias Zeis](#)



VitaminExpress ist bereits seit 1990 im Markt der Nahrungsergänzungsmittel tätig und bietet seinen KundInnen aus aller Welt neben ausgezeichnetem Kundenservice ausführliche Informationen zu seinen Produkten, um Transparenz zu gewährleisten und bei der Auswahl zu unterstützen. Mehr als hunderttausend Kunden sind VitaminExpress treu und haben mehr als 5 Millionen Produkte bestellt.

Nach mehreren Jahren mit einer Java-Lösung entschied sich der Betreiber die nächsten Schritte mit **Magento** zu setzen. LimeSoda übernahm die technische Umsetzung.

Enterprise Edition

Der Webshop setzt auf die Magento **Enterprise Edition** in der (zum Zeitpunkt dieses Blogposts) aktuellen Version 1.13. Besonders die umfangreichen Möglichkeiten der **Kundensegmentierung** für eine gezielte Zielgruppenansprache haben sich bewährt.

Mehrsprachigkeit

Die Website ist auf **Deutsch und Englisch** erreichbar. Alle Inhalte werden in beiden Sprachen vorgehalten. Die von den BesucherInnen bevorzugte Sprache wird automatisch erkannt und die KundInnen auf die jeweilige Version weiter geleitet.

Länderspezifische Anpassung

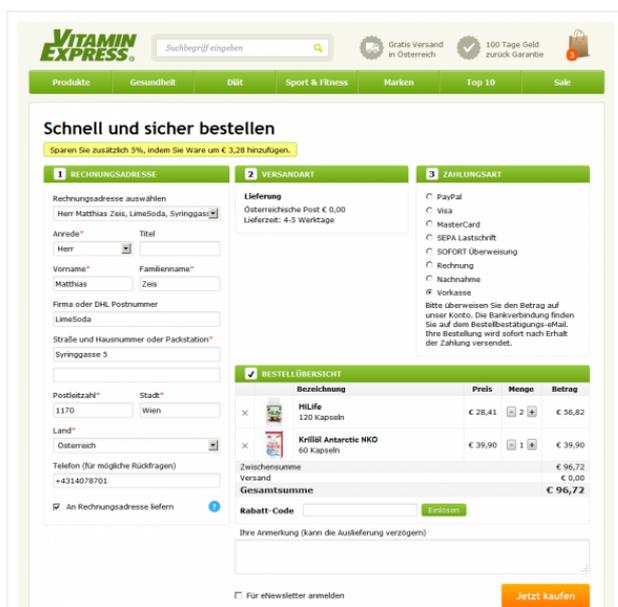
Um den Webshop-KundInnen die passenden Informationen zu präsentieren wurde eine **GeoIP-Funktionalität** implementiert. Mittels GeoIP wird das Herkunftsland der Kundin / des Kunden ermittelt und die entsprechenden Informationen (Lieferkosten, Hotline-Nummern uvm.) dynamisch angepasst.

Kunden

Im Online-Store werden nicht nur die normalen Endkunden sondern auch Geschäftskunden betreut. Über eine **Anbindung zu Amazon** werden zudem die Bestellungen von ebendort über den Webshop abgewickelt.

Suche

Die Suchfunktionalität baut auf der **Solr-Anbindung** der Enterprise Edition auf und bietet den Kunden wesentlich **bessere Suchergebnisse** als die Standard-Suche der Community Edition.



Verbesserter Checkout

Gegenüber dem Magento-Standard-Checkout offenbart der **angepasste Bezahlvorgang** einen noch besseren Überblick über alle für die Bestellung relevanten Daten. Die KundInnen haben alles im Blick. Der Service wird von den KundInnen geschätzt und erhöht zur Freude des Betreibers die Konversionsrate.

Werden Rechnungs- und Lieferadressen oder die Versandart angepasst, dann

werden die restlichen Daten (z.B. die Zahlungsarten und die Kosten) in Echtzeit angepasst, ohne dass die Seite neu geladen werden muss.

Zahlungsarten

Der Shop bietet **alle gängigen Zahlungsarten** von Vorkasse über diverse gängige Anbieter (Kreditkarte, PayPal, Lastschrift, Sofortüberweisung) bis hin zu Rechnung und Nachnahme. Die große Auswahl reduziert die Abbruchrate und führt so zu höheren Umsätzen.

Callcenter-Funktionalität

Die Möglichkeit für **Callcenter**-MitarbeiterInnen, über das Backend Bestellungen anzulegen ist eine wichtige Funktion, weil viele Menschen bei der Hotline anrufen, um sich von ausgebildetem Personal beraten zu lassen und in diesem Zuge gleich die Bestellung aufzugeben. Um die Anlage von Bestellungen zu beschleunigen wurden diverse Verbesserungen des User-Interfaces durchgeführt.

Angepasstes Backend

Das Backend wurde entsprechend der projektspezifischen Anforderungen angepasst.

Aufstellungen (so genannte „**Grids**„) für Bestellungen, Rechnungen, Kunden etc. wurden **angepasst**, um genau die nötigen Informationen anzuzeigen und nach diesen zu filtern und zu sortieren. Bestimmte Werte (z.B. bestimmte Kundengruppen) werden optisch hervorgehoben, um die Arbeit der MitarbeiterInnen zu erleichtern.



Da das Magento-Backend bei größeren Datenmengen in Sachen Geschwindigkeit zu wünschen übrig lassen kann (besonders, wenn verschiedene Datenarten wie Kunden, Bestellungen, Rechnungen und Lieferadressen zusammengeführt werden) wurden **Performance-Optimierungen** durchgeführt. Informationen, die sonst bei jedem Seitenaufruf dynamisch zusammengesucht werden müssen werden vorab zusammengefasst und führen so zu einer schnelleren Auslieferung.

Bestellungs-Management

Um die Bestellungen so schnell wie möglich abzuwickeln werden alle Orders **vollautomatisch** anhand diverser **Regeln** überprüft. Sind alle Voraussetzungen erfüllt, dann geht die Bestellung direkt in das **Fulfillment**. Somit kann eine Bestellung im Optimalfall bereits nach wenigen Minuten vom Logistikzentrum verpackt und verschickt werden.

Falls eine **manuelle Nachbearbeitung** nötig ist werden die Bestellungen für die zuständigen MitarbeiterInnen zur Seite gelegt und mit Kommentaren versehen, die **Hinweise** auf die zu tätigenden Schritte liefern.

Der **Standard-Magento-Order-Flow** wurde in vielerlei Hinsicht an die Prozesse von VitaminExpress **angepasst**. Spezielle Buttons und Funktionen unterstützen die effiziente Umsetzung dieser Prozesse.

ERP-Funktionalitäten

In diesem Projekt dient Magento nicht als „bloßer“ Webshop, sondern **ersetzt das ERP-System**.

Für den Einkauf und die Lagerhaltung

Der Einkauf wird durch Tools zur **Bedarfsermittlung** (Vorhersage der Entwicklung der Umsätze) und den **Beschaffungsprozess** (Erstellung von Bestellungen bei den Lieferanten) unterstützt.

Bei Wareneingängen können die **Lieferungen** direkt in Magento eingebucht werden. Die **Lagerhaltung** wird komplett in Magento abgewickelt und ständig mit dem Logistikzentrum synchronisiert.

Für das Marketing

Das Marketing profitiert von den fortgeschrittenen Möglichkeiten der

Kundensegmentierung in der Enterprise Edition. Durch Export-Möglichkeiten der Kontaktlisten sowie die Anbindung an einen externen **Newsletter-Anbieter** können Kunden schnell und präzise adressiert werden. Auch Banner-Schaltungen können für Kundensegmente vorgenommen werden.

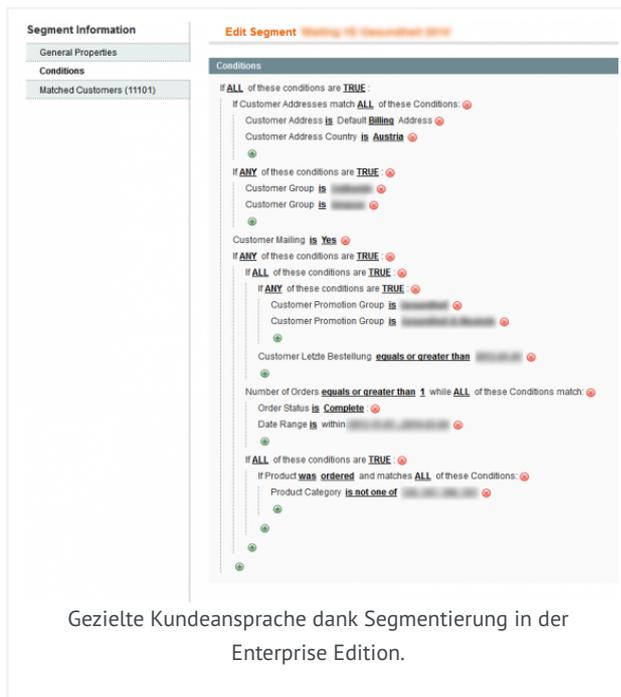
The screenshot shows the 'Order Stop Rules' configuration page in Magento. It is divided into two main sections: 'General' and 'Revised status'.

General Section:

- Enable Cronjob:** Yes (dropdown)
- Cronjob schedule:** 4,9,14,19,24,29,34,39,44,49,54,59 * * * * * (text input)
- Process orders with current status:** A list of order statuses including 'Out of Stock', 'Payment Review', 'PayPal Cancelled Reversal', 'PayPal Reversed', 'Pending', 'Pending Acceptance', 'Pending Payment', 'Pending PayPal', 'Out of Stock', 'Processing', 'Released', and 'Shipped'. 'Out of Stock' is currently selected.

Revised status Section:

- Magento qty is < 0 for any order item:** Yes (dropdown)
- Enabled in cronjob:** Yes (dropdown)
- Rule model:** + defined in config.xml as a System + Configuration value (text input)
- Process rule if the current order was created by:** A list of user roles including 'Backend', 'Cronjob', and 'Frontend'. 'Frontend' is currently selected.
- Process rule if the current order status is:** A list of order statuses including 'Out of Stock', 'Payment Review', 'PayPal Cancelled Reversal', and 'PayPal Reversed'. 'Out of Stock' is currently selected.



Durch Erweiterungen wurden weitere Features wie Remarketing-Kampagnen hinzugefügt.

Für das Controlling

Während Magento bereits einige **Reports** über Verkäufe, Produkte und Kunden liefert wurden diese Funktionalitäten konsequent ausgebaut, um Daten für die Steuerung des Unternehmens zu liefern. Zum Beispiel können Verkäufe über Zeiträume, Länder, Order-Status und Quellen gefiltert oder spezielle Produkt- und Lagerstands-

Reports abgerufen werden.

Anbindung an die Logistik

Die **bidirektionale Anbindung** an das externe Logistikunternehmen unterstützt die automatische Abwicklung des gesamten Verpackungs- und Auslieferungsprozesses. Der Rückkanal vom Logistiker zum Webshop gewährleistet, dass die Verfügbarkeiten im Shop aktuell sind, der Kunde über den Fortschritt seiner Bestellung informiert ist und die angeschlossenen Unternehmensbereiche auf dem aktuellen Stand sind.

Anbindung an die Buchhaltung

Weiters werden im Hintergrund alle Bestellungen, Rechnungen und Gutschriften mit der **Buchhaltung** abgeglichen. Somit kann die Arbeit mit den spezialisierten Buchhaltungsprogrammen umgesetzt werden.

Generierung von Rechnungs-PDFs

Eine weitere ERP-Aufgabe, welche der Webshop übernimmt ist die **Generierung der PDFs** für Bestellungen, Rechnungen und Gutschriften. Die Anpassung der umfangreichen PDFs aufgrund der Kundengruppen, Zielländer, Zahlungsarten und weiteren Faktoren erfolgt automatisch bei der Übergabe an die Logistik.

Technik zur Unterstützung der Prozesse

Da es sich um ein Magento-Projekt der umfangreicheren Art handelt (unter anderem aktuell 143 Extensions, gut 400.000 Zeilen PHP-Dateien-Code alleine in den `app/code/community-` und `app/code/local-`Verzeichnissen, bis zu 9 Entwicklungsumgebungen parallel), müssen auch die **Prozesse dahinter** stimmen. Daher möchten wir auch hier einen kleinen Einblick gewähren.

Entwicklungsumgebungen, Build und Deployment

Die Arbeit mit **Versionskontrollsystemen** ist für zeitgemäße Entwicklung unabdingbar. Das ist natürlich auch bei uns so. Wir setzen seit einigen Jahren auf **Git** und sind sehr zufrieden damit, weil Projekte dieser Größe für Git kein Problem darstellen. Unser

Prozess ist denkbar pragmatisch: alle EntwicklerInnen sind mündige Menschen und können entscheiden, wann ihr Code in Produktion gehen soll. Bedingung: der master-Branch muss immer deploybar sein und ohne Risiko ins Live-System gehen können.

Für den **Build- und Deployment-Prozess** setzen wir **Phing** in Kombination mit **rsync** ein. Wir verzichten bewusst auf Continuous Deployment im engeren Sinn, also das automatische sofortige Ausliefern von Code ins Live-System. Somit haben wir die endgültige Kontrolle, was wann in Produktion geht. Um eine Änderung online zu stellen, müssen wir nur „deploy“ eingeben und 2 Tasten drücken. Der neue Code ist spätestens eine halbe Minute später live. Der Shop bleibt in dieser Zeit erreichbar, da zuerst die Dateien auf den Live-Server synchronisiert werden und die neue Version über das Umbiegen eines Symlinks online gehen.

Datenbank- und Konfigurationsmanagement

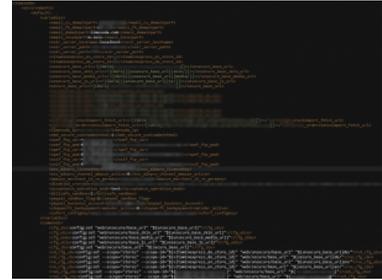
Um in den Entwicklungs-, Test- und Staging-Umgebungen stets mit aktuellen Daten arbeiten und gleichzeitig sicher gehen zu können, dass kein System aus Versehen mit produktiven Schnittstellen kommuniziert, haben wir einige **Tools** im Einsatz:

- **Backporting:** bei Bedarf kann eine Kopie der Live-Datenbank in jede Umgebung eingespielt werden. Entweder kopiert man sich einen aktuellen Stand der Live-Datenbank oder einen der archivierten Snapshots der Datenbank. Der Vorgang benötigt als Angabe nur die Zielumgebung und die gewünschte Datenbankkopie. Der Rest (Erstellung des Datenbank-Dumps, Transferierung in die jeweilige Umgebung,

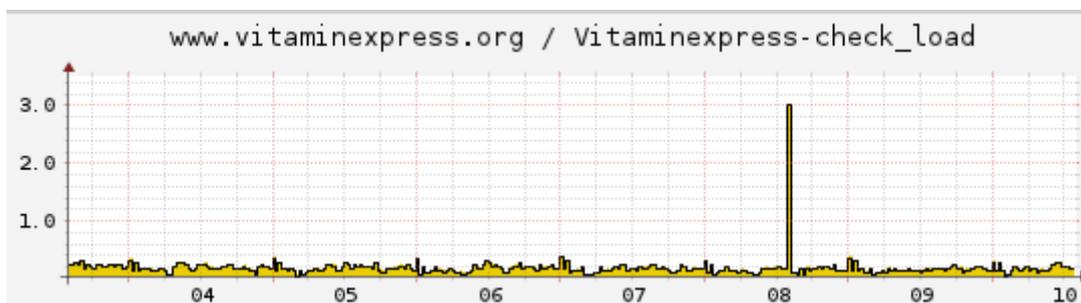
```
magento-deployment > deploy:
    [echo] -----
    [echo] Deployment process completed.
    [echo] -----
BUILD FINISHED
Total time: 17.1528 seconds
```

Einspielung der Datenbank, Anpassung der Konfiguration und Anonymisierung von Daten) erfolgt vollautomatisch.

- **LimeSoda_EnvironmentConfiguration:** diese Extension ist dafür zuständig, die Konfiguration und Daten in einer Umgebung entsprechend der jeweiligen Anforderungen anzupassen. Zum Beispiel werden Verbindungsdaten zu externen Systemen (ERP, Payment-Service-Providern, Logistik, FTPs, ...) angepasst, Caches aktiviert/deaktiviert, falls nötig Extensions oder Cronjobs deaktiviert, E-Mail-Adressen angepasst, Kundendaten anonymisiert und vieles mehr. Befehle und Variablen werden im XML einer Magento-Extension definiert. Alles kann hierarchisch organisiert werden, womit der Aufwand für die Anpassung mehrerer Umgebungen minimiert wird. LimeSoda_EnvironmentConfiguration ist in **n98-magerun** eingebettet und kann daher über die Kommandozeile aufgerufen oder in Build- und Deployment-Prozessen eingebunden werden.
- **LimeSoda_LiveGuard:** Datenbanken leicht rückportieren und die Konfiguration anpassen zu können ist praktisch, doch das schützt leider nicht davor, dass eine Umgebung aus Versehen doch falsche Verbindungsdaten enthält. Zu diesem Zweck haben wir den LiveGuard entwickelt. Diese Magento-Extension überprüft bei jedem Aufruf (egal ob über Web, Cronjob, Shell oder Web-API), ob die Werte wie gewünscht gesetzt sind. Das Modul reagiert einerseits sehr defensiv (d.h. es muss explizit definiert werden, falls ein Check in einer Umgebung nicht ausgeführt werden soll – standardmäßig wird alles überall überprüft) und ist andererseits auf hohe Geschwindigkeit ausgelegt. Durch eine Plugin-Architektur können für jedes Projekt spezifische Checks geschrieben werden.



Server-Monitoring

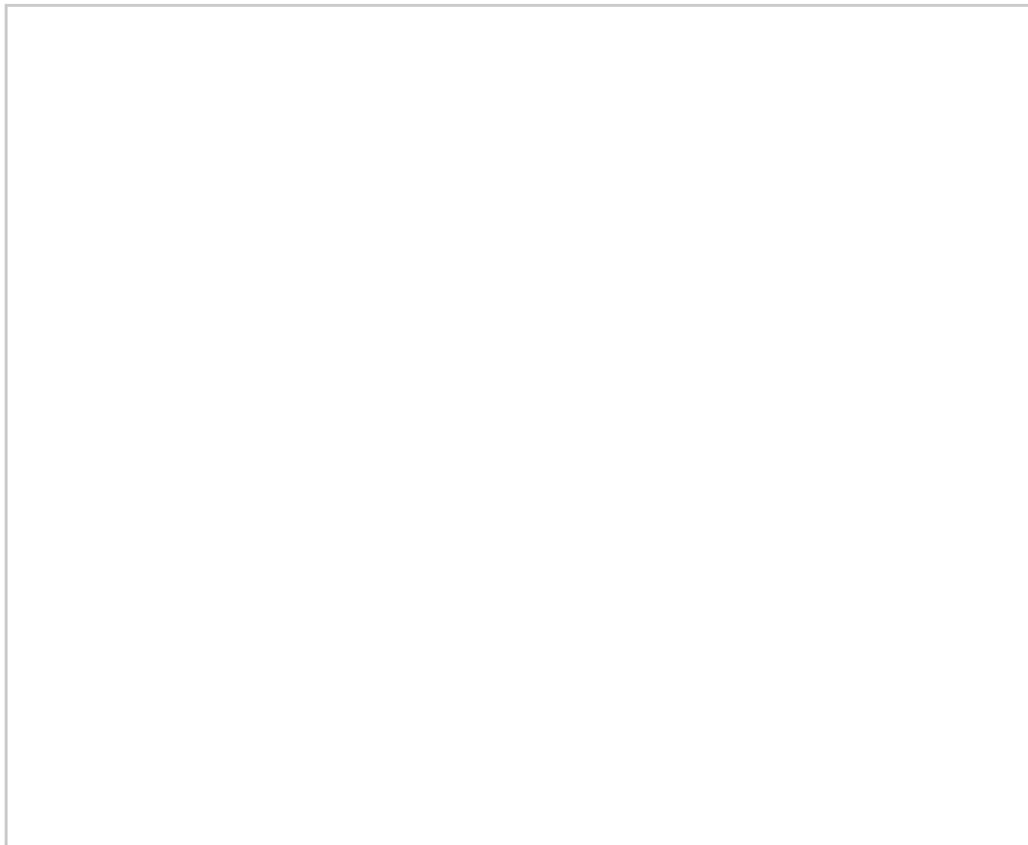


Ein wichtiger Punkt für einen gut laufenden Magento-Shop die beständige **Wartung und Kontrolle des Systems**. Dazu bietet sich ein Monitoring an, das wichtige Parameter überwacht.

Wir verwenden für diesen Zweck das Open-Source-Tool [Icinga](#). Icinga achtet für uns regelmäßig auf die Erreichbarkeit der Website, den Festplattenplatz, die Auslastung des Servers, die Ausführung von Cronjobs, die Größe von Error- und Log-Files, Log- und Report-Tabellen, die Anzahl der Session-Files und einiges mehr. Ziel ist, pro-aktiv zu reagieren wenn etwas aus dem Ruder läuft, Probleme gar nicht erst entstehen zu lassen oder möglichst früh gegenzusteuern.

Präsentation

Man kann aus jedem Projekt neues Wissen und neue Erfahrungen mitnehmen. Unsere Learnings haben wir auf dem [7. Magento-Stammtisch Wien](#) präsentiert. Die [Slides](#) finden Sie bei [slideshare](#):



Learnings aus einem Magento-Enterprise-Projekt von **LimeSoda**



Case Studies

E-Commerce

Entwicklung

Kommentare

Hinterlasse einen Fingerabdruck für die Ewigkeit: Ein Kommentar bei LimeSoda!

NAME *

EMAIL-ADRESSE *

WEBSITE

IHRE NACHRICHT *

(*) Pflichtfeld

AB DAMIT!

KEEP ME INFORMED

Im LimeSoda-Newsletter informieren wir verlässlich über Web-News.

JETZT ANMELDEN

BEWIRB DICH BEI UNS!