



Open Source CMS

Ein Wegweiser durch
die große Unübersichtlichkeit

LinuxTag, Berlin 2007

Jutta Horstmann

data in transit

I T - B E R A T U N G

- Jutta Horstmann (Dipl. Inform., Dipl. Pol.)
- IT-Beraterin (*Data in Transit*)
- Schwerpunkt: Open Source
- Content Management Systeme
- Web-Programmierung
- Open Source Datenbanken
- OpenUsability.org





Die Themen



Was ist Content Management?



Wozu braucht man das?



Warum Open Source?



Welches Open Source CMS?



Open Source CMS im Web 2.0



1

Die Aufgabe

Was wollen Sie machen?

- schnell *umfangreiche* Web-Auftritte realisieren
- Web Best Practices und Standards anwenden
- Einpflegen von häufig ändernden Inhalten, ohne technische Kompetenz vorauszusetzen
- Aufgabenverteilung an Spezialisten: Administration, Redaktion, Design (Rechtesystem, Workflows)
- dynamische Seiten (Teaser, Listen...)
- CMS or not CMS? Lesen:

<http://www.woodshed.de/content-management-system>



2

Eine Lösung: Content Management Systeme

Inhalte erstellen, aufbewahren, verwalten und finden

- Wiederverwendbarkeit von Inhalten: Darstellung in
 - unterschiedlicher Form
 - an unterschiedlichen Orten
 - in unterschiedlichen Formaten
 - für unterschiedliche Medien
- Versionierung
- zeitgesteuerte Freigabe
- Mehrsprachigkeit
- Metadaten

Getrennte Aufgaben

- Benutzerverwaltung und Rechtesystem
- definierte und getrennte Rollen, Aufgaben und Werkzeuge:
 - Administration
 - Redaktion
 - Autoren
 - Designer

Workflows

- Beispiele:
 - bevor Artikel veröffentlicht wird, muss Chefredakteur gegenlesen
 - bevor neuer Nutzer registriert wird, muss Administrator ihn akzeptieren
 - bevor Aufgabe als erledigt markiert wird, muss Projektleiter zustimmen
 - bevor Blog-Kommentar veröffentlicht wird, muss Blogger ihn freigeben

Anpassen und Erweitern

- Customizing getrennt von sonstigem Code
- Erweiterbarkeit von Content-Typen, Rollen, Workflow
- Add-ons (Plug-Ins / Produkte / Extensions)
- Modularer, übersichtlicher Code

Erweitern

- Issue Tracker
- Projekt Manager
- Mailing Listen Manager
- Newsletter
- Adressverwaltung
- Chat
- Forum
- Blog
- Community
- Wiki
- Dokumenten- Management
- Glossar
- Kleinanzeigen
- Job-Board
- Produktpräsentation
- E-Commerce
- Team-Management
- Groupware

- kein Redaktionsworkflow
- meist Wiki-Markup statt WYSIWYG
- wenig Navigation und Strukturierung des Contents
- wenig Konfiguration
- Niedrigschwelliges Ad-Hoc-Publizieren ohne Qualitätssicherung
- Content kann nur eingeschränkt wiederverwendet, syndiziert, gelistet usw. werden

- Die Aufgabe: Inhalte verwalten
 - Die Lösung: Content Management Systeme
- => Welches System wählen?

Weiter:

- Warum Open Source CMS?
- Welches OS CMS?



Warum Open Source?

- niedrige Kosten
- Man bezahlt Service statt Code, es bleibt Geld für:
 - CMS-Anpassung
 - CMS-Integration
 - Templates und Stylesheets
 - eigene Module entwickeln
 - Tests (Usability, Last...)
 - Nutzer-Schulungen

Anpassung und Erweiterung

- Anpassung bis auf Code-Ebene
- Code-Qualität (modularisiert und strukturiert)
- offener Code bietet mehr als offene API
- Integration mit vorhandenen Systemen oft leichter, z.B. durch Unterstützung von XML und LDAP
- grosses Angebot an Dienstleistern

Community

- Wissensaustausch innerhalb der Community
- Community bietet oft kostenlosen Support und umfangreiches Wissen (Mailing Listen, Foren und Dokumentation)
- schnelles Fehler-Schliessen (entweder durch Community oder man macht es selbst)

Flexibilität, Unabhängigkeit

- “Try before you buy”, man kann
 - alles vor dem Einsatz angucken und ausprobieren
 - die diskutierten Probleme auf den ML anschauen
- Kein “Vendor Lock-In”



Nachteile von Open Source CMS

Nachteile von OS CMS

- Nicht kostenlos – viel Geld fließt in Anpassung und Weiterentwicklung
- u.U. zu viel In-House Anpassung durch die vorhandenen Möglichkeiten
 - frisst zu viel Ressourcen
 - schwierig mit Hauptstrang der CMS-Entwicklung zu integrieren
- Nicht “Enterprise-Level”: Zielgruppe sind eher kleine bis mittlere Installationen, es fehlen Features

Mehr Nachteile

- Geringere kommerzielle Support und Service-Level Garantien
- Wenig Support für aufwändige Migrationen zwischen grossen Versions-Sprüngen
- Geringere Nutzerfreundlichkeit?
- Wenige oder schlechte Dokumentation (??)
- zukünftige Entwicklung, Release-Zyklen, Roadmap?

- Vorteile

- geringe Kosten
- Anpassbarkeit
- Community
- Unabhängigkeit

- Nachteile

- fehlende Features?
- nicht benutzerfreundlich?
- hoher Anpassungsaufwand?

Als nächstes:

- Open Source CMS: Ja bitte, aber: Welches?!



Welches Open Source CMS?

Einsatzzweck

- Einzelperson vs. Unternehmen
- Intranet vs. Aussendarstellung
- Publikation vs. Kollaboration vs. Kommunikation
- Wissensmanagement

Anforderungen sammeln

- kommerzieller Support?
- will man CMS selbst installieren? Hosting?
- schnell Grund-Funktionalitäten, oder stabileres, komplexeres System?
- welche Inhaltstypen und Module?
- gibt es hauptamtliche, geschulte Redakteure?
- braucht man komplexe Strukturierung?
- wie komplex sind die gewünschten Workflows?
- wie wichtig ist ein spezielles Layout?

Systeme evaluieren (1)

- Plattform (Sprache, OS, Datenbank, Webserver)
- Lizenz(en)
- Aktivität im Projekt
- Nutzerbasis und Community
- Dokumentation und Code-Qualität
- Anpassbarkeit (Konfiguration, Code, Templates)
- Accessibility

Systeme evaluieren (2)

- Web-Standards
 - “sprechende” URLs
 - editierbare Meta-Daten
 - valides XHTML
 - valides, externes CSS
- gibt es (kommerziellen) Support
- Security
- Performanz

Features evaluieren

- Standard-Umfang “out of the box”
- Zusatz-Module
 - Im welchem Umfang gibt es Zusatzmodule
 - von welcher Qualität (Sicherheit?!)
 - innerhalb des Projekts oder verstreut
 - Dokumentation?
 - wie einfach ist die Installation
 - wie gut integrieren sie sich?

Standard-Features (1)

- sinnvolle Seitenstruktur und Navigation
- Anlegen und Bearbeiten von Content ohne technische Vorkenntnisse
- Einfache Administrations-Tasks (z.B. Menüpunkt hinzufügen) per Weboberfläche
- Layout komplett konfigurierbar und anpassbar
- Redaktionsworkflows, Arbeitsteilung
- dynamische Seiten: 5 letzte News-Einträge, Events der letzten vier Wochen...

Standard-Features (2)

- Accessibility, Standard-Konformität
- Suchmaschinen-Optimierung
- Umfangreiche Auswahl an Plug-Ins
- Support
- regelmäßige Updates und Weiterentwicklungen

Ambitionierte Features

- Versionierung
- Scheduling
- Staging
- Multi-Sites
- Dokumententmanagement
- eCommerce
- komplexere Workflows
- Event Notifications
- Mehrsprachigkeit in Interfaces und Content
- Druckfreundliche Ausgabe
- Ausgabe nach PDF


High-End Features

- SSL-Unterstützung
- Caching
- Load Balancing
- LDAP Authentifizierung
- umfangreiches Template Management

- Einsatzzweck: Wozu brauche ich ein CMS?
- Anforderungen sammeln
- Systeme evaluieren
- Features evaluieren

Als nächstes:

- Abgleich mit vorhandenen Open Source CMS



Open Source CMS

- Beispiele -

Open Source CMS

- Alfresco
- Bricolage
- Drupal
- ezPublish
- Lenya
- Mambo/Joomla
- OpenCMS
- Plone
- Typo3

Vergleichen und Ausprobieren

- <http://www.opensourcecms.org>
- <http://www.cmsmatrix.org>
- <http://www.dataintransit.com/bitspieces/cms-comparison>

- Hersteller: Alfresco Software
- Plattform: Java
- Lizenz: Mozilla Public License & proprietär
- Anwendungsfall: Kollaboration im Unternehmen, Dokumentenmanagement
- nur eingeschränktes Web Content Management
- geringer Umfang an Modulen
- Templates: FreeMarker
- geringe Community-Integration

Bricolage

- Hersteller: Community
- Plattform: Perl/PostgreSQL
- Lizenz: BSD-artig
- Anwendungsfall: Online News Site / eZine
- Bsp.: Salon, MacWorld
- exportiert statisches HTML
- umfangreiche Workflows
- Templates: Perl / HTML::Mason

- Hersteller: Community
- Plattform: LAMP
- Lizenz: GPL
- Anwendungsfall: Online Community, Kommunikation
- einfache Administration und Anpassung
- Templates: PHP+XHTML
- eingeschränkter Workflow
- starke Community, Support, Dokumentation, Add-Ons

- Hersteller: eZ Systems
- Plattform: LAMP
- Lizenz: proprietär und GPL
- Anwendungsfall: Web-”Broschüre”
- Workflow, Staging
- Closed Source Module
- eigene Template-Sprache
- kein In-Site Editing
- eingeschränkte Nutzerfreundlichkeit

- Hersteller: Wyona, Apache Software Foundation
- Plattform: Java
- Lizenz: Apache
- Anwendungsfall: Neue Zürcher Zeitung (NZZ)
- eingeschränkte Usability
- umfangreiche Workflows
- Templates: Cocoon
- Content wird als XML gespeichert
- “überschaubare” Community und Nutzerbasis

Mambo/Joomla

- Hersteller: Miro (Mambo), Community (Joomla)
- Plattform: LAMP, Lizenz: GPL
- Anwendungsfall: Web-”Broschüre”
- benutzerfreundlich und leicht zu installieren
- eingeschränkte Versionierung, Workflow
- keine komplexe Strukturierung möglich
- Code nicht einfach anpassbar, unübersichtlich
- Templates in PHP
- starke Community, grosse Nutzerbasis

- Hersteller: Alkacon
- Plattform: Java
- Lizenz: LGPL
- Anwendungsfall: Web-”Broschüre”
- eingeschränkter Workflow
- Templates: JSP, Velocity oder XML-basierte Sprache
- Community: eingeschränkt; kommerzieller Support von Alkacon

- Hersteller: Community
- Plattform: Zope (inkl. Webserver und DB), Python
- Lizenz: GPL
- Anwendungsfall: Kollaboration, Wissensmanagement
- sehr umfangreiche und erweiterbare Workflows
- Konfiguration und Erweiterung sehr komplex
- umständliche Administration auf 2 Ebenen: Plone/Zope
- grosse Community, Support und Dokumentation
- Templating: ZPT (Zope Page Templates)

- Hersteller: Community
- Plattform: LAMP
- Lizenz: GPL
- Anwendungsfall: umfangreiche Web-”Broschüre”
- grosse Auswahl an Erweiterungsmodulen
- starke, gut organisierte Community, große Nutzerbasis
- Komplexität
- eingeschränkter Workflow
- Templates: TypoScript

- Für jede Aufgabe das passende Tool! :-)
- Umfangreiches Angebot
- Vor der Auswahl vorbereiten:
 - eigene Anforderungen
 - eigene Ressourcen
 - Laufzeit des Projekts

Zuletzt:

- Ausblick auf neue Anforderungen (CMS 2.0 ?)



Open Source CMS im Web 2.0

Ausblick: Neue Anforderungen

- Web 2.0:
 - Neue Geschäftsmodelle
 - Neue technische Möglichkeiten

=> Neue Anforderungen
- Tagging, Folksonomies
- Remixing, Einbindung externer Dienste und Daten
- komplett anpassbare, nutzerfreundliche Oberflächen
- Einbindung, Verwaltung von Multimedia-Content
- Nutzer- statt Redaktionszentriert

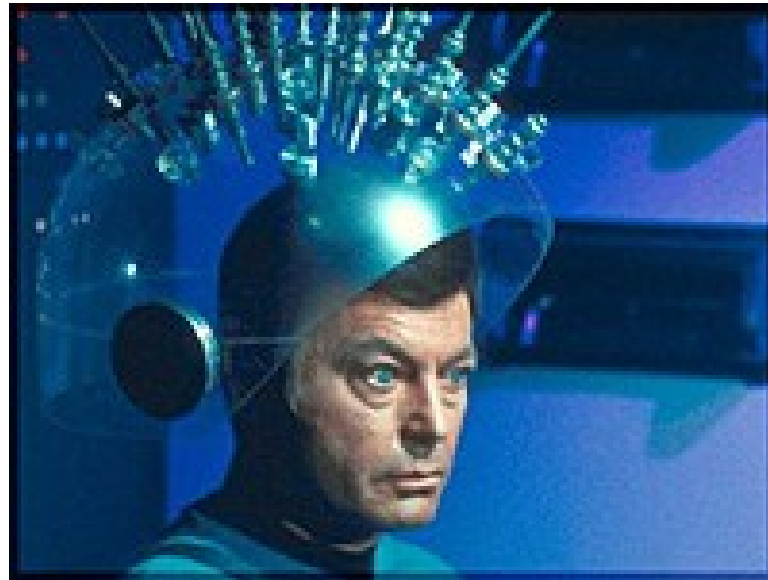
- Suche statt Strukturierung
- Ambitionierte Oberflächen statt Standard-Templates
- Inhalte und Kategorisierung durch Nutzer statt durch Redaktion

=> Das CMS ist tot, es lebe das Wiki?

Oder:

=> CMS + Wiki + Free Tagging + ... ?

Fragen??? Antworten???



data in transit <http://www.dataintransit.com>

Kontakt jh@dataintransit.com

Ressourcen <http://www.dataintransit.com/bitsnpieces/cms-comparison>