

10 Tipps für die Migration deiner Website zu Drupal

Für eine reibungslose Migration deiner Website von einem anderen CMS zu Drupal gilt es einige Stolperfallen zu beachten. Hier findest du praxiserprobten Rat, wie du Content, Nutzer, Oberfläche und Metadaten in zehn Schritten sicher überträgst.

1 Schau dir zuerst genau an, welche Assets du auf deiner bisherigen Website (der Quellsite) hast, wie sie strukturiert sind, wo sie jeweils gespeichert werden, und natürlich welche Elemente, vom Design über die Assets bis zu eventuellen externen Datenbanken, du auf die neue Site (die Zielsite) mitnehmen willst bzw. kannst.

Assets befinden sich üblicherweise an drei Stellen: als Eintrag in Datenbanken, als eigenständige Datei auf einem bestimmten Server und als Element, das an einer Stelle oder in einem Format gespeichert ist, auf das du keinen Einfluss hast. Im Einzelnen gibt es Folgendes zu beachten.

Text-Content

Dazu gehören: Alle Artikel, Blog-Einträge, Seiten und andere Informationen, die dem Nutzer angezeigt werden. Die meisten CMS speichern Text-Content als Einträge mit mindestens einem Titel- und einem Hauptteilfeld. Weitere Felder sind üblich: Eine E-Commerce-Seite zum Beispiel könnte Produktinformationen als Text-Content mit Feldern wie Preis, Größe und verfügbarer Menge speichern. Zusätzlich ordnen die meisten CMS dem Text-Content Metadaten zu wie einen eigenen „Schlüssel“ für jeden Eintrag, Erstellungsdatum, einen Link zum Nutzer, der den Eintrag erstellt hat etc.

In Drupal: In Drupal heißen solche Einträge „Nodes“ und werden in Datenbanktabellen gespeichert, die mit node beginnen. Der Hauptschlüssel des Eintrags befindet sich in der node-Tabelle, während der Content selbst in der node_revisions-

Tabelle gespeichert wird. Drupal hält verschiedene Arten von Content mit Hilfe von „Content-Typen“ auseinander, welche in der `node_type`-Tabelle gespeichert werden. Das Nodetype-Modul bietet ein paar Grundfunktionen, um Content von einem Typ in einen anderen zu verwandeln. Alle Nodes eines bestimmten Content-Typs enthalten den gleichen Satz Felder. In Drupal 7 können nun einfach neue Felder hinzugefügt werden, während man in Drupal 6 dazu das Content Construction Kit (CCK) braucht. Abhängig von deiner Quellsite kann es sein, dass Tabellen zusammengelegt oder aufgeteilt werden müssen. Danach kann der Content mit einem Migrationstool übertragen werden.

Probleme: Fehler passieren oft dann, wenn der Text-Content nicht von Code wie HTML oder PHP befreit wurde. Auch wenn Drupal den Content anhand seines Systems von „Textformaten“ („input formats“ in Drupal 6) filtert, um kein PHP und nur wenige HTML-Tags zuzulassen, ist es ratsam, etwaigen nicht bereinigten Content zu korrigieren, entweder auf der Quellsite, auf der Drupal-Site oder während der Migration.

Weitere Fehlerquellen sind schlecht geformtes HTML und unerwartete Zeichen. Das passiert oft, wenn 8-Bit-Text in ein Format gebracht wird, das diesen nicht versteht, oder in ein anderes Betriebssystem. Tools wie HTML Tidy können diese Probleme oft erkennen, aber du solltest deine Texte trotzdem zumindest anhand von Stichproben überprüfen.

Problematisch kann auch das Ausgangsformat der Texte sein. Während die Migration von MySQL oder PostgreSQL am einfachsten ist, kann es mit anderen Arten von SQL-Daten sowie XML schwieriger werden. Noch komplizierter ist HTML, während nicht-Standard-basierte Formate wie Microsoft Word am schwierigsten sind. Es ist daher empfehlenswert, alle Daten zuerst in MySQL / PostgreSQL zu konvertieren, bevor sie migriert werden. Außerdem solltest du dir für den Fall, dass du mehrere Versuche brauchst oder das Ganze mal wiederholen willst, alle Schritte genau notieren.

Bilder und anderer nicht-textlicher Content

Dazu gehören: Bilder-, Video- und Audiodateien, die oft, aber nicht immer als Einzeldateien auf deinem Server gespeichert und nach Speicherort referenziert sind.

In Drupal: Die Speicherung und Referenzierung dieser Dateien ist in Drupal nicht standardisiert. Zwei Möglichkeiten sind jedoch am beliebtesten. Zum einen kannst du die Dateien einfach in Drupals Datenspeicher hochladen und dann mit HTML referenzieren. Bei der anderen installierst du zusätzlich zum Drupalkern Module wie FileField, ImageField oder ImageAPI für Bilder, mit denen du dies einfacher erledigen kannst. Bei beiden Varianten werden die Dateien selbst fast immer in einem Unterverzeichnis von `/docroot/sites/sitename/files/` gespeichert.

Probleme: Solch nicht-textlicher Content kann auf der Quellsite auf viele verschiedene Unterverzeichnisse verteilt sein, besonders bei einer alten Site oder einer mit vielen Mitwirkenden. Einige Dateien können sich in der Datenbank selbst befinden oder auf einem externen Server wie YouTube.

Wenn du alle Dateien zusammen hast, müssen sie neu referenziert werden, entweder mit HTML oder wie durch die Module vorgegeben, die du für Multimedia benutzen willst.

Nutzerdaten

Dazu gehören: Nutzerdaten ermöglichen die Authentifizierung von Nutzern, um ihnen verschiedene Rechte zuzuordnen und individuelle Nutzer z.B. anhand ihres Namens oder ihrer E-Mail-Adresse wiedererkennen zu können.

In Drupal: Drupal unterscheidet standardmäßig zwischen anonymen Nutzern (die keine Identifizierungsmerkmale angegeben haben) und authentifizierten Nutzern (die mindestens ihren Namen und ihre E-Mail-Adresse angegeben haben). Ein Administrator kann unbegrenzt weitere Nutzer-„Rollen“ festlegen und damit verbundene Nutzungsrechte vergeben.

Drupal speichert grundlegende Nutzerdaten in der users-Tabelle mit einer Nutzer-ID (uid) als Primärschlüssel. Die ersten zwei davon haben Sonderfunktionen: Nutzer 0 sind alle anonymen Nutzer, und Nutzer 1 ist der „Super-Nutzer“, der bei der Installation von Drupal erstellt wird und umfassende Zugriffsrechte hat.

Du kannst das Profile-Modul aktivieren, um den Nutzerprofilen weitere Felder zuzuordnen. Diese werden bei Drupal in den profile_fields und profile_values Tabellen gespeichert.

Probleme: Da die meisten CMS Passwörter verschlüsselt speichern, kannst du sie meist nicht manuell retten, wenn die Passwortmigration misslingt.

Viele Unternehmen speichern Nutzerdaten in einem externen System. Bei der Migration musst du dann die Integration von Drupal mit diesem System einplanen. Diese Situation kommt jedoch so häufig vor, dass dafür Ressourcen wie die Drupal-Module LDAP Integration und LDAP Provisioning zur Verfügung stehen.

Da Tracking und Nutzerauthentifizierung auf Cookies basieren, welche je nach System unterschiedlich sein können, werden die Nutzer sich nach der Migration neu einloggen müssen.

Metadaten

Dazu gehören: Metadaten sind Informationen, die andere Informationen wie Tags, Daten zur Content-Erstellung oder –Veränderung, das letzte Login eines Nutzers etc. beschreiben. Sie beziehen sich immer auf andere Daten.

In Drupal: Metadaten kommen in fast jeder Tabelle der Drupal-Datenbank vor. Sie werden entweder von Drupal oder vom Nutzer erstellt. Dabei kannst du entscheiden, welche Nutzerrollen Metadaten verändern dürfen.

Probleme: Metadaten werden von den CMS sehr unterschiedlich gehandhabt. Manche Flags des einen Systems gibt es in einem anderen gar nicht, oder sie sind auf verschiedene Felder aufgeteilt. Die Entscheidung, welche Metadaten du migrieren willst und wie sie organisiert sein sollen, kann sehr komplex sein.

Designelemente und Benutzeroberfläche

Dazu gehören: Das Design einer Site bestimmt, wie sowohl Administratoren als auch Besucher Informationen finden und erfahren. Es besteht sowohl aus visuellen Elementen wie Grafiken und Layout als auch interaktiven wie Menüs.

In Drupal: Drupal setzt das Design einer Site mit einem „Theme“ um, einer Kombination aus PHP-Dateien für die Seitenlogik und CSS-Dateien für die Darstellung. Normalerweise hat eine Drupal-Site ein einheitliches Theme; du kannst aber auch unterschiedliche Designs für bestimmte Seiten, Content-Typen oder Elemente (z.B. Blogs) festlegen. Design mit Drupal ist eine Wissenschaft für sich. Sollte dir die nötige Erfahrung fehlen, kannst du dich an eine der vielen Design-Agenturen wenden, die mit Drupal arbeiten.

Probleme: Höchstwahrscheinlich werden manche Designelemente sich nicht eins zu eins auf die Zielsite übertragen lassen, da die verschiedenen CMS einfach zu unterschiedlich mit Content umgehen. Du solltest das bisherige Design auf jeden Fall prüfen. Eine Migration ist aber auch ein guter Zeitpunkt für ein neues Design, da dies unter Umständen viel einfacher ist als die Übertragung des alten Designs.

Businesslogik

Dazu gehört: Businesslogik ist ein allgemeiner Begriff für ein System, das aus Daten nützliche Informationen macht. Bei einer Buchbesprechung in einer Zeitschrift etwa können Titel und Autor als separate Objekte gespeichert werden, um sie auf weitere Art und Weise zu nutzen. Dafür müssen diese Informationen natürlich gut referenziert werden.

In Drupal: Drupal kann einige Formen von Businesslogik nativ umsetzen, sei es im Drupal-Kern oder durch Module. Wenn deine Businesslogik von einem externen System kommt, wirst du es wahrscheinlich sehr einfach in deine Drupal-Site integrieren können, auch wenn dafür vielleicht etwas individuelle Programmierung notwendig wird.

Probleme: Businesslogik, die ein fester Bestandteil des Quell-CMS ist, muss eventuell ganz neu programmiert werden, was mit zusätzlichen Kosten verbunden wäre. Entweder kannst du sie als eigenständiges Programm entwickeln, das über ein Protokoll wie SOAP mit Drupal interagiert, oder als natives Drupal-Modul.

Businesslogik, die aus der Struktur einer Site abgeleitet wird, kann sehr kompliziert sein. Du solltest ein Diagramm der Datenobjekte der Quellsite und ihrer Beziehungen erstellen, das dir dabei hilft, die Datenstruktur der Zielsite zu planen. Da es viele Fehlerquellen gibt, solltest du vor der Migration auf jeden Fall Daten mit bereits bekannten Ergebnissen gründlich testen.

2 Verschaffe dir eine Übersicht über alle Funktionen deiner bisherigen Site. Überlege dir, was die Nutzer auf deiner Site machen können. Chat, E-Commerce, Spiele und Social Networking sind alles Funktionen, für die dir eine Vielzahl von Drupal-Modulen zur Verfügung stehen.

3 Nun kennst du deine Quellsite und kannst entscheiden, wie du jeweils mit deinen Assets verfahren willst. Du hast drei grundlegende Möglichkeiten:

1. Du kannst die Assets von der Quellsite herunterladen. Eine Migration ist ideal, um nicht mehr Benötigtes zu entfernen.
2. Du kannst die Assets extern speichern, entweder im alten CMS, einem anderen System oder in einem Remote Service. Dies ist recht üblich. Es kommt dabei darauf an, dass die Verbindung zwischen Drupal und dem externen System schnell und stabil ist und dass Drupal die Assets handhabt wie vorgesehen. Ken Rickard von Palantir empfiehlt, hierfür soweit wie möglich XML-Feeds zu benutzen. Manchmal kann Drupal aber auch direkt auf den Content einer externen Quelle zugreifen.

3. Du kannst die Assets auf die Zielsite verschieben. Manche Assets müssen nur einmal übertragen werden und sind dann dauerhaft auf der Zielsite verfügbar.

4 Bevor du deine Assets nun migrierst, stelle sicher, dass sie alle ordentlich und auf dem neuesten Stand sind. Falls nicht, überlegst du dir, ob du sie am besten vor oder nach der Migration in Schuss bringst.

5 Teste mit einem Ausschnitt der tatsächlichen Daten, wenn deine Datenschutzrichtlinien dies erlauben. Blindtext wie „Lorem Ipsum“ würde eventuelle Datenfehler, die während der Migration auftreten können, nicht sichtbar machen. Der komplette Datensatz wiederum könnte zu umfangreich sein und die Tests hinauszögern.

6 Setze die gewünschten Funktionen auf deiner neuen Drupal-Site um. Alle Module, individuellen Programmierungen und Datenschemata sollten fertig sein, bevor du den Content verschiebst.

7 Informiere alle von der Migration Betroffenen. Auch die besten Designer und Programmierer werden gewisse Unterschiede zu deiner bisherigen Website nicht vermeiden können. Die Nutzer müssen eventuell ein neues Passwort einrichten oder neue Abläufe lernen. Berücksichtige daher die nötige Dokumentation und Training bei deinen Plänen.

8 Verschiebe deine Assets der Reihe nach. Eine Migration sollte in mehreren Abschnitten durchgeführt werden. Joshua Brauer, Drupalist bei Acquia, empfiehlt die folgende Reihenfolge:

1. Nutzer sowie Nutzerrollen und Zugriffsrechte (falls vorhanden). Falls du Nutzerprofile mit dem Content Profile Modul in Nodes speichern willst, kümmerst du dich erst später um die Profile.
2. Taxonomiebegriffe und Vokabular, da du diese dem Content zuordnen musst, wenn du ihn verschiebst.
3. Andere Metadaten.
4. Content.

Über Cocomore

Die Cocomore ist eine Multimedia-Agentur mit gut 100 Mitarbeitern und Büros in Deutschland (Frankfurt/Main und Königswinter) und Bulgarien (Sofia). Mit den zwei Standbeinen Kommunikation und Technologie entwickelt und betreut Cocomore Kommunikations-, Commerce- und CRM-Lösungen. Zu unseren langjährigen Kunden gehören unter anderem Nestlé, Procter & Gamble, Rossmann, RTL, SCA und Sanofi-Aventis. Cocomore hat Projekte in über 30 Ländern realisiert und ist Mitglied im BVDW (Bundesverband Digitale Wirtschaft).

Folgende Drupal-Tools stehen dir unter anderem zur Verfügung:

- **Drush**, <http://drupal.org/project/drush>, eine erweiterbare Kommandozeilen-Oberfläche mit über hundert eigenen Befehlen, erweiterbar durch und kompatibel mit Dutzenden anderen Modulen
- **Migrate**, <http://drupal.org/project/migrate>, eine erweiterbare Umgebung, mit der du Daten in Drupal laden, umwandeln und speichern kannst
- **Feeds**, <http://drupal.org/project/feeds>, migriert Daten, Nutzer und Taxonomiebegriffe aus Dateien oder RSS-Feeds in Nodes.
- **Import HTML**, http://drupal.org/project/import_html, versucht aus statischen HTML-Sites Nodes zu erstellen. Aufwändige Konfiguration, spart aber trotzdem viel Zeit!
- **Node Export**, http://drupal.org/project/node_export, kann Node Content sowohl importieren als auch exportieren. Für einfachere Migrationsaufgaben.
- **Joomla to Drupal**, <http://drupal.org/project/joomla>, verschiebt Content, Benutzer und Taxonomiebegriffe direkt aus der Joomla-Datenbank
- **Wordpress Import**, http://drupal.org/project/wordpress_import, ist ein vielseitiges Tool für Content, Nutzer, Tags und mehr im Format WordPress eXtended RSS (WXS)

9 Plane Probleme mit ein. Migrationen verlaufen kaum reibungslos. Wenn du nach einer Datenverschiebung ein Problem entdeckst, löst du es, probierst es noch einmal und entdeckst vielleicht das nächste Problem, bis es klappt. Du solltest auf jeden Fall immer die Möglichkeit behalten, Veränderungen zurück zu nehmen, um sie korrigieren zu können.

10 Baue Kontakt auf zu einem Drupal-Berater, an den du dich jederzeit wenden kannst. Falls du nicht schon viel Erfahrung mit Drupal hast, stehst du an irgendeiner Stelle vor einem Problem, das du nicht allein lösen kannst. Ein guter Drupal-Berater bringt dich dann ganz schnell weiter. Acquia arbeitet mit Spezialisten von über 150 Drupal-Partnern in aller Welt zusammen, von denen eine bestimmt zu dir passt.

Über Acquia

Wenn du dich einmal für Drupal entschieden hast, unterstützt Acquia dich bei der Umsetzung. Wir bieten dir Acquia Drupal, ein Paket grundlegender Drupal-Module, und den Acquia Dev Desktop, eine einfach zu installierende lokale Drupal-Entwicklungsumgebung, mit denen du schnell loslegen kannst. Acquia bietet seinen Kunden Software, Support, Beratung, Hosting und weitere Dienstleistungen, mit denen deine Drupal-Site schneller auf die Beine kommt und langfristig besser läuft.

© Copyright 2011, Acquia, Inc.

Acquia, Inc.
150 Presidential Way
Suite 310
Woburn, MA 01801
USA

www.acquia.com
sales@acquia.com

+1-978-296-5250