

Alexander Wehrum, Peter Eulberg, Claus Mückschel

Paradigmen des Open-Source-Konzeptes am Beispiel der Software Typo3

Immer häufiger entscheiden sich private Nutzer, Unternehmen wie auch öffentliche Einrichtungen für den Einsatz von Open-Source (OS) -Software. Der Artikel fasst die wesentliche Vor und Nachteile von OS-Software zusammen. Darüber hinaus wird exemplarisch ein durch die Autoren durchgeführtes OS-Projekt vorgestellt, bei dem das Content-Management-System Typo3 zum Einsatz kam.

1 Einführung

Open-Source-Software ist zur Zeit ein vielfach diskutiertes Thema in Wissenschaft und Praxis. OS steht im Kontrast zu rein kommerziellen Softwareprodukten, denn im Gegensatz zu konventioneller Software wird OS meist durch räumlich verteilte, heterogene Gruppen entwickelt, in denen Expertenwissen freiwillig und zumeist ohne Erwartung einer direkten Gegenleistung geteilt wird. Das eigentliche Phänomen von OS ist, dass trotz der größtenteils freiwilligen und unentgeltlichen Beiträge einzelner Entwickler und des scheinbar unorganisierten und kollektiven Entwicklungsprozesses, technisch überzeugende und konkurrenzfähige Ergebnisse hervorgebracht werden.

Immer häufiger entscheiden sich deswegen private Nutzer, Unternehmen wie auch öffentliche Einrichtungen für den Einsatz von OS-Software. So ist der Einsatz von OS Server Plattformen mittlerweile auf ein beachtliches Maß gestiegen. Parallel zu dieser rasanten Entwicklung ist auch auf Desktop-Systemen ein verstärkter Einsatz festzustellen. In zahlreichen Marktsegmenten sind OS-Systeme erhältlich, die kommerziellen Lösungen in nichts nachstehen oder diesen sogar überlegen sind. Beispiele hierfür sind Informations- und Content-Management-Lösungen wie der Webserver Apache oder das Programm Typo3. So ist Apache als Webserver-Software mittlerweile zum Marktführer avanciert, mit einem Marktanteil von 68 % gefolgt von Microsofts IIS mit 21% (Netcraft 2004).

Die Gründe für diese beachtlichen Erfolge von OS-Produkten sind vielfältig.

Die am häufigsten aufgeführten Argumente, die für ihren Einsatz sprechen,

sind die wegfallenden Lizenzkosten, erhöhte IT-Sicherheit, Offenheit und Erweiterbarkeit der Systeme sowie Unabhängigkeit von monopolistischen Softwareanbietern.

Trotz teilweise erheblicher Vorteile beim Einsatz von OS-Produkten gibt es auch einige Problemfelder und Risiken, die es bei einer Entscheidungsfindung bezüglich der Softwareauswahl zu beachten gilt.

Im vorliegenden Beitrag wird zunächst geklärt, welches die wesentlichen Charakteristika einer OS-Software sind. Anschließend werden die Chancen und Risiken beim Gebrauch von OS-Produkten zusammengefasst, die bei einer Entscheidungsfindung behilflich sein sollen. Im letzten Teil wird exemplarisch ein durch die Autoren durchgeführtes OS-

Projekt vorgestellt, bei dem das Content-Management-System Typo3 zum Einsatz kam.

2 Hintergrund und Begriffsklärung

Wenngleich Open-Source-Software schon in den späten 60er Jahren existierte, entstand der eigentliche Begriff erst um 1998 mit Gründung der Open-Source Initiation (OSI). Die OSI überprüft die Lizenz einer Software hinsichtlich 10 festgelegter Kriterien und erkennt an Hand dieser Kriterien die Lizenz und damit das ihr zu Grunde liegende Programm gegebenenfalls als Open-Source-Software an.

Zu den Kriterien der OSI gehören u.s. die folgenden:

1) Freie Weitergabe: Niemand darf in seinem Recht hinsichtlich der Weitergabe (Schenkung oder Verkauf) eingeschränkt werden. In der Lizenz darf folglich keine Lizenz- oder Verkaufsgebühr festgeschrieben sein.

2) Offenheit des Quellcodes: Der Quellcode muss für jeden erhältlich sein (z.B. per Download aus dem Internet). Dabei muss der Quellcode in verständlicher Form abgefasst sein. Ein absichtlich unverständlich geschriebener Quellcode ist nicht zulässig.

3) Abgeleitete Software: Die Lizenz muss Derivate und Veränderungen zulassen, wobei die daraus resultierenden Programme wiederum unter den selben Bedingungen wie die Ausgangssoftware weitergegeben werden können müssen.

4) Keine Einschränkung bezüglich des Einsatzes: Die Lizenz darf das Einsatzgebiet der Software in keinem Fall einschränken.

5) Weitergabe der Lizenz: Die Rechte am zugrunde liegenden Programm müssen auf alle Personen übergehen, die es erhalten. Eine Pflicht zum Erwerb einer zusätzlichen Lizenz darf nicht bestehen.

Auf die genaue Beschreibung der fünf übrigen Kriterien wird an dieser Stelle verzichtet, sie beinhalten a) Das Verbot der Diskriminierung von Personen oder Gruppen, b) das Verbot der Lizenzbeschränkungen auf ein bestimmtes Produktpaket, c) das Verbot der Lizenzbeschränkung für die Weitergabe der Software mit anderer Software und d) das Gebot der Technologieneutralität.

Unterschiedliche Formen von OS-Software

Zunächst lässt sich Open-Source-Software nach ihrer Lizenzform unterscheiden.

Im Oktober 2004 waren insgesamt 54 unterschiedliche Lizenz-Formate von der OSI anerkannt und zertifiziert. Das wichtigste Format ist die GNU General Public License (GPL) mit ca. 71 % Verwendungshäufigkeit, dann folgt die Lesser General Public License (LGPL) mit 11% und als dritthäufigste die Berkely Software Distribution (BSD) mit 7 % (vgl. OSI 2004). Wichtigster Unterschied zwischen GPL und LGPL ist, dass letztere erlaubt, die zugrunde liegende Software als Baustein für eine proprietäre Software zu benutzen, d. h. bei Integration eines LGPL-lizensierten Bausteins muss - im Gegensatz zur GPD - nur der Quellcode des Bausteins offen gelegt werden und nicht der Quellcode des kompletten, neu entstandenen Gesamtwerks.

Das wichtigste Lizenzformat GPL geht auf Richard Stallman zurück, der 1989 die Version 1.0 zusammen mit dem Rechtsprofessor Eben Moglen veröffentlichte. Ziel war dabei vor allem, eine eventuell mögliche Proprietarisierung einer offengelegten und abgeleiteten Software zu verhindern (FSF 2004).

Open-Source-Software lässt sich auch nach ihrer Nutzung und ihrem Vertrieb durch kommerzielle Firmen unterscheiden. Zwar sind die Wurzeln der OSS klar von nicht kommerziellem Gedankengut geprägt, jedoch gibt es eine Reihe von Unternehmen wie Novell oder IBM, die sich die Vorteile von OSS aus rein kommerziellen Interessen zunutze machen. Beispiele hierfür sind die Linux-Distributionen RedHat und Suse, letztere wurde im Jahre 2003 durch den US-Konzern Novell aufgekauft, was ein eindeutiges Indiz ist für das gesteigerte Interesse etablierter Unternehmen an den Marktchancen von OS-Software.

Des weiteren lässt sich OS-Software nach ihrer Organisationsform unterscheiden. Raymond (1999) verglich die Entwicklung von OS-Software mit einem Basar, während klassische Softwareentwicklung dem Bau einer Kathedrale entspräche. Prinzipiell kann die Entwicklung von OS-Software jedoch in beiden Formen, d.h. offen oder geschlossen stattfinden. Allerdings überwiegt die offene Organisation, in der eine große, meist weltweit verstreute Anzahl von Personen Zugriff auf Quellcode, Fehlerdatenbank oder Diskussionsforen und Mailinglisten des Projektes hat und somit an der Weiterentwicklung und an Entscheidungen teilhaben kann. Ein Beispiel für eine eher geschlossene, klassische Form, geprägt durch einen stark organisierten Stil der Entwicklung, ist die Open-Source-Software MySQL (Krishnamurthy 2002).

3 Vor- und Nachteile von OS-Software

Im Folgenden sollen kurz die Vor- und Nachteile beim Einsatz von OS-Software zusammengefasst und anschließend eine Übersicht derjenigen Faktoren gegeben werden, die bei der Auswahlentscheidung bezüglich des Einsatzes von OS-Software beachtet werden sollten. Die Empfehlungen beruhen auf den Erfahrungen, die die Autoren als Mitarbeiter der Connecta AG bei der Umsetzung zahlreicher OS-Projekte gemacht haben.

3.1 Vorteile von OS-Software

Kostenvorteil

Das wichtigste Argument für den Einsatz von OSS sind zweifelsfrei die augenscheinlich niedrigen Kosten. Allerdings sind für eine ökonomisch richtige Entscheidung nicht nur die Anschaffungskosten, sondern die Gesamtkosten zu beachten. Die Gesamtkosten werden üblicherweise nach dem Total Cost of Ownership (TCO) Ansatz der Gartner Group berechnet. Dies beinhaltet alle direkten und indirekten Kosten, die bei der Implementierung und Nutzung über einen bestimmten Zeitraum hin anfallen. Zu den direkten Kosten zählen in erster Linie die Hard- und Softwarekosten, aber auch Kosten für technischen Support, Verwaltungskosten oder für die Schulung der Mitarbeiter. Indirekte Kosten sind zum Beispiel mögliche Ausfallzeiten der Systeme oder die Kosten für das Erlernen der Software im Arbeitsalltag.

Eine Reihe von Studien belegt, dass Open-Source-Software meist nicht nur bei Betrachtung der Anschaffungskosten vorteilhafter als proprietäre Software ist, sondern auch unter Einbeziehung der Gesamtkosten. Die im deutschsprachigen Raum bekannteste Studie ist die von Unilog für die Landeshauptstadt München durchgeführte Untersuchung, bei der Windows XP / Microsoft Office mit Linux und anderen OS-Komponenten verglichen wurde. Laut dem Ergebnis kann die Stadt München beim Einsatz von OS-Software 12 Millionen Euro einsparen (Unilog 2003).

Beständigkeit und Updates

Das nicht zu unterschätzende Lieferantenrisiko bei der Anschaffung proprietärer Software fällt bei OSS weniger ins Gewicht, denn bei OSS begibt man sich nicht in ein Abhängigkeitsverhältnis zu einem Drittanbieter, und ist somit nicht dessen produkt- und marktstrategischen Entscheidungen ausgeliefert. So stellt selbst Microsoft die technische Unterstützung für ältere Windows- oder andere Software-Versionen regelmäßig ein und zwingt die Endnutzer zur Anschaffung neuerer Versionen, die dann unter Umständen tiefgreifende Veränderungen der Soft- und Hardwarearchitektur nach sich ziehen. Etablierte und weit verbreitete Open-Source-Software ist nach heutigem Erfahrungsstand um einiges beständiger. Zudem sind durch die größere Anzahl von Entwicklern die Zyklen für Softwareupdates für neue Funktionen oder Fehlerbehebungen in der Regel wesentlich kürzer als bei proprietärer Software.

Offenheit und Unabhängigkeit

Die Schwierigkeiten und Herausforderungen einer informationstechnischen Infrastruktur liegen weniger in der Phase des Aufbaus als im dauerhaften Betrieb (Czegka et al. 2003). Aus diesem Grund sind bei der Konzeption offene Informationssysteme vorzuziehen, welche sich möglichst ohne großen Aufwand anpassen, erweitern, integrieren oder auch ersetzen lassen.

Hier haben Open-Source basierte Systeme einen entscheidenden Vorteil gegenüber den meisten proprietären Produkten, da zum einen durch den offenen Quellcode und die (meist) gute Dokumentation jeder sich in die Funktionsweise des Programms einarbeiten kann und zum anderen OS-Software in der Regel so programmiert ist, dass sie offen für Erweiterungen ist. Erst durch diese beiden Charakteristika

wird es möglich, dass eine Vielzahl unterschiedlicher, unabhängiger Entwickler das Programm weiterentwickeln können.

Durch die Offenheit von OS-Produkten, die sowohl den Quellcodes, als auch die allgemeine Softwarearchitektur umfasst, können Programme leicht um fehlende Funktionen durch eigene Programmierung ergänzt und somit auch veränderten Anforderungen gerecht werden. Ebenso können völlig neue, gewinnbringende Funktionalitäten problemlos integriert werden. Damit ist man bei OS nicht mehr an die Vorgaben bestimmter Unternehmen und deren Software-Features gebunden. Ganz anders bei kommerziellen Produkten, wo der Quellcode zurückbehalten wird oder die Softwarearchitektur Erweiterungen - wenn überhaupt - nur mit größerem Aufwand zulässt.

Sicherheitsargument

Ein häufiger Einwand gegen OS-Software besagt, dass bei offenem Quellcode das Sicherheitsrisiko wachse, weil damit Schwächen der Software bekannt sind. Zumindest nach heutigem Erfahrungsstand hat sich diese Befürchtung nicht materialisiert. Ganz im Gegenteil, durch die Vielzahl von Entwicklern, die an der Software arbeiten und den Quellcode auf Sicherheitslücken prüfen, werden vorhandene Fehler schneller entdeckt und bestehende Sicherheitslücken alsbald geschlossen. Dies belegte unter anderem im Jahre 2003 eine Studie des MITRE und des US-Verteidigungsministeriums (Mitre 2002).

Kompetente Unterstützung durch die OS-Community

Bei Verwendung von OS-Software steht ein umfangreiches, weltweites Angebot an Unterstützung durch andere Anwender und Entwickler zur Verfügung. Die meisten großen OS-Projekte verfügen über Fehlerdatenbanken, FAQ-Seiten, Mailinglisten und Foren, in denen Neuerungen und Probleme angesprochen und diskutiert werden. Wenn ein Entwickler eine konkrete Frage hat, zum Beispiel wenn er durch die Implementierung eines vorhandenen oder eines eigens entwickelten Moduls eine OS-Software an seine spezifischen Anforderungen anpassen möchte, so kann er sein Problem in einem entsprechenden Forum publizieren. Innerhalb kürzester Zeit ist dann mit der Unterstützung seitens anderer Entwickler zu rechnen. Dieser Umstand verkürzt die notwendige Entwicklungszeit erheblich.

3.2 Nachteile von OS-Software

Gewährleistung und Wartung

Gegenüber konventionellen Produkten ist OS-Software insofern im Nachteil, dass hinsichtlich Wartung und Gewährleistung rechtliche Ansprüche meist völlig ausgeschlossen sind. Dies scheint auf den ersten Blick ein schwerwiegender Nachteil, bei genauerer Betrachtung allerdings bieten sich dem Anwender von OS-Software jedoch durchaus vielfältige Möglichkeiten zur Fehlerkorrektur und zur Anpassung der Software an die eigenen spezifischen Ansprüche, auch wenn hierauf keinerlei rechtlicher Anspruch gegenüber irgendeiner Seite besteht. Wie im letzten Abschnitt beschrieben, sind Fehlerkorrekturen und Anpassungswünsche meist sehr schnell und qualitativ hochwertig über öffentliche Diskussionsforen zu erreichen, weil eine große Anzahl von Anwendern und Entwicklern gemeinsam an der Lösung der Softwareprobleme arbeiten können. Das

Risiko fehlender Gewährleistung fällt daher faktisch nur dort ins Gewicht, wo die Entwicklergemeinde eher klein ist.

Copyright- und Patentrecht-Risiko

Weitere Risiken beim Einsatz von OS-Software ergeben sich im Hinblick auf Rechte Dritter an der Software oder an Teilen hiervon. Ist der Quellcode des OS-Produktes ganz oder partiell zugunsten Dritter geschützt, weil die ihre Softwareentwicklungen nicht zu OS-Software erklärt haben, kann der Einsatz ohne entsprechende Lizenz als Rechtsverletzung eingestuft werden.

Die Tatsache, dass oft unbegrenzt viele Entwickler an der Weiterentwicklung eines Programms beteiligt sind, macht es praktisch unmöglich, jeden Teil des Codes auf eventuelle Urheberrechtsverletzungen hin zu überprüfen. Theoretisch könnte dann der Lizenznehmer einer OS-Software von einem geschädigten Eigentümer der Urheberrechte in Regress genommen werden (Erbert & Ruffin 2004) oder zumindest gezwungen werden, die Software nicht weiter unverändert zu verwenden. Gegebenenfalls muss der beanstandete Code abgeändert oder ersetzt werden, was mit einem erhöhten Aufwand verbunden wäre.

Der Rechtsschutz für Software folgt gegenwärtig aus dem Urheberrecht. Seit einiger Zeit gibt es Bestrebungen dahingehend, dass in Zukunft Software auch patentrechtlich geschützt werden kann, was das OS-Konzept insgesamt gefährden könnte (Blind et al. 2003).

In Japan und den USA ist die Patentierbarkeit von Software bereits Realität. In Europa wird die Ausdehnung des Patentrechts auf Software gegenwärtig noch heftig diskutiert, denn die Einführung der Patentfähigkeit von Software würde kommerziellen Anbietern ein breites juristisches Feld eröffnen, um Klagen gegen OS-Software zu führen, wenn diese OS-Software Bestandteile enthielte, die vom klagenden Anbieter vorher patentiert wurde. Da der Quellcode von OS immer offen verfügbar ist, wäre der Nachweis der Patentverletzung besonders leicht möglich. Andererseits wäre jeder OS-Entwickler ständig in Gefahr, unbeabsichtigt Patentverletzungen zu begehen, da es kaum möglich ist, eine umfassende Patentrecherche durchzuführen (Brügge 2004). Falls es tatsächlich zur gesetzlichen Neuregelung des Patentrechts innerhalb der EU in der Form kommen sollte, dass generell Patente auf Software erteilt werden könnten, würde sich das Risiko für OS-Entwickler und -Anwender, gegen Rechte Dritter zu verstoßen, signifikant erhöhen (Pfuhl 2004). Denn anders als beim Urheberrecht würde das Softwarepatent Ideen und Grundsätze schützen, nicht nur den konkreten Code.

Ein gutes Beispiel für die gegenwärtige Rechtsunsicherheit stellt ein Projekt der Stadt München dar. Das Linux-Migrationsprojekt der Stadt München wurde vorübergehend gestoppt, weil die Stadt patentrechtliche Probleme beim Einsatz von Linux befürchtete. Ein Rechtsgutachten hat vorerst ergeben, dass München wohl keinerlei Risiken zu befürchten hat (Heise-Online 2004).

Begrenztes Angebot

Für viele Anwendungsfelder ist noch gar keine oder nur sehr unausgereifte OS-Software vorhanden. Dies gilt zum Beispiel für die Bereiche Customer Relationship Management, Supply Chain Management, Enterprise Relationship

Management oder andere spezifische Branchensoftware (Gläser 2004). Dieses lückenhafte Angebot kann ein erheblicher Nachteil für die betroffenen Anwendungs-Gebiete darstellen.

4 Tipps zu Auswahl und Problemlösung

Umfassende Kostenanalyse

Grundsätzlich sollte man bei der Entscheidung zwischen proprietärer und OS-Software die gesamten zu erwartenden Kosten (TCO) beachten; es wäre kurzsichtig, sich primär am Kostenvorteil der Lizenzierung von OS-Produkten zu orientieren. Zwar sind die Gratis-Lizenzen ein nicht zu vernachlässigender Faktor, machen aber meist oft nur einen Teil der TCO aus und sollten daher in diesem Zusammenhang gesehen werden. Wie schon ausgeführt, geben aber OS-Produkte mit weiteren geldwert-relevanten Ersparnissen einher, so dass sie auch bei Einbeziehung aller Kosten meist der proprietären Lösung vorzuziehen sind.

Ingesamt sollten OS-Produkte nicht nur bei den Lizenzkosten, sondern auch bei den TCO sowie insgesamt klare Vorteile gegenüber proprietärer Software bieten.

Zur Software-Auswahl

Liegen ein oder mehrere potentiell nutzbare OS-Produkte vor, sollten zunächst zwei Fragen besonders geprüft werden: (1) wie sieht die Community aus und (2) wie ist die Qualität des zugehörigen Produktes? Die hinter der Entwicklung der Software stehende Community und ihre Anwender sollten möglichst nicht zu klein oder inaktiv sein. Dabei interessiert, wie oft neue Releases veröffentlicht werden und wie viele Erweiterungen bereits bestehen. Man sollte auch überprüfen, wie lange es dauert, bis ein User Hilfe für ein gestelltes Problem erhält. Hinsichtlich der Qualität sollte man sich die Zeit nehmen und zumindest einen Teil der Software genauer testen. Nur durch eine große und aktive Community und eine hohe Qualität kann OS-Software ihre Vorteile voll entfalten.

„Release early, release often“

Hat man sich für ein OS-Produkt entschieden und plant, dieses weiterzuentwickeln, um es an die eigenen Bedürfnisse anzupassen, dann sollten die neuen Beiträge zum Programm möglichst früh und ausführlich dokumentiert und an geeigneter Stelle veröffentlicht werden. So können frühzeitig gleichgesinnte Entwickler gefunden werden, die Interesse an dieser oder ähnlichen Erweiterungen haben. Solche frühen „Releases“ sind meist weniger stabil und relativ unvollkommen; dieser Nachteil wird durch den Synergieeffekt einer parallelen und gebündelten Suche und Beseitigung von Fehlern und Mängeln in der Regel mehr als ausgeglichen (Brügge et al. 2004).

Auch wenn es nicht um die aktive Weiterentwicklung geht, weil man nur plant, auf bestehende Lösungen aufzubauen, ist es ratsam, bei Schwierigkeiten frühzeitig auf das vorhandene Wissen der Community zurückzugreifen. Beispielsweise bei Problemen mit der Integration eines weiteren Moduls in die OS-Software sollte eher früher als später in den Datenbanken der Community nach Lösungsmöglichkeiten gesucht werden. Finden sich dort keine Antworten auf das Problem, so sollte es an geeigneter Stelle publiziert werden; wenn die Commu-

nity groß und aktiv ist, erhält man meist bereits nach wenigen Stunden Antwort von einem versierten Entwickler.

Vorzüge gängiger Lizenzen

Auch wenn die langjährige Erfahrung beim Einsatz von OS-Software bisher keine nennenswerten Probleme ergab, sollten einige juristische Kriterien bei der Auswahl von OS-Produkten beachtet werden. So sind stets OS-Produkte vorzuziehen, die auf einem weit verbreiteten und gängigen OS-Lizenz-Format beruhen und einen weitestgehend klaren rechtlichen Status bezüglich der urheberrechtlichen Fragen und Nutzungsmöglichkeiten haben (Ebert & Ruffin 2004). Wird die gewählte OS-Software über einen Distributor erworben, ist zu prüfen, ob dessen Vertrag mit den eigenen Vorstellungen vereinbar ist. Zwar sind die rechtlichen Risiken aufgrund der teilweise ungeklärten Rechtslage schwer kalkulierbar, dennoch sind diese Vorsichtsmaßnahmen empfehlenswert.

5 Das OS-Projekt „Welterbe Oberes Mittelrheintal“

Im Folgenden werden die Erfahrungen exemplarisch an Hand der Website „Unesco Welterbe Oberes Mittelrheintal“, eines typischen OSS-Projekts, beschrieben.

Im November 2003 hat die Staatskanzlei des Landes Rheinland-Pfalz eine Ausschreibung zur Realisierung einer Website für das „Unesco Welterbe Oberes Mittelrheintal“ veröffentlicht. Neben einer Reihe von proprietären Content Management Systemen bot sich auch das OS-Produkt Typo3 für dieses Projekt an, aus verschiedenen Gründen fiel die Entscheidung zugunsten von Typo3 aus.

Anforderungen an das Projekt

Ziel des Projektes war, einem breiten Nutzerspektrum eine zentrale Informationsplattform zum Thema „Unesco Welterbe Oberes Mittelrheintal“ zur Verfügung zu stellen.

Die zu schaffende Plattform sollte durch unterschiedliche Informationsangebote qualitativer und quantitativer Art (z.B. Veranstaltungskalender, Geschichte des Tals, Ausflugsziele, Adressen und Links, Angebote zu Führungen, etc.) einen heterogenen Interessenskreis ansprechen, der eine Vielzahl verschiedener Zielgruppen und Nutzerkreise umfassen sollte.

Eine attraktive, nutzerfreundliche Gestaltung und Visualisierung der Inhalte sollte das Portal zur ersten Webadresse rund um das Welterbegebiet machen. Durch den Einsatz modernster Darstellungsformen, u. a. multimediale Techniken zur Aufarbeitung der zu präsentierenden Inhalte wie 3D-Bilder und Webcam-Einbindungen, sollten die Informationen mediengerecht dargeboten werden.

Um die Nutzung und Pflege des Internetportals möglichst effizient zu gestalten und die Informationen stets aktuell zu halten, war es aufgrund der großen Fülle an Inhalten und Diensten zwingend erforderlich, die Website unter Verwendung eines Content-Management-Systems (CMS) dynamisch zu erzeugen und einen hohen Grad an Automatisierung einzubauen.

Unter anderem wurde angestrebt, dass die Redakteure speziell zugewiesene Bereiche selbständig und komfortabel bearbeiten und erweitern können.

Die Website sollte ferner wegen ihrer internationalen Relevanz in 6 Sprachen inkl. Japanisch verfügbar sein. Eine andere zentrale Anforderung war die Barrierefreiheit der

Website nach § 11 Abs. 1 Satz 2 des Behindertengleichstellungsgesetzes, d.h. der Besucher sollte u. a. die Möglichkeit erhalten, alle Inhalte - ausgenommen Bilder und graphische Elemente - uneingeschränkt in einer reinen Textversion durchsuchen zu können. Es war sicher zu stellen, dass die Website einwandfrei und vollständig genutzt werden kann, auch von Behinderten oder von Benutzern mit anderen Einschränkungen (zum Beispiel langsame Internetverbindung).

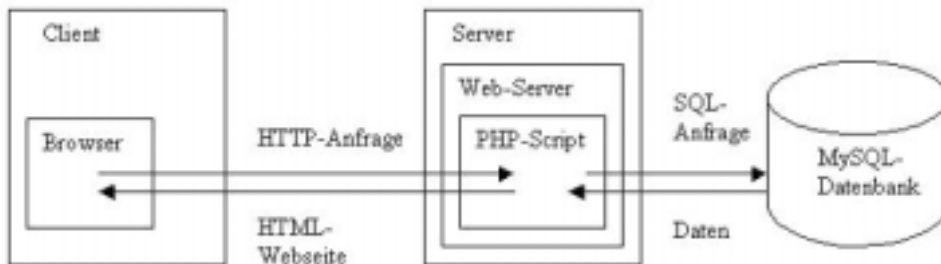


Abbildung 1: Schematische Darstellung der Generierung dynamischer Web-Seiten mit PHP und MySQL.

Zum Content Management System Typo3

Der Realisierung des Unesco-Projekts lag das CMS Typo3 zu Grunde. Typo3 ist ein lizenzkostenfreies Open-Source Programm, welches den rechtlichen Bestimmungen der GPL (Gnu Public License) unterliegt. Mit Hilfe des Programms können Webprojekte dynamisch erstellt und verwaltet werden und zwar in verschiedensten Formen, zum Beispiel als öffentlich zugängliche Websites oder als Intranet-Systeme.

Bei Typo3 handelt es sich um eine plattformunabhängige Serverapplikation, zu deren Benutzung ein gängiger Webbrowser ausreicht. Typo3 ist in der Programmiersprache PHP entwickelt und verwendet konsequenterweise MySQL als Datenbank. Sowohl PHP als auch MySQL sind Open-Source Produkte. PHP zeichnet sich vor allen Dingen durch seine ausgezeichneten Datenbank- und Internet-Protokolleinbindungen aus (Schepp & Zierl 2001), die Kombination mit MySQL hat sich in vielen Internet-Projekten als bewährtes und wirkungsvolles Werkzeug zur dynamischen Generierung von Websites erwiesen (Abb. 1).

PHP ist zudem plattformübergreifend und läuft auf allen gängigen Unix-Versionen einschließlich Linux und Macintosh und auf den verschiedenen Windows-Versionen (Windows 95/98/ME/NT/2000/XP), vgl. Raymans 2001, Schepp & Zierl 2001, Zierl 2001.

Das System Typo3 beruht außerdem auf der hauseigenen Konfigurationssprache TypoScript, die der Anpassung der Software an spezifische Bedürfnisse dient. In seiner Grundversion verfügt das Programm bereits über Funktionalitäten, die normalerweise erst bei kostenpflichtigen Systemen der Enterprise-Klasse üblich sind. So ist eine umfangreiche Rechteverwaltung bereits inbegriffen, die es berechtigten Redakteuren ermöglicht, Beiträge an den ihnen zugewiesenen Stellen komfortabel in das System einzupflegen (Abb. 2 und 3). Es ist des Weiteren möglich, Beiträge zeitgesteuert oder im Vieraugenprinzip nach Freischaltung durch einen übergeschalteten Redakteur zu veröffentlichen.

Weiterhin verfügt das Programm über einen integrierten Erweiterungsmanager, mit dessen Hilfe schnell und prob-

lemlos zusätzliche, durch die große und aktive Typo3-Community entwickelte Module in das System eingebunden werden können (Abb. 4).

Diese und viele zusätzliche Vorzüge von Typo3 haben die Nutzer- und Entwicklergemeinde rasant anwachsen lassen, sodass das Programm ständig erweitert wird und zur Zeit in der Version 3.7.0 vorliegt. Seit den Anfängen von Typo3 im Jahre 2001 hat es eine beachtliche Verbreitung gefunden; und ist mittlerweile mit über 36.000 Installationen im Einsatz

(Typo3.org 2004). Die technischen Voraussetzungen für den Einsatz von Typo3 sind zur Zeit: Apache 1.3.3.1 oder höher, MySQL 4.0.18 oder höher, ImageMagick 4.2.9 oder höher (empfohlen), Freetype 1.3.1 (empfohlen), GDLib 1.8.3 (empfohlen) und PHP 4.3.5 oder höher.

Besonders empfehlenswert sind Systemerweiterungen wie „ImageMagick“ und „GDLib“.

Mit ImageMagick ist es möglich, Bildformate zu konvertieren und bestehende Bilddaten zu bearbeiten. ImageMagick bietet dem Redakteur der Website damit die Möglichkeit, Bilder, die sich bereits auf dem Webserver befinden, zu bearbeiten, z.B. zu skalieren oder zu schärfen. Der Redakteur ist relativ frei in der Auswahl des Formates des Bildes, denn das System übernimmt eigenständig die Umwandlung des Bildes in ein webgerechtes Format. Bei GDLib handelt es sich um eine Grafikbibliothek, mit deren Hilfe unterschiedliche Grafiken kombiniert werden können. In Kombination mit FreeType ist es Typo3 möglich, in einem zuvor definierten Layout Menüpunkte mit grafischen Elementen zu generieren und abzubilden. So kann ein Redakteur das grafische Menü einer Website erweitern, ohne über spezielle Kenntnisse der Grafikbearbeitung oder der Programmierung zu verfügen.

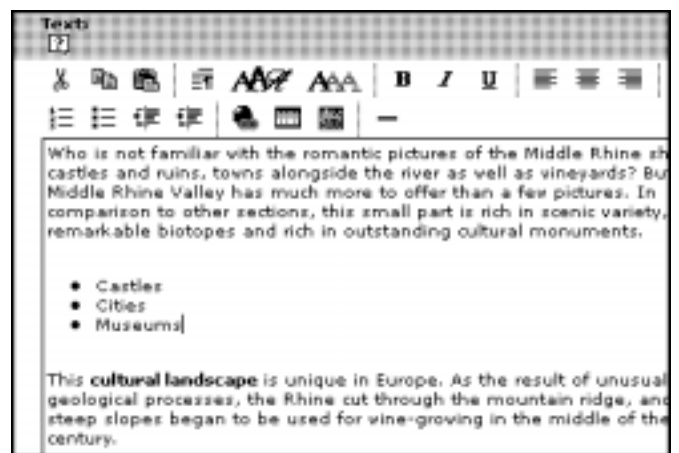


Abbildung 2: Im Rich Text Editor von Typo3 können Texte auf einfache Art und Weise eingegeben und formatiert werden.

Erfahrungen bei der Umsetzung

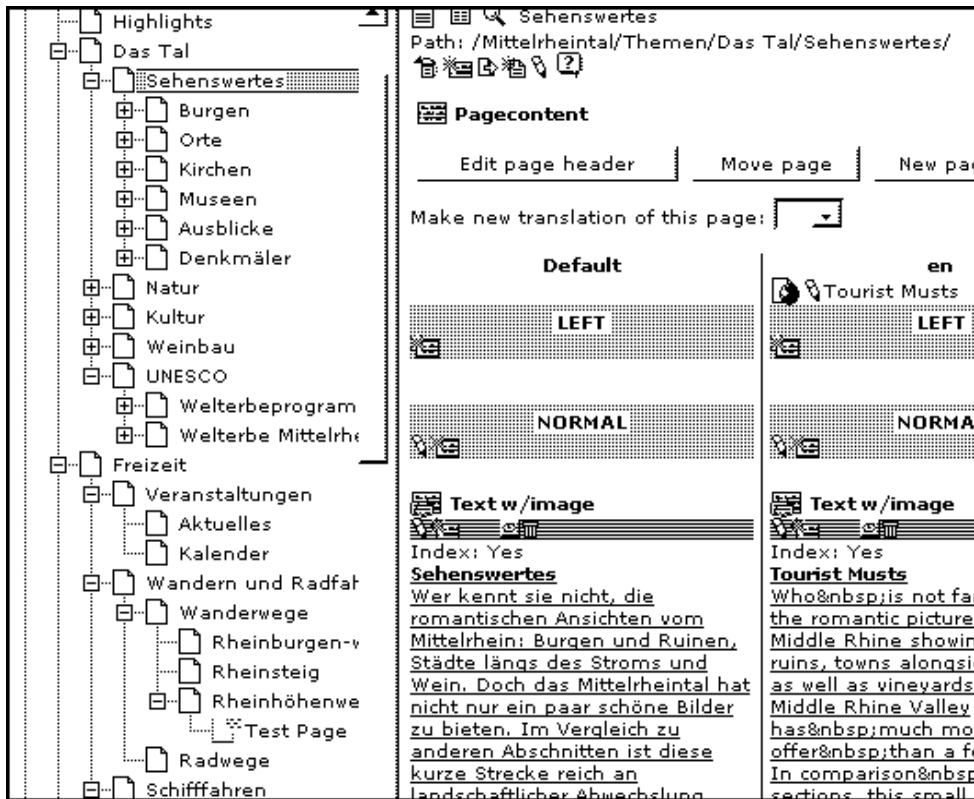


Abbildung 3: Die Struktur von Daten und Seiten wird übersichtlich als Baumdiagramm dargestellt und kann beliebig ein- und ausgeklappt werden.

Zunächst war es notwendig, den bestehenden Webserver auf die oben genannten technischen Voraussetzungen umzustellen. Im vorliegenden Fall, der nicht untypisch ist, stand ein Cobalt Raq4 Server zur Verfügung. Bei solchen älteren Modellen ist das Update der zu Testzwecken eingerichteten Installation nicht immer einfach. Besser und oft auch kostengünstiger ist es daher, sich gleich für den Einsatz eines neuen Root Servers mit vollen Zugriffsrechten zu entscheiden, bei dem die benötigten Pakete problemlos installiert werden können. Wegen der umfangreichen Funktionen von Typo3 ist

es zudem empfehlenswert, dass der Server über mindestens 1,5 GHz und 512 Mbyte RAM verfügt.

Es existieren mittlerweile zahlreiche kompetente Anbieter, die sich auf das Hosting von Typo3-Projekten spezialisiert haben. Diese stellen nicht nur einen vorkonfigurierten Webserver mit der erforderlichen Performance zur Verfügung, sondern installieren in der Regel auch schon Typo3 vorab, sodass sich weniger versierte Anwender die Schwierigkeiten der Grundkonfiguration ersparen und sich direkt mit dem eigentlichen Programm und dessen Konfiguration beschäftigen können.

Mit der installierten Basisversion des CMS Typo3, konnte dann der überwiegende Teil der im Unesco-Projekt benötigten Anforderungen bereits ohne Verwendung von Zusatzmodulen realisiert werden. So sind z.B. eine umfangreiche Rechte-

verwaltung für verschiedene Redakteure und die Möglichkeit der Mehrsprachigkeit der Website bereits von Hause aus vorhanden.

Für die Abbildung der geforderten barrierefreien Version war es nötig, auf ein von der Entwicklergemeinde bereits programmiertes Zusatzmodul zurückzugreifen. Mit einigen wenigen Anpassungen in gut dokumentierten Quellcodes der Erweiterung war das Zusatzmodul relativ schnell implementiert und funktioniert seither einwandfrei.



Abbildung 4: Per Erweiterungsmanager können komfortabel eigene sowie fremde Erweiterungen für Typo3 installiert werden.

weltweites Entwickler-Netzwerk effizient und qualitativ hochwertig realisiert werden konnte.

6 Abschließende Bewertung aus Entwickler- und Benutzersicht

Typo3 erfüllt die bereits angeführten OS-Kriterien in jeder Hinsicht. In erster Linie sind hier die günstigen Gesamtkosten des Programms anzuführen. Mit Typo3 vergleichbare, kommerzielle und kostenpflichtige Software, die die vom Auftraggeber geforderten Merkmale abdeckt, hätte nach einer internen Studie wesentlich höhere Gesamtkosten verursacht. Da der Quellcode von Typo3 verfügbar und gut dokumentiert ist und darüber hinaus das Programm eine für OS übliche, offene Softwarearchitektur aufweist, war es ohne größeren Aufwand möglich, das Programm modular zu erweitern und auf die Bedürfnisse des Auftraggebers anzupassen. Darüber hinaus ist der Auftraggeber in Zukunft nicht von einem einzelnen Softwareanbieter abhängig, da sich jeder versierte Entwickler in die Funktionsweise von Typo3 einarbeiten kann und gegebenenfalls kostenfreien Support der Entwicklergemeinschaft erhält.

Einzig die für die Grundinstallation von Typo3 notwendige Konfiguration des Webservers und die wahrscheinlich anfangs kompliziert erscheinende Konfigurationssprache TypoScript kann den Einstieg in diese OS-Software erschweren. Sind diese Hürden einmal erfolgreich genommen, eröffnet sich sowohl für den ambitionierten Entwickler als auch für die Benutzer ein leistungsfähiges und auf die eigenen Bedürfnisse hin beliebig erweiterbares Programm, welches nicht zuletzt wegen der großen und aktiven Community alle Vorteile einer Open-Source-Software aufweist.

Dank:

Herrn Dr. G.A. Schachtel (Universität Giessen) danken wir für konstruktive Kritik und ergänzende Hinweise.

7 Literatur

- BLIND, K., EDER, J. NACK, R. & STRAUSS, J. (2003): Software-Patente – Eine empirische Analyse aus ökonomischer und juristischer Perspektive. Physica-Springer, Heidelberg.
- BRÜGGE, B., HARHOFF, D., PICOT, A., CREIGHTON, O., FIEDLER, M. & HENKEL, J. (2004): Open-Source-Software - Eine ökonomische und technische Analyse. Springer, Berlin.
- CZEGKA, W., BRAUNE, S., PALM, H., RITSCHER, B., KLUMP, J. & LOCHTER, F. A. (2003): Beispiele ISO 19115 DIS konformer Metadaten in Katalogservices. Zwei Anwendungen aus dem Bereich umwelt- und geowissenschaftlicher Geofachdaten im Rahmen der Metadatencommunity der „GIB“. Beitrag zur U2 Konferenz, Zentrum für Geoinformatik Salzburg
www.unigis.ac.at/club/us/2003/UP_Beitrag_Czegka_Braune.pdf Zugriff am 15.10.2004
- EBERT, C. & RUFFIN, M. (2004): Produkte Entwickeln mit Open-Source-Software – Risiken und Erfahrungen. In: Sauerburger, MITRE (2003): Use of FREE and Open-Source-Software (FOSS) in the US Department of Defence. www.microcross.com/dodfoss.pdf Zugriff: 5.10.2004
- Netcraft 2004 „September 2004 web server Study“ news.netcraft.com/archives/web_server_survey.html Zugriff: 02.11.2004
- OSI (2004): www.opensource.org/licenses/index.php Zugriff: 02.11.2004
- GLÄSER, L. (2004): Open Source Software. Erlangen: Publicis Corporate Publishing.
- TYPO3.ORG (2004): <http://typo3.org/frontpage-menu-links/statistics> Zugriff: 05.10.2004
- HEISE-ONLINE (2004): www.heise.de/newsticker/result.xhtml?url=/newsticker/meldung/49735&words=M%FCnchen%20Linux Zugriff: 05.10.2004
- KRISHNAMURTHY, S. (2002): Cave or Community?: An empirical examination of 100 mature open source projects. First Monday 7 (6) firstmonday.org/issue7_6krishnamurthy/index.html Zugriff: 16.09.2004
- MITRE (2003): Use of FREE and Open-Source-Software (FOSS) in the US Department of Defence <http://www.microcross.com/dodfoss.pdf> Zugriff: 05.10.2004
- NETCRAFT (2004): „September 2004 web server Study“ news.netcraft.com/archives/web_server_survey.html Zugriff: 02.10.2004
- OSI (2004): www.opensource.org/licenses/index.php Zugriff: 02.10.2004
- PFUHL, M. (2004): www.connecta.ag/upload/rechtliche_aspekte_opensource.pdf Zugriff: 02.10.2004
- RAYMANS, H.G. (2001): MySQL im Einsatz mit ODBC, JDBC, PHP und Perl. Addison-Wesley. 617 S.
- RAYMOND, E.S. (1999): The cathedral and the bazaar: Musings on Linux and open source by an accidental revolutionary. Sebastopol: O'Reilly
- SCHPEP, H. & ZIERL, M. (2001): PHP und MySQL-Datenbank. - Internet-Professionell 1/2001: 24-30
- TYPO3.ORG (2004): typo3.org/frontpage-menu-links/statistics Zugriff: 05.10.2004
- UNILOG INTEGRA (2003): "Projekt Client Studie der Landeshauptstadt München – Kurzfassung des Abschlussberichtes inklusive Nachtrag" www.muenchen.de/aktuell/clientstudie_kurz.pdf Zugriff: 06.09.2004
- ZIERL, M. (2001): Dynamische Webseiten. - Internet-Professionell 1/2001: 6-8.

Kurzfassung

OS-Software ist zur Zeit ein vielfach diskutiertes Thema in Wissenschaft und Praxis. Das eigentliche Phänomen von OS ist, dass trotz der größtenteils freiwilligen und unentgeltlichen Beiträge einzelner Entwickler und des scheinbar unorganisierten und kollektiven Entwicklungsprozesses, technisch überzeugende und konkurrenzfähige Ergebnisse hervorgebracht werden. Immer häufiger entscheiden sich deswegen private Nutzer, Unternehmen wie auch öffentliche Einrichtungen für den Einsatz von OS-Software.

Die Gründe für diese beachtlichen Erfolge von OS-Produkten sind vielfältig. Die am häufigsten aufgeführten Argu-

mente, die für ihren Einsatz sprechen, sind die wegfallenden Lizenzkosten, erhöhte IT-Sicherheit, Offenheit und Erweiterbarkeit der Systeme sowie Unabhängigkeit von monopolistischen Softwareanbietern. Trotz teilweise erheblicher Vorteile beim Einsatz von OS-Produkten gibt es auch einige Problemfelder und Risiken, die es bei einer Entscheidungsfindung bezüglich der Softwareauswahl zu beachten gilt. Zunächst wird vorgestellt, was die wesentlichen Charakteristika einer OS-Software sind. Anschließend werden die Chancen und Risiken beim Gebrauch von OS zusammengefasst, die bei einer Entscheidungsfindung behilflich sein sollen. Im letzten Teil wird exemplarisch ein OS-Projekt vorgestellt, bei dem das Content-Management-System Typo3 zum Einsatz kam.

Stichworte: Typo3, Open-Source Software, Patentrechtsrisiko, Auswahlentscheidung, Total Cost of ownership

Summary

At present open source software is a highly discussed topic in organisations and sciences. Despite of the voluntary and apparently unorganized development process, open source software projects lead to technically advanced and competitive software solutions.

Consequently an increasing number of organisations, users and other institutions decide to use open source software products. The most frequently mentioned arguments for the use of open source solutions are low license fees, IT security, independence as well as openness and

expandability of the softwares. Besides of these advantages there are several problems and risks, which should be evaluated carefully when deciding to use open source software.

The article summerizes the main characteristics including opportunities and risks of open source software. Furthermore the article includes a case study of a typical open source project where the content management system Typo3 was implemented.

Keywords: Typo3, open source software, risk of patent law, decision making, total cost of ownership

Autoren

Dipl.-Betriebswirt (FH) Alexander Wehrum,

Vorstand der Connecta AG,

Peter Eulberg

Connecta AG

Rheinstraße 17-21

D-65185 Wiesbaden

Fon: +49 (0)611-34109-0 Fax: +49 (0)611-34109-29

Email: a.wehrum@connecta.ag

Dr. Claus Mücke

Justus-Liebig-Universität, Biometrie und Populationsgenetik
Interdisziplinäres Forschungszentrum für Umweltsicherung
(IFZ)

Heinrich-Buff-Ring 26-32

D-35392 Giessen

Fon: +49 (0)641/ 99 37542 Fax: +49 (0)641/ 99 37549

Email: claus.mueckschel@agr.uni-giessen.de