

Diplomarbeit

Analyse von Ruby on Rails 3 Web Content Management Systemen

Stephan Keller

16. September 2011

Hochschule für Technik, Wirtschaft und Kultur Leipzig (FH)

Dank an
Professor Dr. Ing. Robert Müller, meine Eltern Uta und Uwe
Keller sowie meinem Bruder Michael Keller

Inhaltsverzeichnis

1	Einleitung	4
1.1	Ausgangslage	4
1.2	Motivation und Zielsetzung	5
1.3	Aufbau der Arbeit	6
2	Analyse der bestehenden Content Management Systeme	7
2.1	Vorstellung Alchemy CMS	7
2.1.1	Funktionsprinzipien	8
2.1.2	Erweiterungen	8
2.1.3	Verwendete Technologien	8
2.2	Vorstellung Browser CMS	8
2.2.1	Funktionsprinzipien	8
2.2.2	Erweiterungen	8
2.2.3	Verwendete Technologien	8
2.3	Vorstellung Lokomotive CMS	8
2.3.1	Funktionsprinzipien	8
2.3.2	Erweiterungen	8
2.3.3	Verwendete Technologien	8
2.4	Vorstellung Refinery CMS	8
2.4.1	Funktionsprinzipien	8
2.4.2	Erweiterungen	8
2.4.3	Verwendete Technologien	8
3	Lösungsvorschläge	27
3.1	Übertragung der Konzepte des Domain Driven Design auf Ruby on Rails	27
3.2	Implementierung eines Ruby on Rails Java Content Repository	27
3.3	Übertragung des Typo3 5.0 Phoenix User-Interfaces in Rails 3.1	27

1 Einleitung

1.1 Ausgangslage

Die Skriptsprache PHP gehört weltweit zu den meist genutzten serverseitigen Programmiersprachen. Im August 2011 sind über 75 Prozent der dynamisch generierten Internetseiten mit dem PHP Hypertext Preprocessor erzeugt wurden¹.

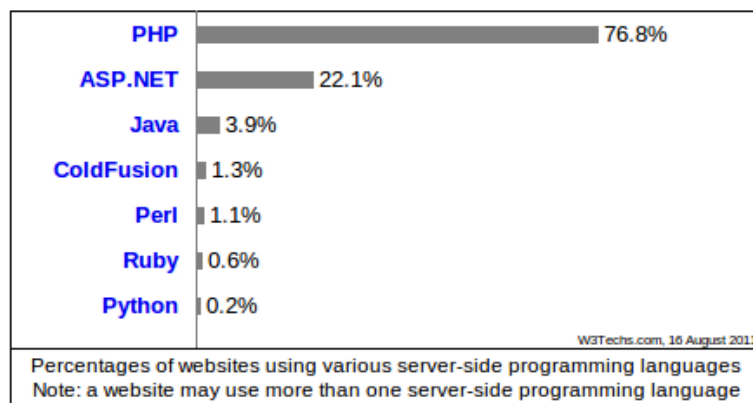


Abbildung 1.1: Nutzung verschiedener Programmiersprachen auf Servern

Auch im Bereich der Web Content Management Systeme² spiegelt sich diese Dominanz wider. Betrachtet man die Angaben des Content Management Portals cmsmatrix.org³, existieren neben den vor allem in Deutschland verwendeten Open Source-Lösungen Typo3, Drupal, Contao oder Joomla! über 500 weitere in PHP implementierte Web Content

¹W3tech erstellt täglich eine aktualisierte Auflistung über die Verwendung von serverseitigen Programmiersprachen. Es werden dabei die nach dem Alexa Ranking eine Million beliebtesten Internetseiten auf ihre Konfiguration untersucht.

²Im folgenden wird für den Begriff Web Content Management Systeme die Abkürzung WCMS verwendet

³<http://cmsmatrix.org> ermöglicht eine Gegenüberstellung der Funktionalitäten von Content Management Systemen unterschiedlicher Programmiersprachen.

Management Systeme unterschiedlichster Ausprägung und Qualität. Ruby als Programmiersprache findet hingegen nur bei etwa 1 Prozent der erfassten Server Verwendung. Die dabei umgesetzten Projekte sind meist individuelle, browser-basierte Applikationen, die für Unternehmen und deren spezifisches Geschäftsfeld entwickelt wurden. Bekannte Vertreter sind hier u.a. die webbasierte Projektmanagement-Applikation Basecamp von 37signals⁴, der Microblogging-Dienst Twitter⁵ und der webbasierte Hosting-Dienst Github⁶ für Software-Entwicklungsprojekte. Diese individuellen Lösungen werden dabei meist unter Zuhilfenahme eines Web Applications Frameworks realisiert, das den Entwicklungsprozess unterstützt und vereinfacht.

1.2 Motivation und Zielsetzung

Das Webframework Ruby on Rails⁷ hat sich seit der Veröffentlichung der Version 1.0 im Juli 2004 zu einem der bekanntesten Webframeworks der Ruby Fangemeinde entwickelt. Startups⁸ sowie etablierte Unternehmen greifen zunehmend auf das Rails Framework zurück, um ihre webbasierten Geschäftsideen und -modelle zu realisieren. Wird neben der Webapplikation zusätzlich eine Internetseite zur Repräsentierung der Unternehmung benötigt, haben sich in der Praxis folgende zwei Lösungsansätze herausgebildet:

1. Bei geringem Umfang der zusätzlichen Internetseite werden die Inhalte manuell in HTML-Dateien angelegt und anschließend in die Rails-Anwendung integriert. Komfortable Möglichkeiten der Content-Verwaltung werden nicht angeboten oder später rudimentär nach implementiert. Änderungen der Inhalte sind teilweise mit erhöhtem Aufwand verbunden oder erfordern zusätzliche Programmierkenntnisse⁹.
2. Komplexe Internetseiten mit vielen Inhalten werden über ein Web Content Management System eines Drittanbieters realisiert. Die Rails-Anwendung fungiert als Zwischenstation und leitet bestimmte Anfragen an das externe WCMS weiter.

⁴Projektseite von Basecamp: <http://basecamp.hq.com/>

⁵ Großteile der Programmierung von Twitter basierten bis April 2011 auf dem Ruby on Rails Framework.

⁶Github greift neben Ruby on Rails noch Webframeworks und Technologien zurück.

⁷Im weiteren Verlauf dieser Arbeit wird für das Webframework Ruby on Rails die Kurzform Rails verwendet.

⁸Der Ebgriff Startup bezeichnet hier junge Unternehmen, die sich mit ihrem neuartigen, meist innovativen Produkt noch nicht am Markt etabliert haben.

⁹Änderungen am Quellcode von Rails-Anwendungen erfordern immer einen Neustart der Anwendung.

Während der erste Lösungsansatz bei wenigen Inhalten noch vertretbar ist, erfordert die Verwendung eines externen WCMS zusätzlichen Installations- und Wartungsaufwand. Weiterhin erhöht sich der Bedarf an Programmierern, da neben Ruby nun auch andere Programmiersprachen Verwendung finden können.

Ziel der vorliegende Arbeit ist es daher, die Möglichkeiten einer rails-basierten Web Content Management Verwaltung zu untersuchen, um so den Einsatz eines externen WCMS überflüssig werden zu lassen.

1.3 Aufbau der Arbeit

Die vorliegende Diplomarbeit gliedert sich in sechs wesentliche Abschnitte.

Im ersten Abschnitt, in Kapitel 2, werden die für diese Arbeit notwendigen theoretische Grundlagen zu Web Content Management Systemen und dem Web Framework Ruby on Rails geschaffen. Es wird ebenfalls auf die neuen Funktionalitäten der Version 3.1 des Frameworks eingegangen. Zusätzlich werden die grundlegenden Charakteristika der Programmiersprache Ruby erläutert. Im zweiten Abschnitt, in Kapitel 3, folgt eine Analyse der bestehenden Ruby on Rails 3 Web Content Management Systeme Alchemy, RefineryCMS, BrowserCMS und Lokomotive CMS. Die ausgewählten Systeme werden dabei vorgestellt und anschließend mit Hilfe eines externen Kriterienkatalogs auf ihre Leistungsfähigkeit untersucht. Kapitel 4 überprüft die analysierten WCM-Systeme auf eventuell vorhandene konzeptionelle und programmiertechnische Schwachstellen. Darauf aufbauend werden in Kapitel 5 mögliche Lösungsansätze demonstriert und dafür notwendige theoretische Grundlagen herausgearbeitet. Kapitel 6 schließt die Arbeit mit einem Ausblick auf weitere Entwicklungen sowie einem Fazit ab.

2 Analyse der bestehenden Content Management Systeme

2.1 Vorstellung Alchemy CMS

Alchemy CMS ist ein von der Hamburger Internetagentur magiclabs* entwickeltes Open Source Content Management System. Es wird unter der GPLv3 Lizenz veröffentlicht und verfügt über einen relativ hohen Funktionsumfang. Die aktuelle Version 1.6.0 ist in den Rails-Versionen 2 und 3 erhältlich¹. Die von magiclabs* angebotene Demoversion ist zum Zeitpunkt der Erstellung dieser Arbeit nur als Rails 2 Variante verfügbar. Eine lokale Testinstallation konnte jedoch eine erfolgreiche Inbetriebnahme der Rails 3 Version sicherstellen.

2.1.1 Funktionsprinzipien

2.1.2 Erweiterungen

2.1.3 Verwendete Technologien

2.2 Vorstellung Browser CMS

2.2.1 Funktionsprinzipien

2.2.2 Erweiterungen

2.2.3 Verwendete Technologien

2.3 Vorstellung Lokomotive CMS

2.3.1 Funktionsprinzipien

2.3.2 Erweiterungen

2.3.3 Verwendete Technologien

2.4 Vorstellung Refinery CMS

2.4.1 Funktionsprinzipien

2.4.2 Erweiterungen

2.4.3 Verwendete Technologien

Inhalte sollen – unabhängig von Zeit- und Standort – durch mehrere Benutzer online verwaltet und erfasst werden können			
Alchemy 1.6.0	Erfüllt: 100%	Refinery CMS 1.0.8	Erfüllt: 100%
Vollständig unterstützt		Vollständig unterstützt	
BrowserCMS 3.3.1	Erfüllt: 100%	Lokomotive CMS	Erfüllt: 100%
Vollständig unterstützt		Vollständig unterstützt	

Integrierte Mediendatenbank zur Erfassung und Verwaltung von Bildern, Multimedia, Texten, Audio, Videos, usw.			
Alchemy 1.6.0	Erfüllt: 100%	Refinery CMS 1.0.8	Erfüllt: 100%
Alchemy bietet eine Bibliothek, in der Bilder und Dateien verwaltet werden können. Eine Angabe zusätzlicher Metadaten zu diesen Ressourcen ist nicht möglich.		Refinery CMS bietet eine einfache Medienverwaltung. Dabei wird zwischen Bildern und anderen Dateien unterschieden. Metadaten können nicht verwaltet werden.	
BrowserCMS 3.3.1	Erfüllt: 100%	Lokomotive CMS	Erfüllt: 100%
Browser CMS verfügt über eine <i>Content Library</i> , die eine einfache Medienverwaltung von Bildern, Dateien und definierten Inhaltselementen ermöglicht. Eine Metadatenverwaltung ist nicht vorhanden.		Locomotive CMS bietet eine Asset-Verwaltung, in der selbst erstellte Inhaltselemente in Containern verwaltet werden können. Das Hinzufügen von beliebigen Metainformationen ist möglich.	

Inhalte sollen ohne spezielle Programmier / HTML-Kenntnisse erfasst und verwaltet werden können			
Alchemy 1.6.0	Erfüllt: 100%	Refinery CMS 1.0.8	Erfüllt: 100%
Alle Inhalte können über den TinyMCE-Javascript WYSIWYG Editor erfasst und formatiert werden.		Alle Inhalte können über den integrierten WYSIWYG-Editor Wymeditor erfasst und formatiert werden. Der Editor ist fest in das System integriert und kann nicht ausgetauscht werden. Ein Plugin, dass die Verwendung eines anderen Editors ermöglicht ist bereits in Planung.	
BrowserCMS 3.3.1	Erfüllt: 100%	Lokomotive CMS	Erfüllt: 100%
In BrowserCMS findet der WYSIWIG-FCKEditor Verwendung. Er kann über installierbare Module beliebig ausgetauscht werden.		Alle Inhalte können über zwei integrierte WYSIWYG-Editoren erfasst und formatiert werden. Im Backend steht der Javascript Editor TinyMCE zur Verfügung. Im Frontend findet der HTML5-WYSIWYG-Editor Aloha zur Manipulierung der Seiteninhalte Verwendung (befindet sich noch in der Entwicklung).	

Inhalte sollen in einer Datenbank gespeichert werden			
Alchemy 1.6.0	Erfüllt: 100%	Refinery CMS 1.0.8	Erfüllt: 100%
Alchemy verwendet Active Record als Datenbankpersistenzschicht. Durch die Verwendung von Migrationen können so eine Vielzahl relationaler Datenbanken unterstützt werden. Zusätzlich existieren einige Adapter, um auch dokumentenbasierte Datenbanken anzusteuern.		Refinery greift ebenfalls auf Rails' Active Record zurück und unterstützt damit mehrere relationale und dokumentenorientierte Datenbanken.	
BrowserCMS 3.3.1	Erfüllt: 100%	Lokomotive CMS	Erfüllt: 100%
Wie bei Alchemy und Refinery CMS wird hier auch auf Active Record zurückgegriffen. Die Entwickler garantieren auf Grund fehlender Tests jedoch nur die Unterstützung von SQLite und MySQL-Datenbanken. Tendenziell können aber alle von Active Record unterstützten Datenbanken eingesetzt werden.		Lokomotive CMS greift im Gegensatz zu seinen Konkurrenten auf die dokumentenorientierten Datenbank MongoDB zurück. Relationale Datenbanken werden somit nicht unterstützt. Eine Umsetzung von Lokomotive mit Active Record ist jedoch geplant.	

Mehrere Benutzer sollen gleichzeitig Inhalte verwalten und erfassen können			
Alchemy 1.6.0	Erfüllt: 100%	Refinery CMS 1.0.8	Erfüllt: 100%
Nutzer und Administratoren können Inhalte gleichzeitig erfassen und verwalten.		Nutzer und Administratoren können Inhalte gleichzeitig erfassen und verwalten. Beim Zugriff zwei oder mehrerer Nutzer auf den gleichen Inhalt führt dazu, dass die Änderungen gültig werden, die zu letzt abgespeichert wurden. Optimistisches und pessimistisches Locking von Datensätzen und Inhalten wird nicht unterstützt.	
BrowserCMS 3.3.1	Erfüllt: 100%	Lokomotive CMS	Erfüllt: 100%
Nutzer und Administratoren können Inhalte gleichzeitig erfassen und verwalten. Beim Zugriff zwei oder mehrerer Nutzer auf den gleichen Inhalt führt dazu, dass die Änderungen gültig werden, die zu letzt abgespeichert wurden. Optimistisches und pessimistisches Locking von Datensätzen und Inhalten wird nicht unterstützt.		Nutzer und Administratoren können Inhalte gleichzeitig erfassen und verwalten. Beim Zugriff zwei oder mehrerer Nutzer auf den gleichen Inhalt führt dazu, dass die Änderungen gültig werden, die zu letzt abgespeichert wurden. Optimistisches und pessimistisches Locking von Datensätzen und Inhalten wird nicht unterstützt.	

Inhalte (Texte, Bilder, Videos etc.) sollen zentral kategorisiert, erfasst und verwaltet werden können			
Alchemy 1.6.0	Erfüllt: 0%	Refinery CMS 1.0.8	Erfüllt: 0%
Die Bibliothek von Alchemy unterstützt lediglich eine Auflistung von Ressourcen. Bilder und Dateien können damit nur in Form einer Listenansicht inspiziert werden. Eine Zuordnung zu Kategorien oder eine Anlage von Ordnerstrukturen zur Erleichterung der Orientierung ist nicht möglich. Die Verwaltung großer Datenmengen scheint daher nur schwer möglich.		Ähnlich wie bei Alchemy gleicht die Ressourcenverwaltung nur einer einfachen Auflistung von Bildern und anderen Ressourcen. Eine Kategorisierung der Inhalte ist nicht möglich. Ebenfalls können keine Ordner zur sinnvollen Strukturierung der Ressourcen erstellt werden. Die Verwaltung großer Datenmengen wird dadurch schnell zu einem Geduldsakt.	
BrowserCMS 3.3.1	Erfüllt: 0%	Lokomotive CMS	Erfüllt: 0%
Die <i>Content Library</i> von BrowserCMS listet wie ihre Vorgänger lediglich die angelegten Bilder oder Dateien auf. Möglichkeiten zur sinnvollen Organisation (Kategorien, Ordner) großer Datenmengen sind nicht vorhanden.		Die Inhaltsverwaltung kann nicht kategorisiert werden. Wie bei seinen Vorgängern sind die Datensätze lediglich in Listenform aufgeführt. Eine logische Strukturierung mit Hilfe von Ordnern ist nicht möglich.	

Inhalte können während der Erfassung über eine Preview-Funktion vorab im Design der Webseite angesehen werden			
Alchemy 1.6.0	Erfüllt: 100%	Refinery CMS 1.0.8	Erfüllt: 0%
Redakteure können ihre erstellten und editierten Inhalte im Backend durch ein Preview-Fenster sichtbar machen. Änderungen an Inhaltselementen können somit sofort nachvollzogen werden.		Refinery CMS verfügt über keine Preview-Funktion der Inhalte. Ist ein Inhaltselement im Backend neu angelegt oder bearbeitet wurden, wird dies auf der Internetseite sofort sichtbar.	
BrowserCMS 3.3.1	Erfüllt: 100%	Lokomotive CMS	Erfüllt: 0%
Wie bei Alchemy werden Inhalte erst nach ihrer Veröffentlichung sichtbar. Bis dahin kann jedoch im Frontend durch Inline-Editing der Seite jedes Inhaltselement bearbeitet werden.		Lokomotive CMS bietet wie RefineryCMS keine Preview-Funktion. Änderungen und neu angelegte Inhalte werden direkt veröffentlicht.	

Zuordnung von standardisierten und frei definierbaren Metadaten zu Inhalten (z.B. Autor, Schlüsselwörter, Benutzerdefinierte Felder) soll möglich sein			
Alchemy 1.6.0	Erfüllt: 0%	Refinery CMS 1.0.8	Erfüllt: 0%
Metadaten zu Inhalten können nicht vergeben werden.		Inhalte werden als einfache Datensätze betrachtet und besitzen daher keine definierten Metadaten.	
BrowserCMS 3.3.1	Erfüllt: 100%	Lokomotive CMS	Erfüllt: 100%
Metadaten können zu einzelnen Inhaltselementen in Form einer Tag-Liste hinzugefügt werden. Diese wird in der Datenbank als Text abgespeichert und bei ihrer Nutzung in einzelne Teil-Strings zerlegt. Das Hinzufügen zuvor definierter Metadaten ist nicht möglich.		Zu den verschiedenen Inhaltselementen können beliebig viele Metadaten hinzugefügt werden. Auch die Darstellung von 1:1 und 1:n-Beziehungen ist möglich. Diese Funktionalität wird dabei vor allem durch die Verwendung der dokumentenbasierten Datenbank MongoDB möglich.	

Inhalte sollen mehrsprachig erfasst und verwaltet werden können			
Alchemy 1.6.0	Erfüllt: 100%	Refinery CMS 1.0.8	Erfüllt: 100%
In Alchemy können Inhalte mehrsprachig angelegt werden. Durch die Auswahl einer bestimmten Sprache wird ein entsprechender Seitenbaum mit allen existierenden Inhalten zu der ausgewählten Sprache erzeugt.		Refinery CMS kann Inhalte mehrsprachig verwalten und ausgeben. Zur Aktivierung der Funktionalität müssen nur die zu unterstützenden Sprachen in einer Konfigurationsdatei angegeben werden (dies kann von Administratoren im Backend vorgenommen werden). Alle Sprachen werden dabei in einem einzigen Seitenbaum verwaltet. Vorhandene Übersetzungen zu einer bestimmten Seite werden durch Einblendung kleiner Flaggensymbole kenntlich gemacht.	
BrowserCMS 3.3.1	Erfüllt: 0%	Lokomotive CMS	Erfüllt: 0%
In BrowserCMS kann durch die Installation der Erweiterung <i>browsercmsi</i> die Unterstützung von mehrsprachigen Inhalten erreicht werden. Der Plugin-Anbieter konnte die 100% Rails 3-Kompatibilität der Erweiterung jedoch nicht garantieren. Von einem Einsatz dieser Lösung in einer Produktiv-Umgebung wird daher abgeraten. Innerhalb der Bewertung von Brower CMS werden daher 0% beim Erfüllungsgrad angegeben.		Inhalte können nur einsprachig verwaltet werden. Erweiterungen, die diese Funktionalität herstellen können, existieren ebenfalls nicht.	

Das CMS soll über eine offene API (Programmierschnittstelle) für individuelle Erweiterung verfügen			
Alchemy 1.6.0	Erfüllt: 100%	Refinery CMS 1.0.8	Erfüllt: 100%
Eine flexible Plugin-DSL erlaubt das Hinzufügen von individuellen Erweiterungen.		Individuelle Inhaltselemente können durch die Verwendung der Refinery Engine Generatoren hinzugefügt werden. Für die weitere Entwicklung stehen die in Rails üblichen Entwicklungstechniken zur Verfügung.	
BrowserCMS 3.3.1	Erfüllt: 100%	Lokomotive CMS	Erfüllt: 100%
Ähnlich wie bei Refinery CMS können neue Module und Inhaltstypen mit Hilfe von speziellen Rails-Generatoren erzeugt werden.		Neue Inhaltstypen lassen sich im Backend durch ein einfaches User-Interface zusammenstellen. Mit wenigen Klicks sind so schnell neue Elemente erstellt. Programmierkenntnisse sind nicht notwendig.	

Für die Verwaltung und Erfassung von Inhalten sollen alle gängigen Internet-Browser (Internet Explorer, Safari und Firefox) eingesetzt werden können			
Alchemy 1.6.0	Erfüllt: 100%	Refinery CMS 1.0.8	Erfüllt: 100%
Vollständig unterstützt		Vollständig unterstützt	
BrowserCMS 3.3.1	Erfüllt: 100%	Lokomotive CMS	Erfüllt: 100%
Vollständig unterstützt		Vollständig unterstützt	

Inhalte sollen einfach importiert / exportiert werden können - dabei kommen Formate wie z.B. XML zum Einsatz			
Alchemy 1.6.0	Erfüllt: 0%	Refinery CMS 1.0.8	Erfüllt: 0%
Alchemy verfügt über keine integrierten Import und Export-Funktionalitäten.		Es existieren auf Nutzerebene keine Möglichkeiten des Im- und Exports. Durch sogenannte Seed-Dateien ist jedoch ein nachträgliches Befüllen der Datenbank möglich. Der Aufruf erfordert jedoch Kenntnisse in Ruby on Rails und ist daher für Normalanwender/Redakteure nicht sinnvoll.	
BrowserCMS 3.3.1	Erfüllt: 0%	Lokomotive CMS	Erfüllt: 100%
In Browser CMS können Inhalte nicht importiert und exportiert werden. Entsprechende Features müssten erst eigenständig implementiert werden.		In Lokomotive CMS kann ein kompletter Internetauftritt mit seinen Inhalten und Ressourcen importiert und exportiert werden. Zum Austausch der Inhalte findet eine <i>Zip</i> -Datei Verwendung, die alle benötigten Ressourcen (Bilder, Dateien, Templates usw.) sowie Inhalte der Datenbank einschließt. Ressourcen werden dabei in vordefinierten Ordnerstrukturen abgelegt. Die Datenbankeinträge aus MongoDB werden innerhalb der <i>Zip</i> -Datei im Unterordner <i>data</i> abgelegt. Die Einträge liegen dabei im <i>YAML</i> -Format vor.	

Integration von Inhalten anderer Webseiten, Multimedia, Applikationen, E-Commerce-Tools			
Alchemy 1.6.0	Erfüllt: 100%	Refinery CMS 1.0.8	Erfüllt: 100%
<p>Der verwendete WYSIWYG-Editor <i>TinyMCE</i> erlaubt in seiner HTML-Ansicht das Einbinden von Fremdinhalten anderer Seiten (z.B. IFrame). Zusätzlich ist die Erstellung von eigenen Inhaltselementen mit Hilfe der Alchemy Plugin DSL-API denkbar. So können auch die verschiedenen Ressourcen aus der Bibliothek von Alchemy Verwendung finden. Standardmäßig verfügt Alchemy bereits über die Inhaltselemente <i>Artikel</i>, <i>Text</i>, <i>Text mit Bild</i>, <i>Bilder</i>, <i>Bildergalerie</i>, <i>Überschrift</i> und <i>Intro</i>.</p>		<p>Refinery CMS verwaltet jede Internetseite innerhalb eines flexiblen WYSIWYG-Editors. Die Integration von vordefiniertem HTML-Code kann dabei durch Nutzung der HTML-Ansicht des Editors erreicht werden.</p>	
BrowserCMS 3.3.1	Erfüllt: 100%	Lokomotive CMS	Erfüllt: 100%
<p>Wie seine Vorgänger auch können innerhalb des WYSIWYG-Editors IFrames oder anderer HTML-Code eingebettet werden. Vordefinierte Inhaltselementen, die vorhandene Ressourcen aus der <i>Content Library</i> einbinden können, müssen eigenhändig angelegt werden.</p>		<p>Wie bei Alchemy und Refinery CMS können innerhalb des WYSIWYG-Editors HTML-Fragmente angegeben werden.</p>	

Granulares Rechte- und Rollenkonzept für Anwender			
Alchemy 1.6.0	Erfüllt: 0%	Refinery CMS 1.0.8	Erfüllt: 0%
In Alchemy existieren vordefinierte Rollen (Registriert, Author, Redakteur, Administrator). Das Anlegen weiterer Rollen zur besseren Differenzierung ist jedoch nicht möglich.		Refinery CMS besitzt kein Rechte- und Rollenkonzept. Dem Anwender kann lediglich der Zugang zu bestimmten Plugins erlaubt oder entzogen werden, um so den Funktionsumfang einzuschränken.	
BrowserCMS 3.3.1	Erfüllt: 100%	Lokomotive CMS	Erfüllt: 0%
In Browser CMS wird in einer Standardinstallation zwischen den Rollen Gast, CMS Administrator und Content Editor unterschieden. Zusätzliche können weitere Backend-Gruppen angelegt werden.		Lokomotive CMS besitzt ein einfaches Rechte- und Rollenkonzept. Es wird zwischen Administratoren, Designern und Autoren unterschieden. Das Anlegen weiterer Gruppen ist nicht möglich.	

Granulares Berechtigungskonzept für einzelne Inhalte, Bereiche, Webseiten			
Alchemy 1.6.0	Erfüllt: 0%	Refinery CMS 1.0.8	Erfüllt: 0%
Die in Alchemy vordefinierte Rollen Registriert, Author, Redakteur und Administrator bestimmen den Funktionsumfang eines Anwenders im Backend. Angelegte Inhalte können jedoch nicht einzelnen Nutzern zugeordnet werden.		Nutzer können alle Inhalte und Bereiche einer Webseite editieren, solange sie zur Nutzung des bestimmten Plugins berechtigt wurden.	
BrowserCMS 3.3.1	Erfüllt: 0%	Lokomotive CMS	Erfüllt: 0%
Der Zugriff auf bestimmte Seiten (Seitenbaumzweige) kann eingeschränkt werden. Zusätzlich bietet Browser CMS ein Erstellen von Frontend-Nutzergruppen an, um so bestimmte Seiten des CMS nur exklusiv ausgewählten Nutzern zur Verfügung zu stellen. Die Zugriffsberechtigung auf installierte Plugins kann ebenfalls für jeden Nutzer individuell festgelegt werden. Leider ist es nicht möglich, einzelne Inhaltselemente für bestimmte Nutzer unzugänglich zu machen.		Der Zugriff auf Seiten und Inhalte kann nicht individuell gesteuert und beeinflusst werden. Besitzt ein Anwender das Recht zum Editieren und Anlegen von Inhalten (Nutzergruppe Redakteur), können alle Inhalte im gesamten CMS bearbeitet werden.	

Schutz vor gegenseitigem Überschreiben erfasster Inhalte durch Check in/ Check out-Mechanismen			
Alchemy 1.6.0	Erfüllt: 0%	Refinery CMS 1.0.8	Erfüllt: 0%
Wird nicht unterstützt		Wird nicht unterstützt	
BrowserCMS 3.3.1	Erfüllt: 0%	Lokomotive CMS	Erfüllt: 0%
Wird nicht unterstützt. Die automatische Versionierung von Inhaltselementen erlaubt jedoch ein nachträgliches, manuelles Sichten und Zusammenfügen verschiedener Versionen.		Wird nicht unterstützt	

Versionierung von Inhalten mit Möglichkeit zur Wiederherstellung vorhergehender Versionen			
Alchemy 1.6.0	Erfüllt: 0%	Refinery CMS 1.0.8	Erfüllt: 0%
Versionierung und Wiederherstellung von Inhalten wird nicht unterstützt.		Versionierung und Wiederherstellung von Inhalten wird nicht unterstützt.	
BrowserCMS 3.3.1	Erfüllt: 100%	Lokomotive CMS	Erfüllt: 0%
Wird unterstützt		Versionierung und Wiederherstellung von Inhalten wird nicht unterstützt.	

Mandantenfähigkeit: Mehrfachnutzung des Systems durch verschiedene Parteien mit kompletter Trennung der Daten und Benutzer			
Alchemy 1.6.0	Erfüllt: 0%	Refinery CMS 1.0.8	Erfüllt: 0%
Wird nicht unterstützt		Wird nicht unterstützt	
BrowserCMS 3.3.1	Erfüllt: 0%	Lokomotive CMS	Erfüllt: 0%
Wird nicht unterstützt		In Lokomotive CMS können mehrere Internetauftritte gleichzeitig verwaltet werden . Eine Trennung der verschiedenen Nutzer und Daten wird jedoch nicht angeboten.	

Linküberprüfung: Automatische Prüfung der Gültigkeit von internen und externen Links mit Möglichkeit zur Korrektur bzw. Benachrichtigung definierter Personengruppen			
Alchemy 1.6.0	Erfüllt: 0%	Refinery CMS 1.0.8	Erfüllt: 0%
Wird nicht unterstützt		Wird nicht unterstützt	
BrowserCMS 3.3.1	Erfüllt: 0%	Lokomotive CMS	Erfüllt: 0%
Wird nicht unterstützt		Wird nicht unterstützt	

Definition von Workflows inkl. mehrstufiger Freigabeprozesse für die Freischaltung von Inhalten			
Alchemy 1.6.0	Erfüllt: 0%	Refinery CMS 1.0.8	Erfüllt: 0%
Wird nicht unterstützt		Wird nicht unterstützt	
BrowserCMS 3.3.1	Erfüllt: 0%	Lokomotive CMS	Erfüllt: 0%
Wird nicht unterstützt		Wird nicht unterstützt	

Möglichkeit für <i>nicht technische</i> User den Workflow zu kreieren, verwalten und zu ändern. Es soll dafür kein Scripting / Programming notwendig sein			
Alchemy 1.6.0	Erfüllt: 0%	Refinery CMS 1.0.8	Erfüllt: 0%
Nutzer können in Alchemy keinen Workflowprozess kreieren. Es ist jedoch möglich, dass Redakteure die von Autoren durchgeführten Änderungen kontrollieren und anschließend veröffentlichen. Ein Austausch zwischen beiden Nutzergruppen ist nicht möglich (z.B. kurze Mitteilung an den Redakteur). Redakteure müssen so Änderungen der Seiteninhalte selbst erkennen.		Wird nicht unterstützt	
BrowserCMS 3.3.1	Erfüllt: 0%	Lokomotive CMS	Erfüllt: 0%
Ein Workflowprozess kann in Browser CMS nicht erzeugt werden. Autoren ist es nur möglich, ihre durchgeführten Änderungen an andere Backend-Nutzer mit Veröffentlichungsrechten weiterzuleiten (Simulierung eines einfachsten Workflows).		Wird nicht unterstützt	

Möglichkeit externe Benutzer in Workflows mit einbinden zu können			
Alchemy 1.6.0	Erfüllt: 0%	Refinery CMS 1.0.8	Erfüllt: 0%
Wird nicht unterstützt		Wird nicht unterstützt	
BrowserCMS 3.3.1	Erfüllt: 0%	Lokomotive CMS	Erfüllt: 0%
Wird nicht unterstützt		Wird nicht unterstützt	

Unternehmensspezifische Bearbeitungsprozesse von Inhalten sollen über frei definierbare Workflows verwaltet werden können			
Alchemy 1.6.0	Erfüllt: 0%	Refinery CMS 1.0.8	Erfüllt: 0%
Wird nicht unterstützt		Wird nicht unterstützt	
BrowserCMS 3.3.1	Erfüllt: 0%	Lokomotive CMS	Erfüllt: 0%
Wird nicht unterstützt		Wird nicht unterstützt	

Trennung von Inhalt und Design unter Verwendung von Templates			
Alchemy 1.6.0	Erfüllt: 100%	Refinery CMS 1.0.8	Erfüllt: 100%
Inhalt und Design werden in Alchemy durch die Verwendung von <i>erb</i> -Templates getrennt. Das Haupt-Template der Seite wird zu Beginn der Entwicklung von einem Designer festgelegt und anschließend in der Anwendung verankert (als fixe Resource im Rails Quellcode). So können Redakteure das Aussehen der Internetseite nicht beeinflussen.		Wie Alchemy verwendet Refinery CMS <i>erb</i> als Template-Sprache. Trotz der so erreichten Trennung zwischen Inhalten und Design erschwert die fehlende Möglichkeit der Anpassung im Backend den Umgang mit dem gesamten WCMS. Dies gilt für das Haupttemplate der Seite sowie für alle Erweiterungen.	
BrowserCMS 3.3.1	Erfüllt: 100%	Lokomotive CMS	Erfüllt: 100%
BrowserCMS unterstützt die Verwendung verschiedener Template-Sprachen. In einer Standard-Installation werden so u.a. werden <i>erb</i> , <i>rjs</i> und <i>rxml</i> angeboten.		In Lokomotive CMS kann für jede Seite ein Template angegeben werden. Die dabei verwendete Templatesprache ist <i>Liquid</i> .	

Mehrfachverwendung von Inhalten an verschiedenen Stellen mit unterschiedlichem Layout			
Alchemy 1.6.0	Erfüllt: 100%	Refinery CMS 1.0.8	Erfüllt: 0%
Inhaltselemente und Seiten können in Alchemy kopiert und wiederverwendet werden. Die Zuordnung eines neuen Templates muss durch den Administrator erfolgen (Änderung am Rails-Quellcode).		Inhalte und Seiten können nicht kopiert und mehrfach verwendet werden.	
BrowserCMS 3.3.1	Erfüllt: 0%	Lokomotive CMS	Erfüllt: 0%
Es können nur Inhalte mehrfach verwendet werden.		In Lokomotive CMS kann nur für jede Seite ein neues Template angegeben werden. Inhalte sind den einzelnen Seiten zugeordnet und nur dort verwendbar.	

Navigationsstrukturen werden automatisch vom CMS generiert, publiziert und verwaltet			
Alchemy 1.6.0	Erfüllt: 100%	Refinery CMS 1.0.8	Erfüllt: 100%
Wird unterstützt		Wird unterstützt	
BrowserCMS 3.3.1	Erfüllt: 100%	Lokomotive CMS	Erfüllt: 100%
Wird unterstützt		Wird unterstützt	

Barrierefreiheit bei den publizierten Seiten soll unterstützt werden			
Alchemy 1.6.0	Erfüllt: 100%	Refinery CMS 1.0.8	Erfüllt: 100%
Barrierefreiheit kann bei entsprechender Umsetzung der Verwendeten Templates und CSS-Dateien erreicht werden.		Barrierefreiheit kann bei entsprechender Umsetzung der Verwendeten Templates und CSS-Dateien erreicht werden.	
BrowserCMS 3.3.1	Erfüllt: 100%	Lokomotive CMS	Erfüllt: 100%
Barrierefreiheit kann bei entsprechender Umsetzung der Verwendeten Templates und CSS-Dateien erreicht werden.		Barrierefreiheit kann bei entsprechender Umsetzung der Verwendeten Templates und CSS-Dateien erreicht werden.	

Inhalte sollen auf verschiedene Medien / Technologien (Cross Media Publishing, SMS /Mobile / WAP / usw.) ausgegeben werden können			
Alchemy 1.6.0	Erfüllt: 0%	Refinery CMS 1.0.8	Erfüllt: 0%
Wird nicht unterstützt		Wird nicht unterstützt	
BrowserCMS 3.3.1	Erfüllt: 0%	Lokomotive CMS	Erfüllt: 0%
Wird nicht unterstützt		Wird nicht unterstützt	

Möglichkeit Inhalte für anderen Webseiten bereitzustellen (XML, Webservice)			
Alchemy 1.6.0	Erfüllt: 100%	Refinery CMS 1.0.8	Erfüllt: 100%
Wird unterstützt		Wird unterstützt	
BrowserCMS 3.3.1	Erfüllt: 100%	Lokomotive CMS	Erfüllt: 100%
Wird unterstützt		Wird unterstützt	

Möglichkeit zur Wahl zwischen dynamischer oder statischer Generierung der Seiten / Inhalte			
Alchemy 1.6.0	Erfüllt: 0%	Refinery CMS 1.0.8	Erfüllt: 0%
Wird nicht unterstützt		Wird nicht unterstützt	
BrowserCMS 3.3.1	Erfüllt: 0%	Lokomotive CMS	Erfüllt: 0%
Wird nicht unterstützt		Wird nicht unterstützt	

Einfache Einbindung von Fremdinhalten welche durch Drittanbieter zur Verfügung gestellt werden			
Alchemy 1.6.0	Erfüllt: 100%	Refinery CMS 1.0.8	Erfüllt: 100%
Über den integrierten WYSIWYG-Editor können HTML-Fragmente eingebunden werden.		Über den integrierten WYSIWYG-Editor können HTML-Fragmente eingebunden werden.	
BrowserCMS 3.3.1	Erfüllt: 100%	Lokomotive CMS	Erfüllt: 100%
Über den integrierten WYSIWYG-Editor können HTML-Fragmente eingebunden werden.		Über den integrierten WYSIWYG-Editor können HTML-Fragmente eingebunden werden.	

Automatisches Anbieten von Druckversion und Weiterempfehlen einer Webseite			
Alchemy 1.6.0	Erfüllt: 0%	Refinery CMS 1.0.8	Erfüllt: 0%
Wird nicht unterstützt		Wird nicht unterstützt	
BrowserCMS 3.3.1	Erfüllt: 0%	Lokomotive CMS	Erfüllt: 0%
Wird nicht unterstützt		Wird nicht unterstützt	

Freie Wahl des Publikationszeitraumes (zeitgesteuertes Auf- / Abschalten / Archivieren) von Inhalten			
Alchemy 1.6.0	Erfüllt: 0%	Refinery CMS 1.0.8	Erfüllt: 0%
Wird nicht unterstützt		Wird nicht unterstützt	
BrowserCMS 3.3.1	Erfüllt: 0%	Lokomotive CMS	Erfüllt: 0%
Wird nicht unterstützt		Wird nicht unterstützt	

Inhalte sollen archiviert werden können.			
Alchemy 1.6.0	Erfüllt: 0%	Refinery CMS 1.0.8	Erfüllt: 0%
Wird nicht unterstützt		Wird nicht unterstützt	
BrowserCMS 3.3.1	Erfüllt: 100%	Lokomotive CMS	Erfüllt: 0%
Wird unterstützt		Wird nicht unterstützt	

3 Lösungsvorschläge

das ist ja geil

3.1 Übertragung der Konzepte des Domain Driven Design auf Ruby on Rails

das ist auch geil

3.2 Implementierung eines Ruby on Rails Java Content Repository

das ist auch cool

3.3 Übertragung des Typo3 5.0 Phoenix User-Interfaces in Rails 3.1

das auch am ende