



Hochschule für Technik, Wirtschaft und Kultur Leipzig (FH)  
Fachbereich Informatik, Mathematik und Naturwissenschaften

Diplomarbeit

**Konzeption eines  
Web Content Management Systems  
für den Fachbereich Informatik, Mathematik und  
Naturwissenschaften der HTWK Leipzig**

Betreuer: Prof. Dr. rer. nat. Klaus Hering  
Bearbeiter: Daniel Otto  
geboren am: 13.03.1979  
Matr.-Nr.: 27379  
dotto@imn.htwk-leipzig.de

Leipzig, den 09.12.2003

## **Eidesstattliche Erklärung**

Ich versichere, dass ich die Diplomarbeit ohne fremde Hilfe und ohne Benutzung anderer als der angegebenen Quellen angefertigt habe und dass die Arbeit in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegen hat. Alle Ausführungen der Arbeit, die wörtlich oder sinngemäß übernommen wurden, sind als solche gekennzeichnet.

Leipzig, den 09.12.2003

Daniel Otto

---

# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>6</b>
<b>2</b>	<b>Die Grundlagen von WCMSs</b>	<b>8</b>
2.1	Systemarten .....	8
2.2	Evolution und Gliederung .....	9
2.3	Der Versuch einer Definition .....	11
2.4	Der Content Life Cycle .....	13
2.5	Vor- und Nachteile .....	14
<b>3</b>	<b>Komponenten und Aufbau eines WCMSs</b>	<b>19</b>
3.1	Assetmanagement.....	20
3.1.1	Abspeicherung des Inhaltes .....	20
3.1.2	Bearbeiten des Inhaltes .....	21
3.1.3	Automatisierungen.....	22
3.1.4	Versionierung .....	23
3.1.5	Länder- und Sprachabhängigkeit.....	23
3.2	Die Workflowkomponente.....	24
3.2.1	Workflowarten .....	24
3.2.2	Das Rechtekonzept .....	24
3.2.3	Geschütztes Editieren .....	25
3.2.4	Werkzeuge des Workflows.....	25
3.2.5	Freigabeverfahren .....	26
3.3	Zugriffs- und Benutzerverwaltung.....	28
3.4	Im- und Exportschnittstellen für Content.....	28
3.5	Programmierschnittstellen.....	29
3.6	Serverkonzepte .....	29
3.6.1	Dynamic Publishing.....	29
3.6.2	Publishing-/ Staging-Server.....	30
3.6.3	Publishing-/ QA-/ Staging-Server.....	31
3.6.4	Applikationsserver .....	32

---

<b>4 Die Auswahl</b>	<b>33</b>
4.1 Wieso ein WCMS für den Fachbereich.....	33
4.2 Überblick über Content Management Systeme.....	35
4.3 Fremdsystemanschaffung oder Eigenentwicklung.....	37
4.4 Umsetzungssprachen.....	40
<b>5 Vorgehensweise bei der Erstellung eines WCMS</b>	<b>42</b>
5.1 Kompetenzen.....	42
5.2 Briefing.....	48
5.3 Konzeption .....	48
5.3.1 Zielsetzung – Vorbereitungsphase.....	48
5.3.2 Inhaltliches Konzept .....	49
5.3.3 Informationsmanagement-Konzept .....	49
5.3.4 Redaktions- und Rechtekonzept .....	50
5.3.5 Media-Konzept .....	50
5.3.6 Technisches Konzept.....	51
5.3.7 Deployment-Konzept.....	52
5.3.8 Schulungskonzept.....	52
5.3.9 Betriebskonzept .....	52
5.4 Programmierung.....	53
5.5 Testphase.....	53
5.6 Abnahme bzw. Freigabe.....	54
5.7 Onlinebetrieb.....	54
<b>6 Die Erstellung</b>	<b>55</b>
6.1 Kompetenzen.....	55
6.2 Briefing.....	56
6.3 Konzeption .....	57
6.3.1 Zielsetzung – Vorbereitungsphase.....	57
6.3.2 Inhaltliches Konzept .....	58
6.3.3 Informationsmanagement-Konzept .....	66
6.3.4 Redaktions- und Rechtekonzept .....	67
6.3.5 Media-Konzept .....	76
6.3.6 Technisches Konzept.....	78
6.3.7 Deployment-Konzept.....	79

6.3.8	Schulungskonzept .....	79
6.3.9	Betriebskonzept .....	80
6.4	Programmierung .....	80
6.5	Testphase .....	81
6.6	Abnahme bzw. Freigabe .....	82
6.7	Onlinebetrieb .....	82
<b>7</b>	<b>Programmierbeispiel</b>	<b>83</b>
7.1	Vorwort .....	83
7.2	Aufbau des Grundsystems .....	83
7.3	Beispiel .....	86
7.4	Ausblick .....	87
<b>8</b>	<b>Resümee</b>	<b>90</b>
	<b>Akronyme</b>	<b>91</b>
	<b>Glossar</b>	<b>93</b>
	<b>Abbildungs- und Tabellenverzeichnis</b>	<b>95</b>
	<b>Literaturverzeichnis</b>	<b>96</b>
	<b>Anhang A: Datenbankschema</b>	<b>99</b>
	<b>Anhang B: Tabellenbeschreibung</b>	<b>100</b>
	<b>Anhang C: Inhalt der CD</b>	<b>103</b>

# 1 Einleitung

Die Internetpräsenz des Fachbereiches Informatik, Mathematik und Naturwissenschaften der Hochschule für Technik, Wirtschaft und Kultur Leipzig (FH) enthält zunehmend dynamisch publizierte Informationen, die nicht durch verschiedene Personen erstellt, bearbeitet, gelöscht und veröffentlicht werden können. Ebenfalls wird die Attraktivität der Seite unter anderem durch den statischen Aufbau, die nicht frequenten Publikationen, Informationsmangel, Nichtaktualität, Mangel an Interaktions- und Kommunikationsmöglichkeiten und das nicht einheitliche Design gemindert. Eine Fachbereichsseite sollte zum Beispiel mit einem Rechtesystem ausgestattet sein. Ein solches System würde einen internen Bereich ermöglichen, welcher für hochschulfremde Personen nicht zugänglich wäre. Dadurch könnten den Professoren, Mitarbeitern und Studenten spezifizierte Informationen (z. B. Notenlisten, Veranstaltungen, Stundenausfälle, Technikinformationen usw.) und eine interne Kommunikationsplattform angeboten werden.

Die Umsetzung einer solchen Internetpräsenz, wobei die Kontrolle, Nachbearbeitung und Publikation der erstellten Informationen sowie die Programmierung der kompletten Seite und Administration des Webservers von den Webmastern abhängig ist, ist nicht vorstellbar. Das herkömmliche Webpublishing ist für komplexere Seiten weniger geeignet, da eine effektive Verwaltung großer Informationsbestände, die Administration von Personen und die damit verbundene Sicherheit nicht bereitgestellt werden können. Daher stützt sich meine Konzeption der Fachbereichsseite auf ein Web Content Management System (WCMS). Dabei wird es nicht nur für die effiziente Verwaltung von großen Inhaltsmengen der Webseite verantwortlich sein. Bei einer erfolgreichen Umsetzung garantiert es unter anderem Qualität, Effizienz, Aktualität und Sicherheit, welche bei dem herkömmlichen Web-Publishing nicht in diesem Maße der Fall sind. [vgl. Zschau et al. 2002, S. 76-78; Krüger/ Kopp 2002, S. 37] Die integrierten Automatisierungstechniken, die Einhaltung eines einheitlichen Designs und Aufbaues der gesamten Webseite, die Konvertierung in verschiedene Ausgabeformate und Einbeziehung von technisch nicht versierten Autoren in den Publishingprozess sind nur einige Gründe, die für die Umsetzung der Internetpräsenz eines Fachbereiches mittels eines WCMS sprechen.

Meine Arbeit beschäftigt sich neben der Erklärung der Grundlagen und des Aufbaues eines Web Content Management Systems hauptsächlich mit der Konzeption eines Basis-WCMS für den Fachbereich IMN und einer damit verbundenen Eigenentwicklung. Bei der Konzeption lege ich besonderen Wert auf die Kompetenzeinteilung und den Entwurf des Inhaltes mit den dazugehörigen Rechten der jeweiligen Personen.

Um ein WCMS zu erzeugen, muss man möglichst viele Tätigkeiten formalisieren, damit sie zur Steuerung des Systems dienen können. Bei der erstmaligen Definition der Produktionsprozesse merkt man schnell, dass dies in eine organisatorische Komplexität degeneriert [vgl. Knobloch 2003, S. 33]. Deshalb muss man seine eigenen Anforderungen an ein WCMS klar und detailliert dokumentieren. Eine Eingrenzung der Anforderungen führt daher eher zu einem erfolgreichen und erweiterbaren Produkt. Auf Grund dessen gebe ich mit meiner Arbeit eine Grundlage eines WCMS, welches den Grundanforderungen gerecht wird und die Möglichkeit von Erweiterungen bietet.

Die programmiertechnische Umsetzung meiner Arbeit beschränkt sich vorwiegend auf die Datenbank und Funktionen, welche zeigen, wie sich ein Rechtesystem auf ein von mir gewähltes Beispiel auswirkt. Dabei werden grundlegende Aufgaben, wie beispielsweise das Bearbeiten oder die Rechtezuweisung auf Strukturen, veranschaulicht. Eine Programmierung des kompletten Systems war im Rahmen meiner Diplomarbeit zeitlich nicht möglich.

## 2 Die Grundlagen von WCMSs

### 2.1 Systemarten

Lange vor dem Internet- und Web-Content-Management-System-Hype mussten Hersteller, welche komplexere Internetauftritte erstellten, auf eigene Produkte ausweichen. Die auf dem Markt erhältliche Software konnte nur schlecht mit großen Informationsmengen umgehen. Durch die herstellerabhängige Entwicklung entstanden immer mehr Systeme, die für spezielle Einsatzgebiete konzipiert waren. Man erkennt viele Parallelen zwischen den einzelnen Softwarelösungen aber auch große Differenzen. Deshalb hat auch jedes bei unterschiedlichen Verwendungen seine Vor- und Nachteile. [vgl. Krüger/ Kopp 2002, S. 54]

#### ○ *Dokumentenmanagement*

Eine der ältesten Formen der Online-Verwaltung ist das Document Management System (DMS). Zu den Aufgaben eines DMS zählen die zentrale Verwaltung von komplexen Dokumenten und die Schaffung von Zugängen zu diesen. Außerdem verfügt es über Freigabeverfahren, Versionierung, Indexierung und anderen Funktionalitäten. Bei einem DMS gibt es keine Trennung von Inhalt und Layout und daher sind die Dokumente einschließlich Formatierungen, Grafiken u. Ä. abgelegt. Der Zugriff auf die Dokumente erfolgt meistens über einen proprietären Client. [vgl. Krüger/ Kopp 2002, S. 55]

#### ○ *Content Management*

Der Content-Management-Begriff steht für die Weiterentwicklungen des Dokumentenmanagement und ist mehr ein Oberbegriff für die einzelnen Teilsysteme. Das Grundprinzip ist aber bei allen Content Management Systemen (CMS) gleich. Und zwar die Aufteilung der Dokumente in ihre Inhaltsobjekte. Zu den Unterkategorien zählen unter anderem das Web Content Management (WCM) und das Media Asset Management (MAM). [vgl. Krüger/ Kopp 2002, S. 55]

#### ○ *Knowledge Management (KM)*

Die Aufgabe des Knowledge Managements (deutsch: Wissensmanagement) ist es, nicht nur Daten abzuspeichern, sondern diese auch in Wissen umzuwandeln. Daten sind kleine



unstrukturierte Teile von Informationen. Stellt man diese Teile in einen Kontext, entstehen daraus Informationen. Mit der Hinzunahme von Metadaten, also Informationen über den Inhalt und die Beschaffenheit, sprechen wir von Wissen. Normalerweise existieren keine Freigabeverfahren. Somit kann das Knowledge Management System (KMS) jederzeit Wissen vermitteln und aufnehmen. Aber es besteht auch die Gefahr, dass das eingetragene Wissen nicht korrekt ist. Ein KMS eignet sich für Intranets, in denen Wissen und Informationen unter Mitarbeitern ausgetauscht werden. [vgl. Krüger/ Kopp 2002, S. 56; Boiko 2001, S. 6]

○ *Portale*

Ein Portal ist eine Erweiterung von Web Content Management Systemen (WCMS). Es nutzt ebenfalls die Zugriffsrechte und Gruppen- und Rollenzuordnung des jeweiligen Benutzers. Zusätzlich kann der Benutzer seinen „Arbeitsplatz“ selbst gestalten. Das heißt, dass er sich nur Daten anzeigen lässt, die für ihn wichtig erscheinen. Meist besteht auch die Möglichkeit der Einbindung kleinerer Applikationen (Tracks bzw. Module) in sein Portal, welche zum Beispiel eine tägliche Datenbankabfrage ausführen. [vgl. Krüger/ Kopp 2002, S. 59]

## **2.2 Evolution und Gliederung**

Durch die Entwicklung des Internets und die damit verbundene Nutzung als Informationsplattform wurden Wege für eine effektive Verwaltung der ständig wachsenden Informationsmengen gesucht. Doch auch die hohen Anforderungen an Aktualität, Qualität und Effizienz trugen zur Entstehung des Web Content Managements bei. In der Entwicklung gab es viele Versionen und Auffassungen, wie ein WCMS auszusehen hat. Es wurden immer bessere Möglichkeiten gesucht den Inhalt (englisch: content) zu verwalten und zu nutzen. Diesen Weg kann man grob in drei Generationen untergliedern. [vgl. Zschau et al. 2002, S. 73]

Die erste Generation entstand aus den Bedürfnissen der Nutzer heraus. Anfallenden Informationsmengen sollten effektiv verwaltet bzw. verarbeitet werden. Es wurden zwei Möglichkeiten geschaffen. Entweder konnte man HTML-Seiten dynamisch aus Datenbanken erzeugen oder man benutzte ein Dokumentenmanagement für diese Seiten.

Das Management dieser Seiten übernahmen zwar immer noch Experten, aber es war einfacher zu handhaben. Es gab keine interne Zugriffsverwaltung und nur einen manuellen Workflow. [vgl. Zschau et al. 2002, S. 73]

Zu der zweiten Generation zählt man Seiten, die den Nutzern durch viel Automation die Arbeit erleichterten. Dazu gehören die automatische Navigation und Publikation des Contents der Seite. Mittels eines festen Workflows wurden die Arbeitsabläufe kontrolliert und gesteuert. Ein wichtiges Merkmal dieser Generation war nicht nur, dass man Metainformationen zu Web-Objekten zuordnen konnte, sondern auch die technische Umsetzung des Content Life Cycles (→ Kapitel 2.4 *Content Life Cycle*). Das „echte“ Content Management war damit geboren. Endlich gab es eine Benutzerverwaltung, welche die Trennung von Inhalt und Layout ermöglichte. Durch diesen Schritt konnten technisch nicht versierte Nutzer unter anderem die Struktur der Seite verändern. [vgl. Zschau et al. 2002, S. 74]

In der derzeitigen dritten Generation wird das WCMS zu einem Baustein. Schnittstellen ermöglichen eine Integration in fast jede IT-Umgebung. Auch lassen sich viele Funktionen jetzt frei definieren. Dazu gehören unter anderem der Workflow und die Metainformationen. Ebenso ist eine Individualisierung des Contents für die einzelnen Benutzer und Benutzergruppen (Personalisierung) möglich. [vgl. Zschau et al. 2002, S. 74]

Die dritte wird letztendlich nicht die letzte Generation gewesen sein. Vor allem wird die Personalisierung ein wichtiger Punkt in der nächsten Zeit bleiben. Eine inhaltliche Anpassung an die Bedürfnisse des Benutzers steigert das Interesse an einer Webseite. Auch die Automatisierung wird ein großer Faktor für die zukünftigen Systeme sein. Durch Checking-Programme werden inhaltlich, logisch oder thematisch falsche Verlinkungen erkannt und verhindert.

Aber auch in Zukunft werden Dokumente nicht nur online angeboten. Offline-Formulare, wie zum Beispiel Quittungen, Listen und ähnliches, welche online verwaltet werden können, um sie jederzeit zu drucken oder zu versenden werden immer eine große Rolle spielen. Dabei wird sehr wahrscheinlich das Dokumentenmanagement in das WCMS integriert. Somit wird das DMS für die Offline-Prozesse und die Versionierung zuständig sein und das WCMS für alle Online-Prozesse.

## 2.3 Der Versuch einer Definition

Es existiert keine offizielle Definition für ein Web Content Management System. Jeder Anbieter hat seine eigene Auslegung. Daher werde ich einige grundlegende Fakten aufzeigen, welche ein WCMS ausmachen.

Ein Web Content Management System ist ein System, mit dem man Inhalte verwalten kann. Um dieses programmiertechnisch zu realisieren, sind eine Trennung von Inhalt, Gestaltung und Funktionalität einer Webseite und die Verwaltung dieser Teile notwendig. Man nennt dieses auch das Grundprinzip eines Web Content Management Systems. [vgl. Krüger/ Kopp 2002, S. 22] Abbildung 2-1 zeigt das Agieren der einzelnen Komponenten, das letztendlich die Darstellung der fertigen Webseite ermöglicht.

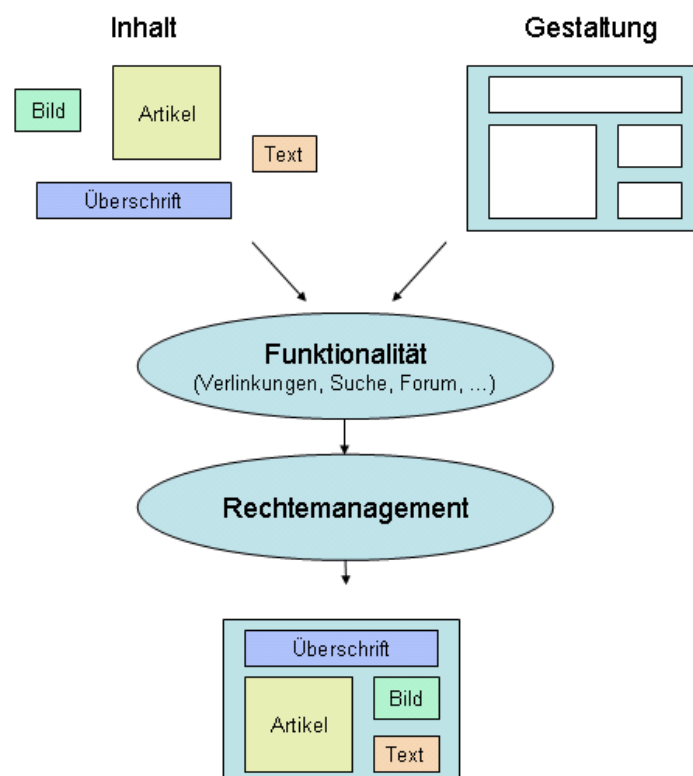


Abbildung 2-1 Grundprinzip eines Web Content Management Systems

Mit dem Wort **Inhalt** (englisch: content) werden alle digitalen Assets einer Webseite zusammengefasst, welche der Endnutzer sieht. Digitale Assets sind kleine Einheiten bzw.

Bausteine. Sie werden in derzeit fünf Asset-Arten untergliedert. [vgl. Zschau et al. 2002, S. 40]

- Die üblichen Webinhalte, welche Texte und Bilder einer Seite sind. Damit sind also die statischen Inhalte einer Seite gemeint.
- Die multimedialen Assets, zu denen Audiodaten, Videos und Animationen zählen.
- Die applikationsgebundenen Assets. Dazu gehören Dokumente, die ein separates Programm zur Darstellung benötigen. Zum Beispiel die PDF-Dokumente von Adobe<sup>1</sup> oder Tabellen, die mit Microsoft<sup>2</sup> Excel erstellt wurden.
- Transaktionale Assets sind für die Informationsweitergabe verantwortlich. Sie beinhalten die Daten, die man für die eindeutige Identifikation des aktuellen Nutzers braucht und weitere Daten, welche man z. B. für eine Bestellung benötigt.
- Die Community-Assets sind Inhalte von Chats und Foren, die der Nutzer eingegeben hat. Diese Daten sind zwar bei der Erstellung noch nicht verfügbar, aber sie müssen zu einem späteren Zeitpunkt auch verarbeitet werden.

Die **Gestaltung** (englisch: layout) einer Webseite wird durch so genannte Vorlagen oder auch Templates bestimmt. Sie sind für die gesamte Darstellung sowie die Anordnung des Contents verantwortlich und werden dadurch nicht von einer Inhalts- oder Funktionalitätsänderung beeinflusst. Durch dieses Gerüst von Platzhaltern ist die Einhaltung des Corporate Designs (CD) und der Corporate Identity (CI) gewährleistet. [vgl. Krüger/ Kopp 2002, S. 23]

Damit man mit einer Webseite arbeiten kann, braucht man **Funktionalität**. Dazu gehören nicht nur die Grundfunktionen wie Formulare und die Suche, sondern auch komplexere Funktionen wie z. B. ein Chat. Durch die Funktionalität kann man mit einer Seite interagieren.

Um die einzelnen Abschnitte dieser Trennung zu verwalten, werden Zugriffsrechte verteilt. Das **Rechtmanagement** teilt die einzelnen Benutzer in Gruppen und Rollen ein und gibt diesen bestimmte Rechte. Wann jemand etwas machen muss, bestimmt der Workflow, der Regeln beinhaltet, die den Arbeitsablauf bei der Erstellung und Verarbeitung der Seite

---

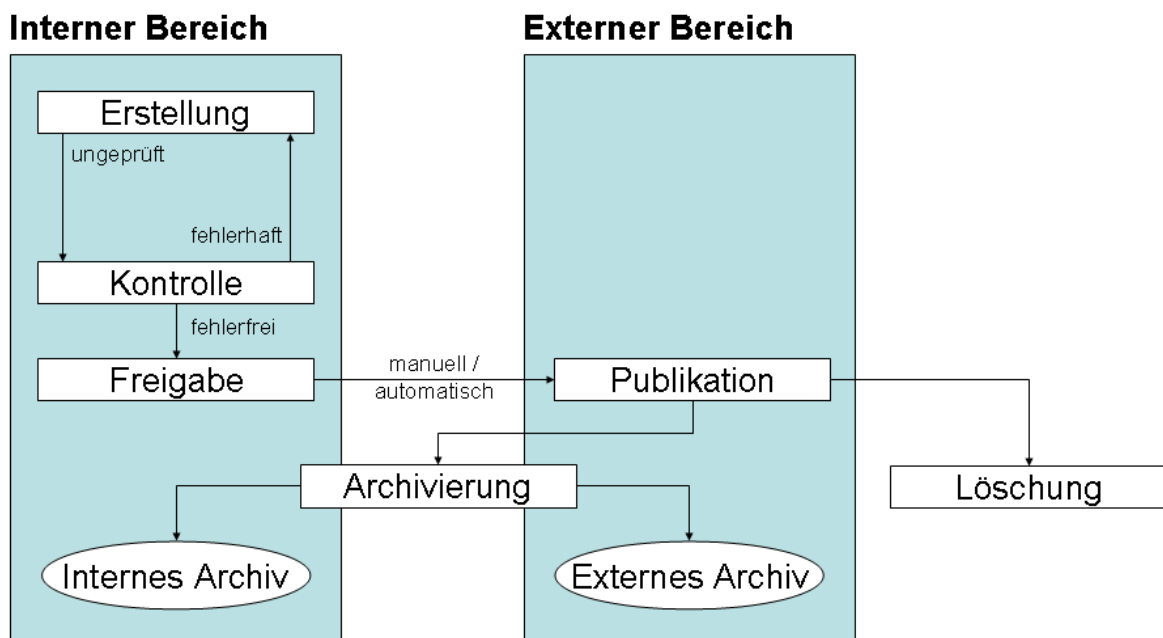
<sup>1</sup> <http://www.adobe.com/>

<sup>2</sup> <http://www.microsoft.de/>

steuern. Er wird direkt vom System umgesetzt und kontrolliert und nicht wie beim herkömmlichen Web-Publishing offline überwacht. Dabei wird der Content Life Cycle, auf den ich im nächsten Kapitel eingehe, automatisiert vollzogen. Durch das Rechtesystem und den Workflow ist es möglich, dass jeder Mitarbeiter in seinem zugeteilten Fähigkeitsbereich eingesetzt wird. Somit können die Autoren die Inhalte dezentral und Workflow-gesteuert erstellen und pflegen. Die Grafiker sind für die Gestaltung der einzelnen Templates zuständig und die Programmierer garantieren die Funktionalität der Seite. [vgl. Krüger/ Kopp 2002, S. 28]

## 2.4 Der Content Life Cycle

Der Content Life Cycle wird auch als Freigabezyklus oder Vier-Augen-Prinzip bezeichnet und ist der Lebensweg eines Inhaltes von der Idee bis zur Archivierung bzw. Löschung. Er ist bei einem Web Content Management System technisch umgesetzt. [vgl. Zschau et al. 2002, S. 77]



**Abbildung 2-2** Content Life Cycle [vgl. Zschau et al. 2002, S. 56]

Es fängt bei der Erstellung der digitalen Assets durch den Autor an. Mittels Programmen, wie zum Beispiel eines Texteditors, wird aus einer Idee ein Content. Fehler und Verstöße gegen bestimmte Rechte (z. B. Copyright) würden die Qualität der Seite stark mindern.

Eine Kontrolle ist daher unabdingbar. Fällt die Prüfung durch den Kontrolleur negativ aus, wird der Inhalt wieder bei dem Autor vorgelegt. Eine Verfälschung der Inhalte durch Dritte ist somit ausgeschlossen. Bei Korrektheit wird der Content freigegeben. Durch die Freigabe wird der Inhalt entweder sofort publiziert und damit für alle öffentlich sichtbar oder er wird erst zu einem bestimmten Termin (zeitgesteuert) veröffentlicht. Nachdem der Inhalt nicht mehr aktuell ist, wird er gelöscht oder archiviert. Der Inhalt kann in ein internes Archiv gespeichert werden oder den Nutzern weiterhin zur Verfügung stehen. Zu den Vorteilen der Archivierung zählen das schnelle Rollback, die Wiederverwendung und die Recherchemöglichkeit. Sie wird daher dem Löschen bevorzugt. [vgl. Zschau et al. 2002, S. 54]

## 2.5 Vor- und Nachteile

Durch das Grundprinzip eines WCMS ist eine effektive Nutzung der Kernkompetenzen aller Mitarbeiter möglich. Das bedeutet auch eine höhere Motivation, da dadurch eine größere Aufgabenteilung erfolgt. Zum Beispiel erstellen Redakteure ihren Inhalt und müssen sich nicht mit deren Darstellung beschäftigen.

Doch das ist nicht der einzige Vorteil einer Trennung. Sie ermöglicht auch eine Wieder- bzw. Mehrfachverwendbarkeit der Inhalte (englisch: reuse). Was schon einmal erstellt wurde, muss man nicht für jeden Nutzungskontext wiederholt machen. Die Voraussetzung dafür ist die medienneutrale Datenstrukturierung. Das bedeutet, dass jede Überschrift, jede Beschreibung, jeder Link usw. separat und nur als reine Daten (keine Formatierungen) abgespeichert werden. In diesem Zusammenhang sagt Knobloch: „Wer medienneutrale Datenmodellierung ernst meint, ist gezwungen, Informationseinheiten in Atome (oder mindestens Moleküle) zu fragmentieren oder Substrukturen einer Datenansammlung adressierbar zu machen.“ [Knobloch 2003, S. 34]

Dadurch, dass jedes Inhaltselement unformatiert im Content Repository gespeichert ist, ist das Cross-Media-Publishing, also die Konvertierung in verschiedene Ausgabeformate kein Problem. Wie in einem Baukastensystem können die einzelnen modularisierten Contentbauteile zu einem Output zusammengesetzt werden oder stehen problemlos zur Wieder- und Weiterverwendung zur Verfügung. [vgl. Wilhelm 2002] Der Output wird

mittels der verschiedenen Sprachen in die jeweiligen gewünschten Ausgabearten konvertiert. Dabei wird die Generierung von XML-Dateien immer wichtiger, weil es ein medienneutrales Format ist. Zu den XML-basierten Markup-Sprachen gehört zum Beispiel die Wireless Markup Language (WML). Sie wurde speziell für mobile Endgeräte entwickelt. Auch Ausgaben als PDF-Dateien (Portable Document Format), SMS (Short Message Service) und CD-ROM sind möglich. Doch auch Offline-Medien, wie zum Beispiel die Ausgabe auf Papier, sind kein Problem. [vgl. Zschau et al. 2002, S. 203]

Mittels eines WCMS ist es möglich, zu jedem Inhaltselement Metainformationen zu speichern. Damit ist zum Beispiel eine Klassifizierung, eine Zuordnung zu einem Autor oder die Verwaltung des jeweiligen Publikationszeitraumes möglich. Diese Möglichkeit ist unabdingbar für ein effizientes Informationsmanagement. Es trägt nämlich zur Automatisierung der Webseite bei. [vgl. Zschau et al. 2002, S. 77]

Die Templateprogrammierung erleichtert erheblich die Erstellung neuer Seiten. Es müssen beispielsweise nicht bei jeder neuen Seite Tabellen angelegt und ausgerichtet werden. Außerdem ermöglichen Templates eine Designveränderung (englisch: redesign) der ganzen Seite in wenigen Schritten unter Beibehaltung des Corporate Designs. Dadurch spart man Arbeitsaufwand und somit auch Geld.

Der integrierte Freigabezyklus in einem WCMS garantiert die Qualität des Inhaltes. Nach dem Prinzip „Vier Augen sehen mehr als zwei“ wird ein Inhalt nicht direkt nach der Erstellung publiziert, sondern zuerst von mindestens einem Mitarbeiter kontrolliert. Damit können nicht nur Schreibfehler, sondern auch Verstöße gegen rechtliche Grundlagen verhindert werden. Denn viele Mitarbeiter wissen nicht, dass zum Beispiel nicht jedes Bild publiziert werden darf. Eine qualitätsgesicherte Webseite erhöht die Präsenz im Internet enorm. Ein weiterer Vorteil des Freigabezyklus ist die Verkürzung der Zeit, die der Inhalt bis zu seiner Veröffentlichung braucht, auch Time-to-Web genannt. Ebenso die technische Abbildung des Workflows bringt Vorteile. Sie bietet eine konsistente Kontrolle der Abläufe und Aufgaben. Die Teilaufgaben gelangen direkt an die zuständigen Mitarbeiter. Eine gleichzeitige Bearbeitung desselben Inhaltes durch verschiedene Mitarbeiter wird durch Versionierung und Check-in/ Check-out-Verfahren ausgeschlossen. Nach der Bearbeitung aller Aufgaben kann der Inhalt automatisch durch das WCMS publiziert

werden. Diese Verfahren gewährleisten eine hohe Aktualität und wiederum eine optimale Personalbelastung. [vgl. Zschau et al. 2002, S. 76 f.]

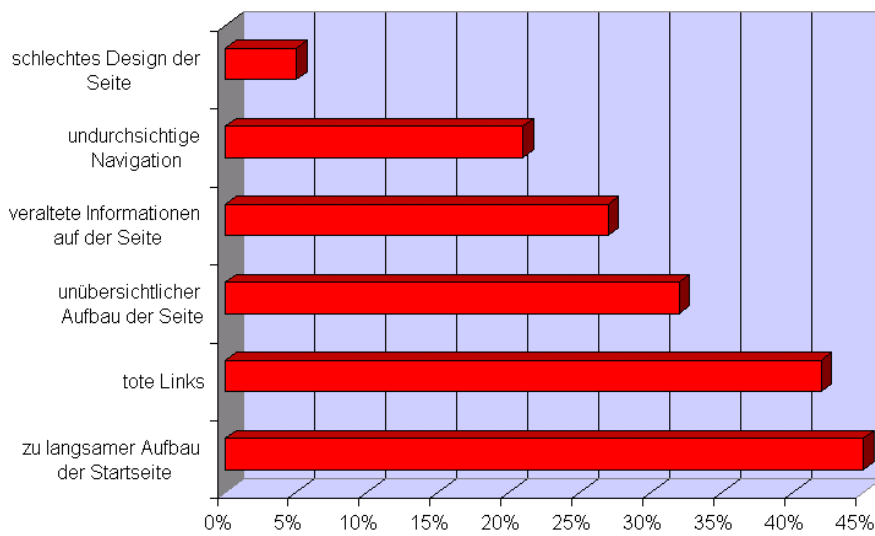
Durch Authoring-Möglichkeiten werden immer wiederkehrende Aufgaben durch das System erledigt. Dadurch werden nicht nur die Benutzer entlastet, sondern es wird auch Aktualität und Funktionalität gewährleistet. Eine Sitemap braucht zum Beispiel nicht mehr gewartet zu werden, da sie vom System generiert wird. Die Links brauchen nicht auf Konsistenz geprüft werden. Funktionen übernehmen diese Aufgabe und geben zum Beispiel einen Hinweis aus, dass die Seite derzeit nicht zur Verfügung steht oder löschen diese Verknüpfung. Im Laufe der Zeit kann man durch diese Automatisierungen Personalkosten einsparen, die man sonst für die Systemwartung ausgegeben hätte bzw. können die Mitarbeiter sich auf wichtigere Aufgaben konzentrieren. [vgl. Zschau et al. 2002, S. 77]

Die dezentrale Bearbeitung der Inhalte und das zentrale Content Repository sind weitere Kriterien, die für ein Web Content Management sprechen. Jeder Autor kann mittels eines Internetzugangs von jedem Ort auf der Welt aus seinen Inhalt erstellen und bearbeiten. Dieser wird zentral gespeichert und steht somit auch anderen Mitarbeitern zur Verfügung. Diese beiden Fakten bedeuten nicht nur, dass jeder Standort in den Publishingprozess eingebunden werden kann, sondern auch eine Weiterverwendung vorhandener Inhalte und eine hohe Aktualität, da neue Inhalte jederzeit erstellt und veröffentlicht werden können. [vgl. Zschau et al. 2002, S. 78]

Ein WCMS besitzt Schnittstellen und kann somit erweitert werden und als Erweiterung in bestehenden IT-Strukturen genutzt werden. [vgl. Zschau et al. 2002, S. 78]

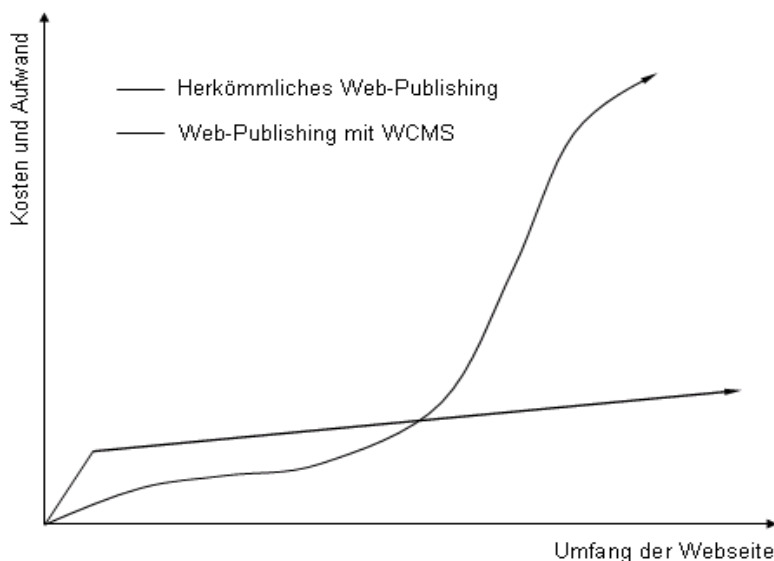
In Abbildung 2-3 wird verdeutlicht, welche Ursachen für Anwender existieren, eine Webseite nicht mehr zu besuchen (Mehrfachnennungen waren möglich). Wie schon angesprochen werden diese Probleme durch ein WCMS verringert bzw. ganz beseitigt.





**Abbildung 2-3** Ursachen für Webseitenmeidung [Versteegen 2001, S. 20]

Trotz der vielen Vorteile gibt es bei einem WCMS auch Nachteile. Die beiden größten, welche viele Firmen von der Einführung eines WCMS abbringen, sind der initiale Aufwand für die Implementierung und die sehr hohen Anfangskosten. Besonders die Migration der bestehenden Webseiten erweist sich als große Hürde bei der Entscheidung für ein WCMS. [vgl. Weber 2002] Die folgende Abbildung zeigt aber, dass sich diese Nachteile mit der Zeit zum Vorteil wandeln. Denn durch die Automatisierung von Routinearbeiten, die Templateverwendung, den technisch umgesetzten Workflow etc. werden mit der Zeit die Arbeiten an der Webseite stark erleichtert. [vgl. Zschau et al. 2002, S. 78]



**Abbildung 2-4** Kostenentwicklung einer Webseite [vgl. Zschau et al. 2002, S. 79]

Die Webseiten eines WCMS werden meist erst bei Aufruf dynamisch generiert. Die Benutzer bekommen dadurch zwar immer die aktuellste Seite, aber die Indizierung durch Suchmaschinen wird erschwert bzw. ganz verhindert. Durch Spider, welche die Suchmaschinen benutzen, werden die URLs untersucht. Bei einer URL-Syntax, die auf eine dynamische Generierung einer Seite hinweist, umgehen die meisten Suchmaschinen diese. Ein Informations-Paradoxon, das durch eine Umgestaltung der URL-Parameter umgangen werden kann. [vgl. Zschau et al. 2002, S. 293]

Weitere Nachteile entstehen meist durch die Anwender selbst. Bei einer schlechten Planung und Umsetzung folgt unabdingbar eine Verschlechterung des Systems. Dabei kann es vorkommen, dass die Aktualität nicht mehr gewährleistet werden kann, die Funktionalität nicht ausgeschöpft wird und Performanceprobleme auftreten. Deshalb sollte man auch viel Zeit für die Konzeptionsphasen (→ Kapitel 5 *Vorgehensweise bei der Erstellung eines WCMS*) aufbringen.

### 3 Komponenten und Aufbau eines WCMSs

Ein Web Content Management System besteht immer aus fünf Komponenten: das Assetmanagement, die Workflowkomponente, die Zugriffs- und Benutzerverwaltung, die Im- und Exportschnittstellen für den Content und die Programmierschnittstellen. [vgl. Zschau et al. 2002, S. 201] Die Bestandteile sind in Ebenen untergliedert und in Abbildung 3-1 veranschaulicht.



**Abbildung 3-1** Komponenten eines WCMS

Die 1. Ebene enthält die wichtigsten Bestandteile: Das Assetmanagement mit den Funktionen Verwaltung, Strukturierung und Darstellung des Contents und die Workflowkomponente als Verbindung der Assets mit dem Workflow. In der 2. Ebene werden die Zugriffe auf das WCMS gesteuert und die Rechte der einzelnen Benutzer verwaltet. Die 3. Ebene ist für die Verbindung und das Austauschen von Informationen mit externen Quellen zuständig.

Im Folgenden werden alle dazugehörigen Komponenten näher erläutert.

## 3.1 Assetmanagement

Um den Inhalt einer Webseite zu verwalten, benötigt man ein Assetmanagement. Wie der Name Content Management System schon sagt, ist dies die wichtigste Komponente eines WCMS. Das Assetmanagement soll aber nicht nur den Inhalt verwalten, sondern ihn auch strukturieren und darstellen. Denn bei einem WCMS ist der Inhalt von der Darstellung getrennt und muss zu einer Webseite zusammengeführt werden. Genauer gesagt, wird der Inhalt, der aus vielen digitalen Assets besteht, in die vorgefertigten Vorlagen (englisch: templates) eingebettet und danach dargestellt. [vgl. Zschau et al. 2002, S. 202]

### 3.1.1 Abspeicherung des Inhaltes

Die Abspeicherung des Inhaltes erfolgt zentral im Content Repository. Das hat den Vorteil, dass die Systemverwaltung und die Sicherung der Daten stark vereinfacht werden. Es existieren zwei Grundmodelle, den Inhalt zu speichern.

#### ○ *Dateien und Verzeichnisse*

Dateien haben nicht nur eine klare Adressierung über eine URL-Adresse, sondern sind einfach zu erstellen und zu verwalten, da viele Editoren auf dem Markt existieren. Dadurch, dass die Dateien statisch sind, hat man eine sehr gute Performance. Die Seiten müssen nämlich bei einem Abruf nicht neu generiert werden, sondern können direkt geladen werden. Außerdem können Dateien auch Programmcode enthalten. Demzufolge können Scripte oder sogar komplette Programme im Dateisystem abgespeichert sein. Die Abspeicherung in Dateien hat nicht nur Vorteile. Die Inhalte einer Datei sind meist unstrukturiert und können auch schlecht strukturiert werden. Daher ist es auch nicht möglich, direkt auf eine Information zuzugreifen oder eine effektive Suche zu verwenden. [vgl. Zschau et al. 2002, S. 204]

#### ○ *Datenbanken (objektorientierte, relationale)*

Im Gegensatz zum Dateisystem wird der Content in Datenbanken strukturiert abgespeichert. Somit sind Wiederverwendbarkeiten und eine optimale Suchen gewährleistet. Um einen Inhalt zu bearbeiten, reicht ein Texteditor nicht mehr aus. Für einen Datenbankzugriff braucht man spezielle Applikationen oder Webformulare. Die

aufgerufenen Seiten werden dynamisch generiert, was die Performance stark verringert. Deshalb wurden Cacheverfahren entwickelt, welche die Webseiten bei einer Inhaltsaktualisierung temporär statisch generieren. [vgl. Zschau et al. 2002, S. 206]

○ *Hybride Architektur*

Die Hybride Architektur ist eine Mischform beider vorher genannten Modelle. Sie vereinigt die Vorteile beider Ansätze und wird somit am meisten genutzt. Dabei dient das Dateisystem zur Verwaltung der unstrukturierten bzw. wenig strukturierten Informationen wie zum Beispiel Grafiken, Audio- und Videodateien und Scripte, welche auf dem Webserver ausgeführt werden. Datenbanken hingegen werden zur Verwaltung von Textassets und Metainformationen der Assets benutzt. [vgl. Zschau et al. 2002, S. 208] „Dieser hybride Ansatz hilft, eine flexible und offene Gestaltung beizubehalten und jeweils dort zu dynamisieren und zu strukturieren, wo es Sinn macht.“ [Zschau et al. 2002, S. 208]

### **3.1.2 Bearbeiten des Inhaltes**

Es existieren zwei Möglichkeiten, den Content einer Seite zu editieren. Die erste Variante ist die Bearbeitung mittels eines *Browser-Clients*. Der Nutzer kann seinen herkömmlichen Browser benutzen und muss sich nicht an ein neues Editierwerkzeug gewöhnen. Nach der Anmeldung bekommt er im System seine Aufgaben und Eingabeformulare aufgelistet und kann diese dann abarbeiten. Browser werden zu jedem Betriebssystem standardmäßig mitgeliefert, daher ist die Bearbeitung der Inhalte von jedem Rechner mit Netzzugang aus möglich. Diese Möglichkeit nennt man auch dezentralisierte Bearbeitung. In der zweiten Variante kann man den Inhalt mit einem *anwendungsspezifischen Client* bearbeiten. Mit ihm ist es möglich, dass alle Funktionen des WCMS auf einer Oberfläche vereint sind. Nachteilig ist die arbeitsplatzabhängige Installation, da es sich um eine spezifische Software handelt, die nicht mit jedem Rechner mitgeliefert wird. Damit steigt der Aufwand für den Administrator, weil er diese nicht nur installieren, sondern auch einrichten, erklären und warten muss. Damit ist die Bearbeitung des Inhaltes ortsgebunden in Bezug auf den Administrator. [vgl. Zschau et al. 2002, S. 216]

### 3.1.3 Automatisierungen

Content-Authoring umfasst alle Verwendungsmöglichkeiten von strukturiertem Content, welche automatisch Seiten oder einzelne Objekte aus intern und extern abgespeichertem Inhalt generieren. Das Prüfen fehlerhafter interner und externer Links (englisch: Link-Checking) gehört heutzutage zum Standard eines jeden WCMS. Es ist zwar kein originäres Content Authoring, aber eine Automatisierung. Dabei kann das System im Fehlerfall den Link löschen bzw. eine entsprechende Nachricht an den Nutzer ausgeben und den Verantwortlichen für diesen Bereich informieren.

Bei Veränderungen des Inhaltes einer Seite möchte und sollte der Nutzer von diesen erfahren. Mittels der Metainformationen eines Objektes weiß das WCMS, wann ein Asset verändert wurde und kann daher eine Seite mit den neuesten Veränderungen generieren.

Um sich auf einer Seite zurechtzufinden, benötigt man eine Navigation. Dieses Element kann ein WCMS aus der vorhandenen Struktur der Daten automatisch erstellen und ist somit immer aktuell.

Was das Inhaltsverzeichnis eines Buches ist, ist die Sitemap für eine Webseite. Sie verschafft den Nutzern eine komplette Übersicht über die Seite. Doch eine Sitemap ist durch die Menge an Informationen und den sich ständig ändernden Inhalt nur schwer manuell erstellbar. Genau wie bei der Navigation wird durch das WCMS diese Seite mithilfe der im System abgelegten Datenstruktur generiert.

Damit eine Seite effektiver genutzt wird, werden immer häufiger Related Links eingesetzt. Durch diese themenbezogenen Links wird der Nutzer angeregt, durch themenverwandte Rubriken zu browsen. Mittels Metainformationen werden zu jedem Objekt die zugehörigen Themenbereiche festgelegt, sodass sie in Beziehung mit anderen Objekten stehen. In diesem Zusammenhang spricht man schon von Personalisierung. Durch die Abspeicherung der Interessen und personalisierten Daten des Nutzers in Profile, wird das Angebot durch das WCMS individuell an die jeweilige Person angepasst. [vgl. Zschau et al. 2002, S. 209]

### **3.1.4 Versionierung**

Aufgrund der hohen Aktualität einer Webseite verändert sich deren Inhalt stetig. Bei einer Veränderung eines bzw. mehrerer Assets wird vorher vom WCMS eine Kopie abgespeichert. Dieses Verfahren nennt man Versionierung. Die Vorteile dieser Methode liegen klar auf der Hand. Falls die neuen Assets den Ansprüchen nicht gerecht werden, wird der Änderungsvorgang wieder rückgängig gemacht. Man kann diese Kopie auch zu einem späteren Zeitpunkt wiederverwenden und nachvollziehen, wann welche Assets online waren. [vgl. Zschau et al. 2002, S. 215] Wenn sich die Inhalte öfter ändern, sollte die Versionierung zeitabhängig gestalten werden, damit sich nicht zu viele Daten ansammeln.

### **3.1.5 Länder- und Sprachabhängigkeit**

Heutige Webseiten existieren nicht nur in der Sprache der Autoren, sondern sind auch in der Sprache der Nutzer verfasst. Eine Untersuchung der Gartner Group zu Folge zeigt, dass 75% der Internet-User bei einer muttersprachenfremden Webseite nach einer Übersetzung suchen oder die Seite verlassen. [vgl. Schwarze 2001, S. 52] Bei der Umsetzung muss aber einiges beachtet werden. Natürlich wäre es schön, wenn man jedes einzelne Wort einfach nur übersetzen müsste. So leicht ist es aber nicht. Der sprachliche Zusammenhang würde nicht mehr stimmen. Auch müssen kulturelle Unterschiede, wie Maßeinheiten, Datumsformate, Währungen und weitere Formate abgeändert bzw. umgerechnet werden. Auch das in Deutschland verlangte Impressum ist in vielen anderen Ländern nicht erforderlich. Diese sprachlichen und landesabhängigen Veränderungen sollten aber nicht nur auf den publizierten Webseiten angewandt werden, sondern auch für die Editoren und Benutzeroberflächen der Bearbeiter. Durch die sprachliche Anpassung der Clients wird die Benutzung vereinfacht und man erhält damit eine höhere Effektivität und Akzeptanz der Anwendung. Eine Vielzahl von unterschiedlichen Sprachen erfordert eine Anpassung des Workflows. Daher müssen die sprachlich abhängigen Assets in allen bereitgestellten Sprachen verfügbar sein. Dies bedeutet mehr Aufwand für die Mitarbeiter.

## **3.2 Die Workflowkomponente**

Ebenfalls zu den wichtigsten Komponenten des WCMS gehört die Workflowkomponente. Für jede Komponente (Assets, Templates usw.) einer WCMS-Webseite kann ein separater Workflow erstellt werden. Über die Workflowkomponente werden die Arbeitsschritte und deren Abfolge von jedem Mitarbeiter für diese Komponenten festgelegt, automatisiert und überwacht. Erst durch diese Komponente ist eine dezentrale und aufgabenbezogene Arbeit in größeren Teams möglich. [vgl. Zschau et al. 2002, S. 223]

### **3.2.1 Workflowarten**

Wenn ein Workflow direkt einem Inhaltselement zugeordnet ist, wird dies als objektorientierter Workflow bezeichnet. Dabei wird zu jedem Objekt festgelegt, wer der Autor ist und wer es freigeben darf. Eine andere Art des Workflows sind die frei definierbaren Workflows. Sie werden objektunabhängig festgelegt. Objekte werden dann den jeweiligen Abläufen zugeordnet. [vgl. Zschau et al. 2002, S. 229]

### **3.2.2 Das Rechtekonzept**

Damit jeder Nutzer nach seinen Aufgaben und Fähigkeiten eingesetzt werden kann, benötigt man eine Rechteverwaltung. Dafür muss dem System jeder Benutzer bekannt sein. Bei wenigen erfassten Personen bietet sich das Rechtekonzept auf Nutzerebene an. Dabei werden Sammlungen von Rechten (Rollen) den einzelnen Personen direkt zugeteilt. Mit wachsender Personenzahl erhöht sich auch der administrative Aufwand. Ein weiterer Nachteil ist, dass der Workflow nur direkt einzelne Personen adressieren kann. Bei eventuellem Ausfall einer Person, kann der ganze Arbeitsablauf zum Stillstand kommen. Eine bessere Lösung ist das Gruppenprinzip. Dabei werden die einzelnen Benutzer in Gruppen und meistens auch Untergruppen eingeteilt. Über die Gruppen wird der Zuständigkeitsbereich eines jeden Benutzers bestimmt. Jeder Gruppe werden Rollen zugeordnet. Die Rollen bestimmen die Aufgaben und Rechte des Nutzers. Eine Rolle kann ebenfalls in verschiedenen Gruppen existieren. Dieses Prinzip erlaubt es, dass neuen Nutzern nicht jedes Recht einzeln zugeteilt werden muss, sondern nur die Zugehörigkeit zu



Gruppen. Somit ist eine gruppenbezogene und gleichzeitig rollenbezogene Aufgabenverteilung möglich. [vgl. Zschau et al. 2002, S. 223]

Das Rechtekonzept wird auch benutzt, um die Verantwortlichkeiten bei der Umsetzung des Content Life Cycles zuzuordnen. Somit übermittelt das System die anfallenden Arbeiten mit den zugehörigen Nachrichten an die im Content Life Cycle bestimmten Mitarbeiter. [vgl. Zschau et al. 2002, S. 227]

### **3.2.3 Geschütztes Editieren**

Wenn mehrere Mitarbeiter einen Content bearbeiten wollen, kommt es zu Kollisionen, welche einen Datenverlust verursachen können. Um dies zu verhindern, werden so genannte Check-in/ Check-out Routinen verwendet. Dabei wird der Inhalt nach dem Öffnen auf Check-out gesetzt und ist somit für alle anderen Nutzer gesperrt. Nachdem der Bearbeitungsvorgang beendet ist, wird der Content wieder freigegeben. [vgl. Zschau et al. 2002, S. 228]

### **3.2.4 Werkzeuge des Workflows**

#### *○ Benachrichtigungen*

Kommunikationen des WCMSs mit dem Nutzer erfolgen per E-Mail bzw. internem Messagesystem und über systeminterne Anzeigen.

#### *○ Statusanzeigen und Filter*

Um Zustände zu überprüfen, werden Statusanzeigen benutzt. Sie können auch genutzt werden, wenn keine Änderungen oder Aktionen stattgefunden haben. Beispielsweise sind Abfragen von Metainformationen der Assets möglich. Für komplexere Abfragen über das Content Repository und die Metainformationen werden Filter eingesetzt. Dazu gehört zum Beispiel eine Abfrage, in welchem Zeitraum ein Autor welche Assets bearbeitet hat.

#### *○ Aufgabenverteilung und -listen*

Projektplanungssoftware kann derzeit noch nicht mit einem WCMS kommunizieren. Darum werden alle Aufgaben über das WCMS geregelt. Der Vorteil davon ist, dass kein

Wechsel zwischen den Applikationen erfolgen muss, um festzustellen, welche Aufgaben noch zu erledigen sind und welche nicht.

○ *Protokollierung*

Es existieren zwei Arten der Protokollierung. Zum einen die objektbezogene Protokollierung. Sie erfasst alle Änderungen eines Objektes. Dazu gehört, wann welcher Mitarbeiter ein Objekt erstellt, gepflegt und gelöscht hat. Zum anderen gibt es die administrative Protokollierung. Dadurch werden die Zugriffe auf das System registriert.

○ *Previews*

Die Previewfunktionalität dient zur Visualisierung der neu erstellten Inhalte im Umfeld der Webseite. Eine Qualitätssicherung der Inhalte wird daher sehr vereinfacht. Diese Vorschau ist bei Betrachtung nicht auf der Live-Webseite zu sehen.

○ *Vertreterregeln*

Wenn ein Mitarbeiter fehlt oder nicht zur Verfügung steht, darf die Arbeit an einer Webseite nicht zum Stillstand kommen. Um das zu gewährleisten, kann man im Workflow zu jedem Nutzer einen Vertreter bestimmen. Siehe auch → Kapitel 3.2.5 *Freigabeverfahren*. [vgl. Zschau et al. 2002, S. 231-233]

### **3.2.5 Freigabeverfahren**

Das Freigabeverfahren ist unverzichtbar für ein gutes WCMS. Es garantiert die Qualität des publizierten Inhaltes. Jeder erstellte bzw. veränderte Inhalt wird einem oder mehreren Kontrolleuren gezeigt. Diese Person bzw. Personen entscheiden, ob der Inhalt veröffentlicht wird oder nicht. Es existieren zwei Verfahren der Freigabe: das lineare und das parallele Freigabeverfahren.

Bei der linearen Freigabe durchläuft der neue Inhalt der Reihe nach jede zugeordnete und freigabeberechtigte Person. Falls mindestens eine Person ausfällt, wird die Reihe unterbrochen und es kommt zu keiner Publikation. Darunter leidet die Aktualität der Seite. Daher hat man Verfahren entwickelt, die dieses Problem umgehen. Zum einen könnte man einen Stellvertreter bestimmen (Vertreterregel), welcher die Aufgaben der ausgefallenen

Person übernimmt. Eine andere Möglichkeit wäre, einen Freigabeberechtigten nach einer gewissen Zeit zu übergehen.

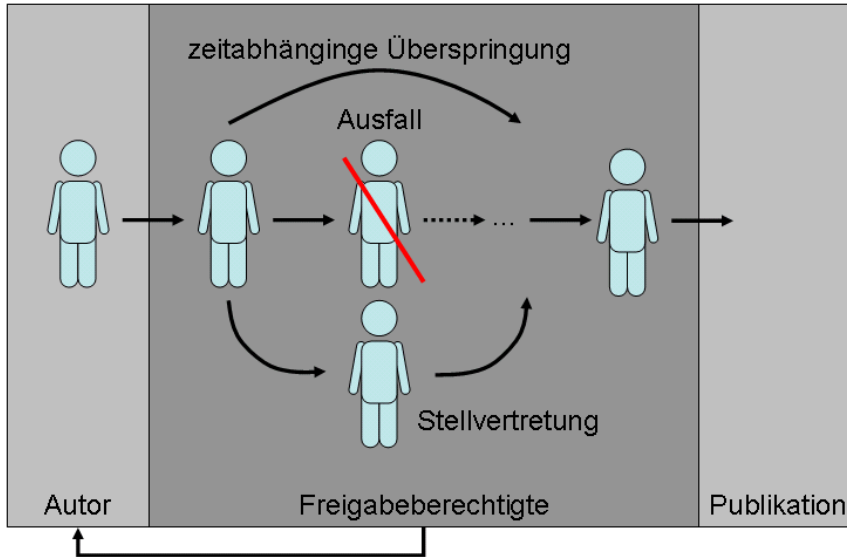


Abbildung 3-2 Lineares Freigabeverfahren

Eine flexiblere Variante ist das parallele Freigabeverfahren. Dabei werden verschiedene Wege definiert, welche zur Freigabeerfüllung notwendig sind. Zum Beispiel reicht bei Angabe von mehreren Freigabeberechtigten die Freigabe durch einen, um diese abzuschließen. [vgl. Krüger/ Kopp 2002, S. 44]

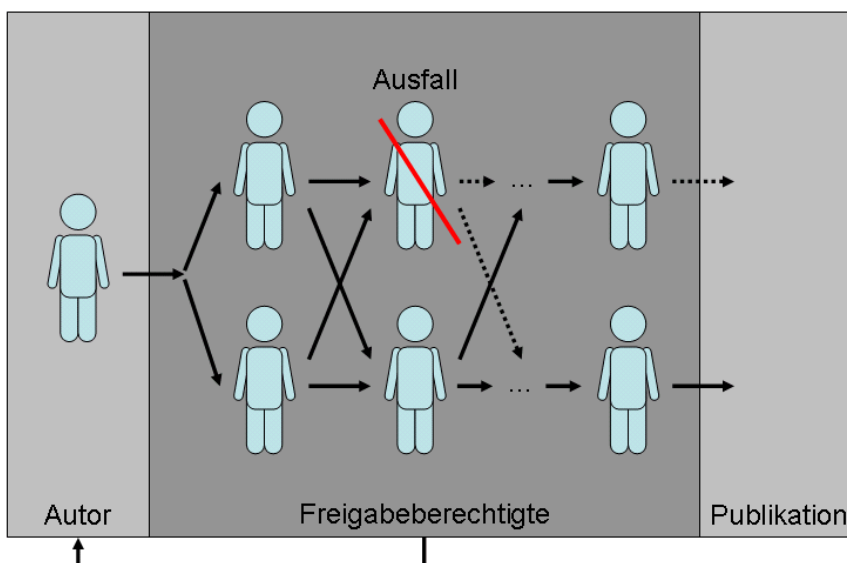


Abbildung 3-3 Paralleles Freigabeverfahren

### 3.3 Zugriffs- und Benutzerverwaltung

Nutzer sollen nur die Inhalte sehen, für die sie auch eine Berechtigung haben. Die Verwaltungsfunktionen des WCMSs regeln diese Rechte und schützen somit vor Fremd- bzw. Fehlzugriffen. Die Zugriffsverwaltung ist dafür verantwortlich, dass sich jeder Nutzer mit seiner Kennung und seinem Passwort authentifiziert. Sind diese Daten fehlerhaft oder nicht vorhanden, gilt der Nutzer als anonym. Ihm wird dann eine eingeschränkte Nutzung gewährleistet oder der Zugang vollständig verwehrt. Durch die Benutzerverwaltung sind dem System nach der Nutzerauthentifizierung die Berechtigungen der Person bekannt. Die Umsetzung der Sicht auf Funktionen und Inhalte kann auf zwei verschiedene Arten realisiert werden. Entweder erlaubt man dem Nutzer die Sicht auf alle Möglichkeiten und Inhalte, die das System bietet und seine Rechte werden bei jeder Aktion geprüft. Die bessere Variante ist die Anzeige von dem, was er auch nach seinen Rechten im System machen und sehen darf. Das erlaubt eine übersichtlichere Anordnung für den Nutzer und damit auch eine effektivere Arbeit. [vgl. Zschau et al. 2002, S. 235]

### 3.4 Im- und Exportschnittstellen für Content

Um mit externen Quellen zu kommunizieren bzw. zu agieren, benötigt man Schnittstellen. Mit ihnen ist es möglich, verschiedene digitale Dokumente zu importieren und gegebenenfalls umzuwandeln. Auch Abfragen auf Fremddatenbanken sind mittels ODBC/JDBC-Schnittstellen kein Problem für ein Web Content Management System. Der Import bestehender Webseiten ist ebenfalls möglich. Diese Methode wird aber in den meisten Fällen nicht angewandt, da die Webseiten oft keine Strukturierung aufweisen und keine Trennung von Inhalt und Darstellung besitzen. Somit ist der Import trotz verschiedener Möglichkeiten sehr schwierig. Bei einer vorhandenen Trennung können verschiedenste Arten von digitalen Assets importiert, exportiert und umgewandelt werden. Dies bedeutet einen Mehrwert der Webseite, da die unterschiedlichsten Ausgabeformate generiert werden können. Zum Beispiel das standardmäßige PDF-Dokument der Firma Adobe oder WML für WAP-fähige Handys bzw. PDAs. [vgl. Zschau et al. 2002, S. 235]

### 3.5 Programmierschnittstellen

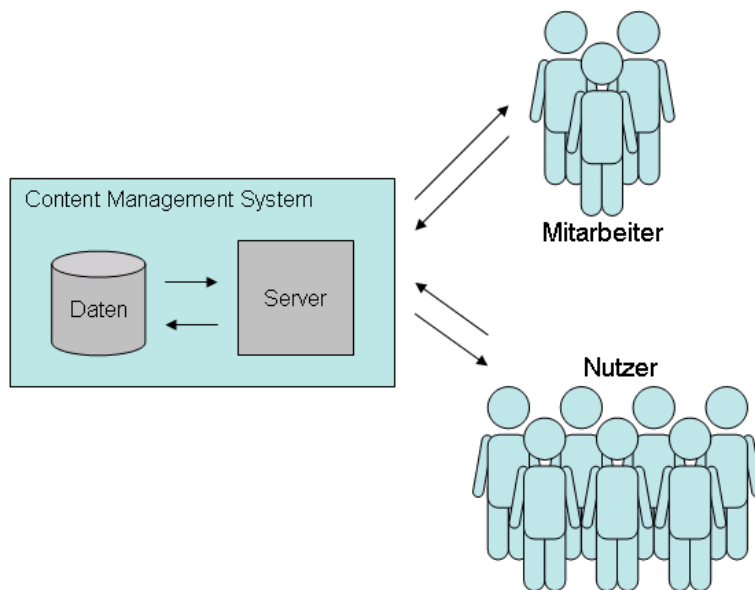
Eine Zusammenführung von mehreren Applikationen zu einer integrierten Web-Infrastruktur erfordert individuelle Schnittstellenprogrammierung. Ein weiterer unabdingbarer Einsatz der programmierbaren Schnittstellen besteht in der Weiterverwendung von bestehenden IT-Systemen und deren Integration in das WCMS. Ebenfalls für Erweiterungen des WCMSs mittels Scriptsprachen oder höherer Programmiersprachen werden sie benötigt. Diese Schnittstellen für funktionale Erweiterungen nennt man Application Programming Interface (API). Das sind Routinen und Objekte, mit denen eine Kommunikation mit anderen Applikationen und Erweiterungen erfolgen kann. [vgl. Zschau et al. 2002, S. 240]

### 3.6 Serverkonzepte

„Der Server eines WCMS enthält das Content Repository, verwaltet den Workflow, das Rechtekonzept und bietet Schnittstellen nach außen.“ [Zschau et al. 2002, S. 243] Durch die Zugriffe entstehen große Lasten auf dem Server, die mit den folgenden Serverkonzepten weitestgehend gering gehalten werden. Die Wahl eines geeigneten Konzeptes ist projektabhängig.

#### 3.6.1 Dynamic Publishing

Wie der Name schon sagt, werden hierbei nur dynamische Webseiten veröffentlicht. Das bedeutet, dass alle Informationen, die ein Nutzer abrufen, dynamisch zur Laufzeit generiert werden. Das gilt nicht nur für die potentiellen Nutzer des Systems, sondern auch für die Redakteure, Grafiker und Techniker. Wie man in Abbildung 3-4 sieht, erfolgen alle Zugriffe über ein und denselben Server. Bei einer hohen Zugriffsrate oder einem langsamen Server kann das leicht zu Performanceproblemen führen. Eine Abhilfe für das Problem bringt das Caching von hoch frequentierten Seiten. Durch die Zwischenspeicherung der Seiten werden die Antwortzeiten des WCMSs deutlich geringer. Genutzt wird dieses Konzept vor allem bei datenbankbasierten WCMS. [vgl. Zschau et al. 2002, S. 244]



**Abbildung 3-4** Prinzip Dynamic Publishing [vgl. Zschau et al. 2002, S. 244]

### 3.6.2 Publishing-/ Staging-Server

Bei diesem Prinzip gibt es einen internen und einen externen Server. In Abbildung 3-5 ist zu sehen, dass der interne Server (Publishing-Server) allein für die Arbeit der Mitarbeiter bereitgestellt wird. Nach der Freigabe werden die Seiten manuell oder zeitgesteuert vom Publishing-Server auf den Staging-Server übertragen. Bei der Übertragung werden statische Seiten generiert. Somit sind die Seiten auf dem externen Server schnell abrufbar und trotzdem durch dynamische Updates des WCMS aktuell. Weitere Vorteile, welche für das Publishing-/ Staging-Server-Prinzip sprechen, sind die Sicherung des internen Servers mittels einer Firewall und die Erhöhung der Betriebssicherheit durch die Trennung von Redaktions- und Webserver. Falls ein Server ausfällt, kann mit dem anderen weitergearbeitet werden. Der Mehraufwand für die Verwaltung von zwei Systemen ist gegenüber dem Mehrwert eher gering. [vgl. Zschau et al. 2002, S. 245]

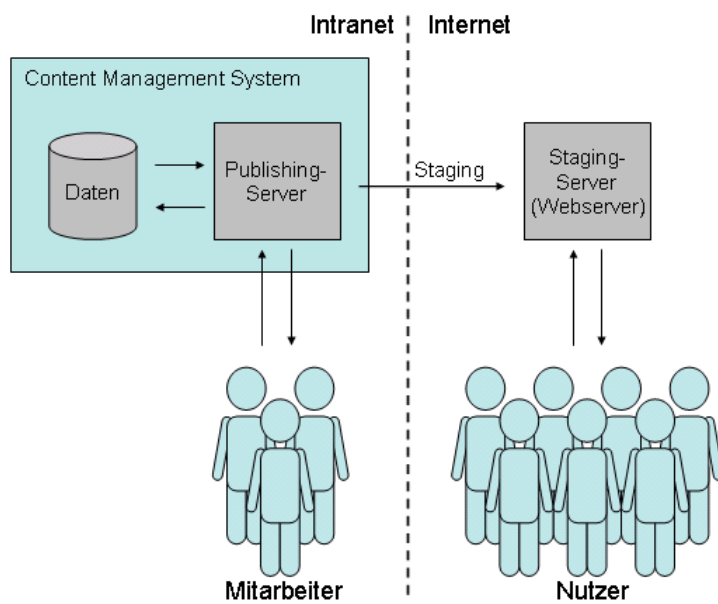


Abbildung 3-5 Prinzip Publishing-/ Staging-Server [vgl. Zschau et al. 2002, S. 246]

### 3.6.3 Publishing-/ QA-/ Staging-Server

Eine Weiterentwicklung des Publishing-/ Staging-Servers ist das Konzept des Publishing-/ QA-/ Staging-Servers. Der zusätzliche Qualitätssicherungsserver (englisch: Quality Assurance Server) im internen Bereich wird allein für die Kontrolle und Freigabe von Inhalten verwendet. Durch diese Verteilung des Workflows ist eine performante Arbeitsumgebung für viele Mitarbeiter gewährleistet. [vgl. Zschau et al. 2002, S. 247]

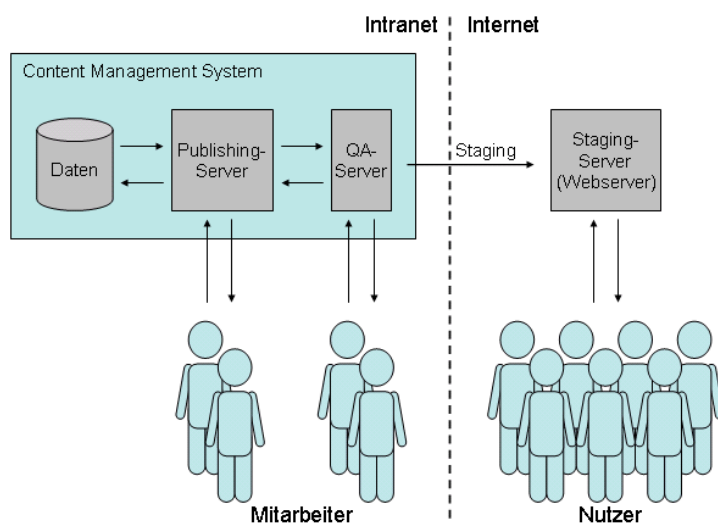
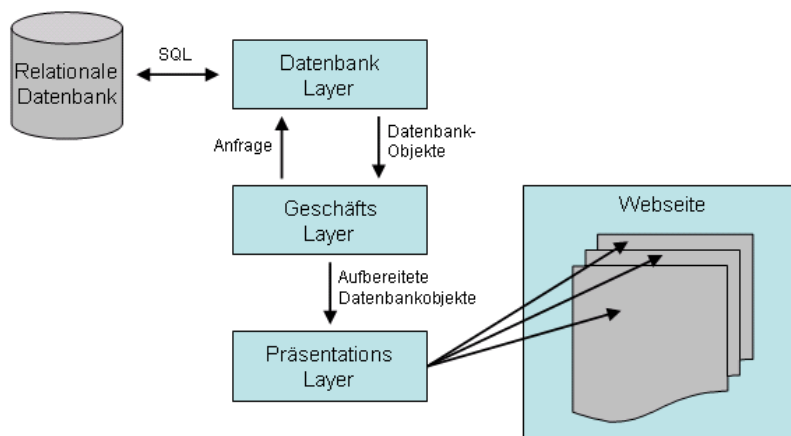


Abbildung 3-6 Prinzip Publishing-/ QA-/ Staging-Server [vgl. Zschau et al. 2002, S. 247]

### 3.6.4 Applikationsserver

Applikationsserver verbinden die Clients und die Präsentationslogik mit den Datenbanken und der Datenhaltungslogik. Auf ihnen laufen die Softwarekomponenten, in denen die Geschäftsprozesse abgebildet werden (Geschäftslogik). [vgl. N.N. 2003a] Zu den wichtigsten Eigenschaften der Applikationsserver zählen die Enterprise Applications Integration (EAI), Sessionmanagement, offene Integrierung von Businesslogik, Clustering, Load-Balancing und Transaktionsmanagement. [vgl. Zschau et al. 2002, S. 247] In Abbildung 3-7 werden die wesentlichen Komponenten eines Applikationsservers aufgezeigt. Über den Datenbank-Layer werden die Daten aus den Datenbanken ausgelesen und an den Geschäfts-Layer weitergegeben. In ihm ist die Geschäftslogik enthalten. Er transformiert die überlieferten Daten zu anwendungsspezifischem Kontext. Durch den Präsentations-Layer wird der Inhalt mit der Struktur und Darstellung verknüpft. [vgl. Busch 2000]



**Abbildung 3-7** Basiskomponenten eines Applikationsservers [vgl. Busch 2000]



## 4 Die Auswahl

Als erstes wird die Systemart ausgewählt. Dabei kommt es auf die Einsatzmöglichkeiten und -gebiete der späteren Webseite an. Bei der Wahl des zweckmäßigsten Systems ging ich nach folgenden Punkten vor:

- Möglichkeit der Trennung von Inhalt und Layout
- Rechteverwaltung mit Freigabeverfahren für Qualitätssicherung
- keine Überdimensionierung des Funktionsangebotes

Eine Einhaltung dieser Punkte ist lt. Kapitel 2.1 *Systemarten* mit einem Web Content Management System gewährleistet. Daher halte ich die Umsetzung eines solchen Systems für die zweckdienlichste.

### 4.1 Wieso ein WCMS für den Fachbereich

Die Aufführung von einigen Beispielen soll zeigen, ob sich die Umsetzung eines WCMS für den Fachbereich IMN der HTWK lohnt. Sie sollen aber nicht zeigen, dass ein WCMS die alleinige Lösung ist. Natürlich können die meisten Hindernisse auch mit dem herkömmlichen Webpublishing beseitigt werden. Aber ein WCMS vereinfacht in vielen Fällen die Arbeit, die durch die anfallenden Probleme auftritt.

Für die derzeitige Homepage des Fachbereichs Informatik, Mathematik und Naturwissenschaften ist Frau Dörte König verantwortlich. Sie selbst sagte: „Für das Auftreten der Hochschule nach außen wäre es sicher besser, wenn die HTWK, gemäß ihrer englischsprachigen Seiten und Corporate Identity, ein einheitliches Layout für alle Seiten, egal ob Rektorat, Dezernate, Fachbereiche oder anderes, hätte. Das bunte Durcheinander ist einfach grauenhaft und wirkt nach außen nicht gerade seriös.“ Der Einsatz eines WCMS ermöglicht mittels Templates eine konsistente Pflege und die Einhaltung des Corporate Designs und der Corporate Identity auf jeder Seite. Außerdem können die Templates getrennt von Inhalt und Funktionalität direkt von Grafikern erstellt werden. Aber nicht nur die Einhaltung des Layouts, sondern auch die Verteilung der Aufgaben auf mehrere Personen erhöht die Präsenz des Fachbereiches. Für die derzeitigen Inhalte könnten und sollten mehr Personen verantwortlich sein. Mit einem WCMS müsste man nicht den Weg

über den Webmaster Frau König gehen (bottleneck<sup>3</sup>), sondern könnte seinen Inhalt von jedem beliebigen Internetzugang aus veröffentlichen. So könnten zum Beispiel Firmen einfacher ihre freien Praktikumsplätze einstellen oder Studenten ihre Bücher anbieten. Durch die Aufteilung der Aufgaben auf technisch wenig bzw. nicht versierte Mitarbeiter und durch den Content Life Cycle wäre die Möglichkeit gegeben, die Aktualität, Qualität und den Umfang der Inhalte stark zu erhöhen. Viele automatisierte Funktionen unterstützen dabei den Publishingprozess. Darunter zählen die Sitemap-Generierung, Suchfunktionen, Link-Checking und automatische Benachrichtigungen über anfallende Tätigkeiten. Die hochschulinternen Prozesse können ebenfalls vereinfacht werden. Zum Beispiel können die Professoren die Noten direkt der Sekretärin und den jeweiligen Studenten zur Verfügung stellen. Dadurch haben die Studenten einen direkten Überblick über ihre erbrachten Leistungen und die Sekretärin müsste keine Aushanglisten anfertigen, da diese direkt aus der Übersicht generiert werden könnten. Das ist durch die medienneutrale Datenstrukturierung möglich, mit der man alle Inhalte in entsprechende Formate bringen kann. Somit sind nicht nur Notenlistengenerierungen denkbar, sondern auch Informations-CD-ROMs, druckbare Veranstaltungskalender und vieles andere.

Derzeit existiert kein Bereich auf der Fachbereichsseite, der frequentierte Neuigkeiten, wie beispielsweise Planänderungen und Lehrveranstaltungsausfälle, beinhaltet. Mit dem WCMS wird ein Neuigkeitenbereich geschaffen, der noch dazu einfach zu aktualisieren ist. Durch die Rechteverwaltung wird die Sichtbarkeit der Bereiche, die für den Nutzer von Interesse sind, angepasst. Professoren bekommen zum Beispiel Übersichten und Benachrichtigungen über noch fehlende Noten oder den Termin für die nächste Fachbereichsratsitzung. Studenten hingegen sehen ihre persönlichen Noten oder Einschreibetermine. Diese Personalisierung schafft Vertrauen zum System und damit auch die Nutzung dessen.

Mittels meiner Konzeption kann eine Lehrveranstaltung geschaffen bzw. vorhandene genutzt werden, welche Studenten nicht nur den Umgang mit Scriptsprachen und Sicherheiten im Internet aufzeigen, sondern auch die Planung und den Aufbau von komplexeren Internetpublikationen lehrt.

---

<sup>3</sup> Flaschenhalsprinzip: Ausführungen von vielen Tätigkeiten sind von einer bzw. wenigen Personen abhängig

## 4.2 Überblick über Content Management Systeme

In diesem Kapitel möchte ich einen Überblick über mögliche Fremdsysteme für den Fachbereich geben. Bis August 2001 gab es schon über 150 Content Management Systeme. Durch die hohe Anzahl und durch meine zeitlich und im Umfang begrenzte Diplomarbeit habe ich anhand der folgenden Kriterien einige CMS aus meiner näheren Betrachtung selektiert. Das CMS für den Fachbereich sollte:

- ein webbasiertes CMS sein
- APIs besitzen
- über Freigabeverfahren verfügen
- Suchmaschinenfunktionen bieten
- die Datenbanken Oracle bzw. MySQL unterstützen
- uneingeschränkt Browser unterstützen
- kostenfrei sein

Anhand der mir vorliegenden Quellen [Krüger/ Kopp 2002], [Zschau et al. 2002], [Reitmaier 2003] und [Wolff 2003] fielen folgende 47 Systeme nicht in die nähere Auswahl: Hyperwave, 3i-IDMF, Appix Portal System, Dimedis@it 3.8, Editor's Office Prefectus, Hyper.Net, InfoSite, Interconomy.net, SchemaText, CM4all, Intrexx 2001, inSITE eBusiness Suite, CoreMedia Publisher 3.1, Absolute Suite, CMS 2001, Communiqué, CONTENTS 2.0 professional, InterNews 2000, CU-E-Business Server, iRacer, NPS 5, RedDot Professional 4.0, TeamSite 5.0, Vignette, VIP 8, 4i WCM Edition, MS Content Management Server, Content Server, eKnowledge Suite, Imperia, MediaSurface, Obtree C3, Pirobase, SixCMS, Tridion DialogServer, VIP ContentManager 8, Webman, Icoya OpenContent, Astarte WebEdition, Campware Campsite, Powerslave, InterRed 3.5, PANSITE, Synformation.com, Roxen Platform, Web4biz und OpenCMS. Die drei Systeme phpCMS, Typo3 und Zope erfüllen alle Kriterien und werden demzufolge näher betrachtet.

### **phpCMS ([www.phpcms.de](http://www.phpcms.de)):**

Zu den Vorteilen von phpCMS zählen seine kostenfreie Nutzung, das Statistik-Modul und das integrierte Caching-Modul. Trotzdem ist das System vielmehr für den kleineren Gebrauch geschaffen, weil es nicht über eine Datenbank bzw. eine Schnittstelle zu einer Datenbank verfügt. Die Verwaltung ist filebasiert und verfügt

daher nicht über Vorteile, wie zum Beispiel performante Speicherung großer Datenmengen und die Rollbackfunktion, welche Datenbanken bieten. [vgl. Zschau et al. 2002, S. 370 f.]

**Typo3 ([www.typo3.com](http://www.typo3.com)):**

Ein großes Plus ist die sehr gute Umsetzung der Benutzeroberfläche für die Ersteller neuer Webseiten, da die Mitarbeiter von Typo3 viele Kriterien von größeren Systemen wie zum Beispiel RedDot und Powerslave abgeschaut haben. Daher ist die Oberfläche übersichtlich und bietet sehr viele Funktionen. Typo3 besitzt innovative Grafikfunktionen. Beispielsweise können Bilder in Fließtexte eingebaut werden und dynamisch erzeugte Navigationsleisten, Überschriften etc. können als Grafikelemente erzeugt werden. Weiterhin besitzt Typo3 einen WYSIWYG-Editor, der aber nur unter Benutzung von Windows und dem Internet Explorer von Microsoft verwendet werden kann. Mittels des Extension Managers kann Typo3 einfach erweitert werden, ohne im Kernsystem zu arbeiten. Weitere Vorteile sind die einfache Versionskontrolle, eine vom Dokumenttyp unabhängige Suchfunktion und die Möglichkeit, statische oder dynamische Seiten zu generieren. Nachteilig kann man die hohen Anforderungen von Typo3 betrachten, wozu PHP4, MySQL, GDlib, Freetype und ImageMagick gehören. Ebenfalls die neu zu erlernende Sprache TypoScript, mit der viele Seiten konfiguriert werden können. Ein weiterer Nachteil ist das fehlende Workflowsystem. Eine hohe Einarbeitungszeit bei einer so hohen Zahl an Konfigurationsmöglichkeiten ist ebenfalls einzuplanen. Aber es existieren genügend Dokumentationen, welche auch zum Teil in deutsch verfasst sind ([www.typo3.info](http://www.typo3.info), [www.typo3.org](http://www.typo3.org)). Das System wird ständig weiterentwickelt. Dazu gehört die neue „Mediabase“, in der man bald alle Metadaten zu jeder Datei in einer Datenbank verwalten kann. Ein weiteres Projekt ist das Live Production Environment (LPE). Damit wird es möglich sein, Teile der Typo3-Datenbank auf andere Server zu replizieren. Das soll die Trennung von Publishing- und Staging-Server ermöglichen und die Einbindung von anderen Datenbanken außer MySQL zulassen. [vgl. Wolff 2003]

**Zope ([www.zope.org](http://www.zope.org)):**

Zope ist eine Plattform zur Realisierung dynamischer Webseiten. Zu den Vorteilen von Zope zählen das rollenbasierte Authentifikationssystem, der integrierte Webserver (ZServer), das Objekt-Datenbanksystem (ZODB), eine Metadatenverwaltung und ein

Versionsmanagement. Des Weiteren ist es möglich, auf externe Datenbanken zuzugreifen und mit anderen Web-Servern zu kooperieren. Zu den Nachteilen zählen die mangelhafte Dokumentation, das nicht vorhandene Workflow-System und keine direkte Unterstützung von XML. Auch ist Zope nur durch die Programmiersprache Python erweiterbar. Für die Erstellung der meisten Seiten braucht man neben Pythonkenntnissen auch Erfahrungen in DTML. Zope ist ebenfalls nur eine Grundlage für Content Management, da es nur rudimentär mit Funktionalitäten ausgestattet ist. [vgl. Zschau et al. 2002; Güttler 2003; N.N. 2002]

Zusammenfassend kann gesagt werden, dass die meisten Systeme nicht für den Fachbereich IMN der HTWK Leipzig geeignet sind. Der Hauptgrund dafür liegt vor allem in der finanziellen Lage der Hochschule. Es stehen keine Gelder für ein WCMS bereit. Somit muss ein solches kostenfrei verfügbar sein. Zu den beschriebenen Systemen ist noch zu erwähnen, dass Zope am Fachbereich vorhanden ist und auch schon eingesetzt wird (E-Learning-Portal<sup>4</sup> der HTWK). Zope verfügt aber im Vergleich zu Typo3 über relativ wenige Funktionen, womit der Einsatz von Typo3 zweckvoller erscheint.

### **4.3 Fremdsystemanschaffung oder Eigenentwicklung**

Viele Systeme, die auf dem Markt existieren, sind nur durch den WCMS-Hype entstanden und werden daher meist nicht den Anforderung gerecht, die man an ein solches hat. Eine Aufzählung der Funktionalitäten, Kriterien und Fakten soll zeigen, ob die Anschaffung eines Fremdsystems im Gegensatz zu einer Eigenentwicklung passender für den Fachbereich IMN der HTWK Leipzig ist.

Ein Vorteil von selbst programmierten Systemen ist die freie Erweiterbarkeit. Diese ist zwar zeit- und arbeitsintensiv, aber direkt an ein Problem angepasst. Von einem Hersteller gelieferte Applikationen und Konnektoren zu anderen Applikationen sind meist sehr allgemein programmiert und daher auch meistens nicht perform. Eine andere Möglichkeit ist eine Auftragserteilung an den Anbieter eines WCMS für angepasste Erweiterungen. Diese sind dann aber meist mit hohen Kosten und Wartezeiten verbunden. Viele Anbieter geben daher dem Nutzer die Möglichkeit, eigene Erweiterungen zu schreiben. Sie bieten

---

<sup>4</sup> <http://elearning.htwk-leipzig.de/>

eine standardisierte und offengelegte API an. Das Problem dabei ist die programmiersprachenabhängige Schnittstelle zum fertigen System. Doch nicht nur die Anbindung muss erlaubt sein, sondern auch die Zugriffsrechte auf die internen Funktionalitäten des WCMS. Acht von 31 Anbietern<sup>5</sup> einer Zusammenstellung der technischen Daten von Web Content Management Systemen besaßen keine API.

Für die Speicherung der anfallenden Datenmengen werden Datenbanken benutzt. Einige Hersteller bieten Flatfile-Systeme an, welche eher nur für kleinere WCMS-Lösungen brauchbar sind. Diese Systeme benutzen Dateisysteme für die Datenspeicherung. Das Verfahren ist nachteilig beim Zugriff mittels anderer Systeme. Der Vorteil ist der geringe Hardware-Aufwand sowie die Kosten. [vgl. Krüger/ Kopp 2002, S. 45] Wenn man ein größeres WCMS betreiben möchte, braucht man professionelle Datenbank-Managementsysteme (DBMS), wie zum Beispiel die der Firma Oracle<sup>6</sup>. Solche DBMS können riesige Datenmengen performant und zuverlässig speichern. Außerdem verfügen sie über viele Funktionen. Dazu gehört unter anderem die Rollback-Funktion, welche eine Wiederherstellung eines Standes einer bestimmten Zeit erlaubt. Mittels Abfragesprachen wie SQL wird auf die Daten zugegriffen. Die HTWK verfügt über Oracle DB und MySQL<sup>7</sup>. Bei einer Entscheidung für diese Speicherart würden somit keine zusätzlichen Kosten anfallen.

Die Rechteverwaltung ist bei allen WCMS-Anbietern anders definiert. Es sollte eine ausführliche Rechteverwaltung vorhanden sein, damit eine spezialisierte Vergabe erfolgen kann. Das heißt, dass man jedem einzelnen Inhaltsobjekt Rechte zuweisen kann. Mögliche Rechte wären zum Beispiel die Erstellung, Betrachtung, Bearbeitung, Löschung, Veröffentlichung und die Freigabe. Um nicht jedem Nutzer separat die einzelnen Rechte zuzuweisen, sollte man die Möglichkeit der Gruppen- und Rolleneinteilung besitzen. Dabei werden die einzelnen Nutzer zu Rollen zugeordnet und diese wiederum in verschiedene Gruppen zugeteilt. Das hat den Vorteil, dass man für mehrere Nutzer die Rechte in einem Arbeitsschritt und zentral abändern kann. Auch hier sollte man großen Wert auf Arbeitsteilung legen und schauen, ob es auch andere Administratoren im System geben darf. Dazu gehören die so genannten Subadministratoren, welche für einzelne Bereiche

---

<sup>5</sup> siehe [Krüger/Kopp 2002, S. 288-293]

<sup>6</sup> <http://www.oracle.com/>

<sup>7</sup> <http://www.mysql.com/>

zuständig sind. Bei eigenen Systemen ist dieser Bereich sehr schwer zu programmieren, da viele Arbeitsbereiche, Rechte und Rollen definiert und diese zusammengeführt werden müssen. Programmierfehler bzw. Irrtümer können hier schwerwiegende Folgen verursachen. Eine ausführliche Evaluation ist daher notwendig.

Eine Suchmaschine ist unabdingbar für einen guten Webauftritt. Aber nicht jede Suchmaschine ist gleich. Wenn man nur reine HTML-Seiten nach einem Suchstring durchsuchen kann und dieser noch genau übereinstimmen muss, ist das für kleinere WCMS-Projekte ausreichend. Bei komplexeren Projekten müssen auch andere Dokumente, wie zum Beispiel PDF, durchsucht werden. Dabei kann zur normalen Suchstringfunktion die phonetische Suche und die Suche nach sachlich verwandten Begriffen verwendet werden. Um die Performance der Suchmaschine so groß wie möglich zu halten, sollte das WCMS über offene Schnittstellen verfügen. Folglich können neue Seiten direkt im System nach Inhalten durchsucht und indexiert werden. Wenn keine Schnittstellen vorhanden sind, muss die Seite von „außen“ durchsucht werden. Dies würde den Server stark auslasten. Selbst programmierte Suchfunktionen können direkt an die Probleme angepasst werden und sind somit performanter. Beispielsweise kann es eine einfache Suchfunktion geben, wenn man eine Namensliste durchsuchen möchte. Wenn aber die komplette Webseite durchsucht werden soll, ist eine erweiterte Suche adäquater.

Wenn in einem Unternehmen mehrere Systeme vorhanden sind, an denen sich der Nutzer identifizieren muss, werden Directory-Server eingesetzt. Das erleichtert die Arbeit für den Administrator, weil er nicht mehr jedem Nutzer für jedes System Rechte und Einstellungen einrichten muss. Aber auch der Nutzer profitiert davon. Er muss sich nur noch einmal anmelden („Single-Sign-On“) und ist somit automatisch für alle Systeme identifiziert. Das spart Zeit und deshalb auch Geld. Dabei arbeitet das System nach dem Standard X.500 bzw. LDAP und authentifiziert den Benutzer gegenüber allen eingesetzten Systemen. Daher sollte man bei einem WCMS darauf achten, dass es ein möglichst offenes System zur Identifizierung ist, das auch mit einem LDAP-Server verbunden werden kann. Leider besitzt der Fachbereich bzw. die Hochschule derzeit keinen globalen Directoryserver. Die Nutzer werden derzeit mit dem Namensdienst NIS (Network Information System) verwaltet. Laut einem Gespräch mit dem Systemadministrator des Fachbereichs IMN (Andreas Flechtner) ist mittels Scripten eine vereinfachte Nutzererstellung für das WCMS möglich.

Es existieren keine CMS-Anbieter, die mehr als 1% Marktanteil haben. [vgl. Christian 2003] Daher muss intensiv geprüft werden, welche Referenzen die einzelnen Anbieter haben, welche Firmen diese Content Management Systeme nutzen und ob sie zufrieden mit dem Angebot und der Leistung sind. Durch den geringen Marktanteil kann es schnell geschehen, dass ein Anbieter Bankrott geht und man ein CMS hat, welches nicht mehr erweitert bzw. gewartet wird.

Wie im Kapitel 4.2 angesprochen, ist keine finanzielle Förderung vorgesehen. Die Gelder, welche einem Fachbereich pro Jahr zur Verfügung stehen, werden für andere Dinge benötigt. Somit muss das System kostenfrei sein.

Ein weiteres Kriterium, das für die Eigenentwicklung spricht, ist der Lehrwert für Studenten. Sie können dabei lernen, wie man größerer Systeme plant und erstellt, wie verschiedene Sprachen zur Umsetzung angewandt werden und wie man Webseiten gestaltet und grafisch aufeinander abstimmt.

Zusammenfassend sprechen die freie und kostenlose Erweiterbarkeit, Nutzung der vorhandenen Datenbank-Managementsysteme, individualisierte Rechteverwaltung, problembezogene und angepasste Suchfunktionen, Möglichkeit der Anbindung an einen Directoryserver, Fremdfirmenunabhängigkeit, Kostenfreiheit und der hohe Lehrwert für Studenten mehr für eine Eigenentwicklung als für eine Fremdsystemanschaffung. Daher fällt meine Entscheidung auf ein selbst programmiertes Content Management System.

#### **4.4 Umsetzungssprachen**

Um herauszufinden, welche Sprachen zum Einsatz kommen, können nicht nur die Kriterien der einzelnen Sprachen betrachtet werden. Vielmehr muss geprüft werden, welche Sprachen am Fachbereich IMN der HTWK Leipzig gelehrt werden. Die zusätzliche Einstellung von Personal bzw. das Einführen einer neuen Computersprache ist nicht vertretbar und auch nicht nötig. Es werden derzeit genügend Programmiersprachen am Fachbereich IMN gelehrt, die für eine Umsetzung eines WCMS ausreichend sind. Ich habe mich für die Meta-Markup-Sprache PHP, die Markup-Sprache XML und die Abfragesprache MySQL entschieden.



Der größte Vorteil von PHP ist die Einfachheit der Programmierung. Es verfügt über eine Vielzahl vordefinierter Funktionen, welche von einfachen Stringbearbeitungen bis hin zu komplexen Datenbankabfragen reichen. Die Sicherheit wird durch die serverseitige Ausführung des Quellcodes garantiert. Somit sieht der Endnutzer den Quellcode nicht, sondern nur das generierte Resultat. Diese Einfachheit und Sicherheit von PHP hat sich durchgesetzt. Tausende Webserver setzen es mittlerweile ein. Scripte und Hilfen sind in einer großen Zahl im Internet verfügbar. PHP lässt sich in fast jeden Webserver integrieren, wobei der Apache Webserver besondere Unterstützung und Funktionen anbietet. [vgl. Krüger/ Kopp 2002, S. 63] Der Fachbereich IMN verfügt über einen Apache Webserver der Version 2.0.47 mit SSL-Verschlüsselung und PHP 4.3.3. Auch die Abfrage von MySQL-Datenbanken ist mittels PHP einfach zu realisieren.

MySQL ist eine schnelle und flexible Multi-User, Multi-Treaded SQL Datenbank. Die Hauptgründe für den häufigen Einsatz von MySQL bei Providern und Suchmaschinenbetreibern sind die hohe Geschwindigkeit, Stabilität und einfache Bedienbarkeit. Es ist z. B. schneller als Oracle DB oder Informix<sup>8</sup>. Am Fachbereich ist die Version 4.0.14. verfügbar. [vgl. Stepken 1999]

XML ist eine Teilmenge von SGML (Beschreibungssprache für Dokumente) und speziell für die Anwendung im Internet konzipiert. Sie ist leicht an die eigenen Bedürfnisse anzupassen und erweiterbar. Zusammenfassend sagt Wilhelm: „XML bietet eine einfache Möglichkeit Inhalte strukturiert zu beschreiben und zu behandeln“ [Wilhelm 2001] Diese Möglichkeit wird durch die so genannte DTD (Document Type Definition) bzw. durch das XML-Schema gegeben, welche die Struktur und die Elemente der XML-Datei definieren. Diese medienneutralen Daten können dann mittels XSL (eXtensible Stylesheet Language) für verschiedene Ausgabeformate modifiziert und dargestellt werden. Derzeit müssen XML-Dateien für die Präsentation im Web in HTML umgewandelt werden, da die meisten Browser Probleme beim Umgang mit XML und XSL haben. [vgl. Zschau et al. 2002, S. 333] Durch die medienneutrale Abspeicherung und Strukturierung wird XML als Schnittstelle zu anderen Systemen genutzt. Aber auch Daten, welche datenbankkonform vorliegen, können ohne XML als Schnittstelle genutzt werden.

---

<sup>8</sup> <http://www-3.ibm.com/software/data/informix/>

## 5 Vorgehensweise bei der Erstellung eines WCMS

### 5.1 Kompetenzen

Ohne geeignetes Personal kann ein WCMS nicht den Erwartungen gerecht werden, die man an ein solches hat. Wie jedes größere System braucht man Mitarbeiter, die spezielles Fachwissen besitzen. Nach der Content Management Bible<sup>9</sup> gibt es acht Personengruppen. Dazu gehören Managers, Business Analysts, Information Architects, Infrastructure Staff, Software Developer, Publications Staff, Content Processing Staff und Content Creation Staff. Diese Gruppen werden meist für eine genauere Spezialisierung in Rollen gegliedert, welche im Folgenden in Anlehnung an [Boiko 2002] und [Christ 2003] genauer beschrieben werden.

#### **Manager**

##### Content Manager

Der Content Manager versteht die Gesamtheit eines CMS und ist daher Leiter der Planung und Ausführung der anfallenden Tätigkeiten. Dabei trägt er eine hohe Verantwortung, damit die Umsetzung und der Betrieb eines CMS erfolgreich ablaufen. Er vertritt die Anforderungen an ein CMS in der Firma und auch außerhalb. Somit ist er auch Feedback-Agent und nimmt daher Nutzeranfragen entgegen, bearbeitet diese und leitet sie weiter.

##### Project Manager

Ein Project Manager betreut das Personal, Umfang, Ablaufplan und Etat der Projekte. Ebenfalls erstellt er die Projektplanung und führt diese durch.

##### Production Manager

Er managt das Produktionspersonal, ordnet die Produktionsprozesse für höchste Effizienz an, ist für die Qualitätssicherung verantwortlich, erstellt Produktionsablaufpläne und setzt diese durch. Auch die Archivierung von

---

<sup>9</sup> siehe [Boiko 2001, S. 177-203]

Content-Objekten gehört zu seinen Tätigkeitsfeldern. Zusammenfassend gesagt ist er Redakteur, Programmierer, Archivar und Betriebsleiter in einem.

### **Business Analysts**

Er findet heraus, wie das Content Management Projekt in die gesamte Organisationsstrategie passt. Zusätzlich erstellt er Strategien für die Zusammenarbeit und Hilfestellung in der Organisation, verwaltet die Auftragsprozesse, erstellt und veröffentlicht den Projektauftrag, entwickelt Strategien für das Arbeitspensum und das Budget des CMS-Betriebes und arbeitet mit dem Projektteam zusammen.

### **Information Architects**

#### Content Analyst

Zu den Aufgaben eines Content Analyst gehört die Erfassung der Inhaltsanforderungen und die Entwicklung eines logischen Designs für die Inhaltsaspekte. Er ist somit für den Aufbau der Inhalte (Inhaltliches Konzept) verantwortlich.

#### Metator

Ein Metator prüft die Korrektheit der erstellten Metadaten und fügt fehlende hinzu. Außerdem schult er andere Mitarbeiter in Bezug auf den Umgang und den Sinn von Metadaten.

### **Infrastructure Staff**

#### CMS Administrator

Er kennt den Aufbau des gesamten CMS und kann daher den ganzen CMS-Prozess, die Mitarbeiter und die Workflows organisieren. Dabei achtet er darauf, dass keine Fehler auftreten und dass alle Prozesse ohne Probleme ablaufen.

### Deployment Analyst

Er entscheidet unter anderem, welche Hard- bzw. Software, Datenbank, Plattform und Server eingesetzt werden.

### Trainer and Documentations Specialist

Der Trainer and Documentations Spezialist ist für die Ausbildung des Personals eines CMSs und die Dokumentation zuständig.

## **Software Developers**

### Software Analyst

Zu den Aufgaben eines Software Analysts gehören die Auswahl der Entwicklungsumgebung und die Erstellung einer Entwicklungsstruktur. In dieser Struktur wird die Erstellung, Versionierung, Prüfung und Gliederung des Programmiercodes organisiert.

Er entscheidet, welche Features eines CMSs speziell angefertigt bzw. von Fremdanbietern genutzt werden. Außerdem schreibt er die Spezifikation und Entwicklungspläne für die Programmierung.

### Template and CMS Developer

Für die Programmierung der Eingabe- und der Seiten-Templates ist der Template and CMS Developer zuständig.

### Custom Application Developer

Sie programmieren die benutzerdefinierte Funktionalität. Dazu gehören unter anderem die Authoring-Funktionen, Messenger und die Navigation.

### Software Integrator

Ein Software Integrator ist für die internen und externen Schnittstellen eines Content Management Systems zuständig.

### Test Analyst and Test Engineer

Die Sicherung der Qualität wird durch die Test Analyst and Test Engineer gewährleistet. Sie erstellen Testpläne und führen diese durch.

## **Publications Staff (Veröffentlichungsmitarbeiter)**

### Publication Analyst

Er gestaltet die Grafik der Seite. Dabei gehört die Auswahl der Gestaltungstools zu seinen Aufgaben. Er erstellt die generelle Spezifikation für die Publikation. Das heißt, die Gestaltungsrichtlinien (Farben, Schriftarten usw.) oder auch Style genannt. Ebenfalls macht er Vorschläge und Pläne für alles, was das Team für die Erstellung und den Betrieb eines CMS braucht.

### Publication Designer

Zu seinen Aufgaben gehört die Erstellung der einzelnen Spezifikationen der Publikation. Er arbeitet mit dem Page Developer und dem Template Developer zusammen, damit die Seiten, die sie gestaltet haben, auch programmiert werden können. Außerdem spricht er sich mit dem Content Analyst ab, damit der Inhalt so formatiert ist, dass er auf der Seite dargestellt werden kann.

### Page Developer

Ein Page Developer setzt die Spezifikation um. Er arbeitet mit dem Template Developer zusammen, damit aus der Gestaltung und dem Template ein fertiges File entsteht.

### User Interface Specialist

Er ist für das Nutzerinterface verantwortlich. Dazu gehören das Weblayout, Web-Application-Design, Nutzbarkeit und Gesamtaussehen der Seite.

## **Content Processing Staff**

### Conversion Analyst

Diese Analysten entwerfen bzw. wählen die Werkzeuge für das CMS-Team aus. Dafür erstellen und entwerfen sie die Spezifikation und Methoden der Programme, welche die Tool Developer umsetzen.

### Tool Developer

Sie erstellen die Programme, welche den Umwandlungsprozess automatisieren. Dazu gehören Programme (Scripte, Filter, ...) zur Informationsumwandlung, Komponentenerstellung, Automatisierung und Metadatengenerierung.

### Content Processor

Sie prüfen die Ergebnisse der automatisierten Prozesse auf Korrektheit und führen manuelle Prozesse aus. Außerdem fügen Sie Metadaten nach vorgeschriebenen Regeln zum Content hinzu.

### Content QA Specialist

Qualitätssicherung ist ein wichtiger Teil bei der Inhaltsaufbereitung. Der Content QA Specialist prüft, ob die Arbeit der automatisierten Prozesse und der Mitarbeiter den gesetzten Qualitätszielen entspricht.

## **Content Creation Staff (Inhaltserstellungs-Mitarbeiter)**

### Acquisitions Manager (Beschaffungs-Manager)

Er ist für die Beschaffung von neuen Contentquellen verantwortlich. Dabei verhandelt er mit den Quellen, erstellt Vereinbarungen und ist Ansprechpartner für den weiteren Kontakt.

### Traffic Cop

Ein Traffic Cop überwacht die Inhaltsanlieferungen der Autoren und externen Quellen. Er sichert den ordnungsgemäßen Ablauf des CMS-Workflows und dass niemand in Verzug gerät. Ebenfalls sorgt er dafür, dass keine „bottlenecks“ auftreten und keine Konflikte bei den Verantwortlichkeiten entstehen.

### Writers und andere Content Creators

Autoren, Illustratoren, Ton-, Video- und Animationsverantwortliche gehören zu dieser Personengruppe. Sie sind daran gewöhnt, Inhalte für

verschiedenste Benutzergruppen zu erstellen und können Inhalte für unterschiedliche Ausgabemedien anfertigen.

Editor

Er prüft und bearbeitet die erstellten Inhalte der Content Creators. Dabei ist er für alle Inhaltstypen (außer Grafiken) und das gesamte Erscheinungsbild verantwortlich.

Graphic-Editor

Er hat dieselben Aufgaben wie der Editor. Nur beziehen sie sich auf alle Grafiken und Bilder im WCMS.

Mitarbeitergruppe	CMS-Planung und Erstellung	CMS-Betrieb
Managers	Content Manager, Project Manager	Content Manager, Production Manager
Business Analysts	Business Analysts (hohe Beteiligung)	Business Analysts (regelmäßige Beteiligung)
Information Architects	Content Analyst	Metator
Infrastructure Staff	Deployment Analyst, Trainer and Documentation Specialist	CMS Administrator
Software Developers	Software Analyst, Template and CMS developer, Custom Application Developer, Software Integrator, Test Analyst and Test Engineer	Template and CMS Developer, Test Engineer
Publication Staff	Publication Analyst, Publication Designer, Page Developer, User Interface Specialist	Publication Designer, Page Developer
Content Processing Staff	Conversion Analyst, Tool Developer, Content Processor, Content QA Specialist	Content Processor, Content QA Specialist
Content Creation Staff	Writers und andere Content Creators, Editor, Grafic-Editor	Acquisitions Manager, Traffic Cop, Writers und andere Content Creators, Editor, Grafic-Editor

**Tabelle 5-1** CMS-Jobs und ihre Einsatzzeiten [Boiko 2001, S. 178 f.]

## 5.2 Briefing

Bei einem Briefing werden die Ziele des Auftraggebers mit dem Auftragnehmer abgesprochen. Dabei wird vor allem geklärt, was die inhaltlichen Schwerpunkte sind und wer das System nutzen wird. [vgl. Zschau et al. 2002, S. 252] Außerdem kann man den zeitlichen Ablauf planen, den Etat und ob Outsourcing vorgesehen ist. Outsourcing ist die Einbeziehung von Fremdfirmen. Im Zusammenhang mit Content Management spricht man auch von Content-Syndication. Dabei stellen Content-Provider Inhalte bereit, welche man für die eigene Seite nutzen kann. [vgl. Versteegen 2001, S. 23] Oft wird dies durch Hyperlinks realisiert. Die Nachteile dieser Methode sind die abweichenden Darstellungsweisen vom eigenen Corporate Design und der Wechsel auf Webseiten der anderen Anbieter. Durch das Austauschprotokoll ICE (Information and Content Exchange Protocol) ist es möglich, individuelle Anpassungen der Informationen vorzunehmen. [vgl. Christ 2003, S. 31]

## 5.3 Konzeption

Die eigentliche Konzeption des Systems besteht aus mehreren Teilen, wobei man bei der Abarbeitung die Reihenfolge einhalten sollte. Denn die meisten Phasen bauen auf den vorherigen auf.

### 5.3.1 Zielsetzung – Vorbereitungsphase

Die einzelnen Punkte des Briefings werden jetzt genauer definiert. Zu Beginn der Arbeit sollten bestehende Webseiten evaluieren, um sich einen Überblick über den aktuellen Stand des Marktes zu verschaffen. Ferner muss die Frage nach den Zielen des WCMS geklärt werden. Danach werden die Themenbereiche festgelegt, ohne auf den konkreten Inhalt einzugehen. Beispielsweise ist zu prüfen, ob und welche Interaktionsmöglichkeiten (Chat, Forum etc.) es geben soll.



### 5.3.2 Inhaltliches Konzept

Bei dem Inhaltlichen Konzept geht es darum, welche Inhalte realisiert werden sollen. Dabei werden hauptsächlich die Kategorien, deren Struktur und die verwendeten Assets aufgeführt. Vereinfacht gesagt, werden die Tabellenköpfe einer Tabelle definiert. Denn der wirkliche Inhalt wird von den Autoren selbst erstellt. Man nennt diese Phase auch Identifikation des Contents. Diese Konzeption sollte ausführlich und umfangreich angefertigt werden, damit es später nicht zu Komplikationen kommt. Es können Rubriken aufgeführt werden, die am Anfang noch nicht zur Realisierungsphase gehören. Dadurch können Schnittstellen für die spätere Anbindung der neuen Komponenten geschaffen werden. Ebenfalls wird festgelegt, welche dauerhaften Quellen benutzt werden und ob die rechtliche Seite für die Veröffentlichung geklärt ist. Beispiele dafür wären monatliche Magazine, aus denen wichtige Termine veröffentlicht werden oder Newsticker von Nachrichtenseiten.

### 5.3.3 Informationsmanagement-Konzept

Welche Formate des Content Repositorys verwaltet bzw. weiterverarbeitet werden, wird im Informationsmanagement-Konzept beschrieben. [vgl. Zschau et al. 2002, S. 254] Hierzu zählen die Text-, Grafik-, Video-, Audio- und 3D-Formate. Falls es Formatumwandlungen geben soll, muss dies vorher geklärt und ermöglicht werden. Um Probleme bei einem Import und Export zu verhindern, werden die einzelnen An- und Auslieferungsarten aufgelistet. Somit wird erkannt, welche Schnittstellen man benötigt.

#### Anlieferungsarten

- Datenbanken
- FTP-Upload
- E-Mails mit bzw. ohne Attachment
- Streaming-Software
- Webseiten-Formulare mit Upload-Funktionalität

#### Auslieferungsarten

- Datenbanken

- FTP-Download
- Newsletter
- Plug-ins
- Audio-/ Videoplayer
- die eigene Webseite

### 5.3.4 Redaktions- und Rechtekonzept

In diesem Konzept werden die Gruppen, Rollen und Rechte festgelegt und welche Prozesse der Inhalt durchlaufen soll. Hierfür werden zuerst Gruppen und Rollen definiert und anschließend die Rolle zu den Gruppen zugeordnet. Danach werden zu den Kategorien die Contentquellen, verantwortliche Gruppen, Rollen und Rechte, Aktualisierungszeiträume, Schnittstellen für den Inhaltsaustausch, Benachrichtigungsarten und Informationstypen aufgeschlüsselt. Die Einteilung in Informationstypen nennt man auch Klassifizierung des Contents und beschreibt, ob ein Asset

*dynamisch* (hoher Grad an Aktualität z. B.: Börsenkurs, Mailbox),

*statisch* (keine Veränderung der Information z. B.: Biographien, Gesetzestexte)

oder *semidynamisch* (Mischform dynamisch/ statisch z. B.: Angestelltenliste,)

ist.

### 5.3.5 Media-Konzept

Im Media-Konzept wird die grafische Benutzeroberfläche (GUI) der Seite beschrieben. Dabei muss darauf geachtet werden, dass die Designrichtlinien nach der Corporate Identity (CI) und dem Corporate Design (CD) eingehalten werden. Des Weiteren werden die einzelnen Benutzerprofile und Inhaltsdarstellungen ausgefertigt. Dazu zählt auch eine ausführliche Navigationsplanung. „Schließlich möchte der Benutzer in einer übersichtlichen Umgebung mit möglichst wenigen Mausklicks an die gewünschten Informationen kommen.“ [Zschau et al. 2002, S. 255] Zu den Richtlinien der benutzerfreundlichen Gestaltung zählen unter anderem die optimale Schriftart und -größe,

Superzeichenbildung<sup>10</sup>, richtiger Einsatz von Farbe, Informationsanordnung, Browserunabhängigkeit, Einführung von Symbolen und Grafiken, Tutorial und kontextsensitive Hilfefunktionen. Dabei sollten die wichtigsten Normen<sup>11</sup> der Gestaltung, wozu die

*DIN 66 234 Teil 8*

(Bildschirmarbeitsplätze, Grundsätze ergonomischer Dialoggestaltung),

*ISO Standard 9241*

(Ergonomische Anforderungen für Bürotätigkeiten mit Bildschirmgeräten),

*DIN EN ISO 14915-1*

(Software-Ergonomie für Multimedia-Benutzungsschnittstellen - Teil 1: Gestaltungsgrundsätze und Rahmenbedingungen),

*DIN EN ISO 14915-2*

(Software-Ergonomie für Multimedia-Benutzungsschnittstellen - Teil 2: Multimedia-Navigation und Steuerung),

*DIN EN ISO 14915-3*

(Software-Ergonomie für Multimedia-Benutzungsschnittstellen - Teil 3: Auswahl und Kombination von Medien) und die

*DIN EN ISO 13407*

(Benutzerorientierte Gestaltung interaktiver Systeme)

zählen, eingehalten werden. Zu beachten ist die Wahl einer nicht zu komplexen sowie einer nicht zu einfachen Darstellung, damit sporadische als auch professionelle Anwender nicht verschreckt werden.

### **5.3.6 Technisches Konzept**

Nach den Ziel- und Anforderungsanalysen muss entschieden werden, ob man eine bestehende IT-Infrastruktur erweitert, ein System neu anschafft oder ob man eine selbst entwickelte Lösung bevorzugt. Nachdem man eine Lösung favorisiert hat, wird geklärt, welche Hardware und Software eingesetzt werden soll. Eine Evaluierung der vorhandenen Komponenten ist hierbei notwendig, da eine Neuanschaffung sehr kostenintensiv ist. Dabei wird dargelegt, welche Server, welches Betriebssystem, welches Datenbanksystem und

---

<sup>10</sup> nach Felix von Cube (1971): Bildung größerer Informationseinheiten aus einzelnen Zeichen

<sup>11</sup> <http://www.normung.din.de/>

welche Programmiersprachen genutzt werden sollen. Anschließend werden Überlegungen zur Art der Ablage und Verwaltung der Daten getroffen. Mit diesen Kenntnissen werden das Datenbankdesign und eine Tabellenbeschreibung angefertigt.

### **5.3.7 Deployment-Konzept**

Die Serverstrategie und die Zeitplanung der einzelnen Prozesse vom Aufsetzen des Systems bis zur Inbetriebnahme werden im Deployment-Konzept bearbeitet. [vgl. Zschau et al. 2002, S. 256] Die Entwicklung, Qualitätssicherung, Präsentation und Archivierung hängen stark zusammen und können auf separaten Servern betrieben werden. Über die Zweckmäßigkeit dieser Variante entscheiden die Anzahl der Mitarbeiter, die Projektgröße und die Bedeutung des Projektes. Die wichtigsten Serverkonzepte sind in → Kapitel 3.6 *Serverkonzepte* aufgeschlüsselt.

### **5.3.8 Schulungskonzept**

Für einen reibungslosen Einstieg und eine effektive Nutzung des fertigen Systems müssen die Anwendergruppen geschult werden. Bei Fremdsystemanschaffungen werden die Schulungsmaßnahmen durch die Firma selbst übernommen bzw. wird ein Schulungsunternehmen dafür engagiert. Bei selbst entwickelten Projekten muss schon in der Entstehungsphase für eine ausführliche Dokumentation gesorgt werden, damit eine schnelle Einarbeitung durch die Projektleitenden gewährleistet werden kann. Der Einlernprozess fällt für die einzelnen Anwendergruppen unterschiedlich aus. Themenschwerpunkte können dabei die Inhaltserstellung, Anwendung von Applikationen, die Fehlerbehebung, die Generierung von weiteren Templates und die Administration von Zugriffsrechten sein. [vgl. Zschau et al. 2002, S. 257]

### **5.3.9 Betriebskonzept**

Damit das System im Onlinebetrieb auch fehlerfrei läuft, wird ein Betriebskonzept angefertigt. Darin wird geklärt, wie bei auftretenden Fehlern bzw. Systemabstürzen vorgegangen wird. Außerdem werden die Personen aufgeführt, die in Notfällen oder bei

auftretenden Fehlern benachrichtigt werden müssen. Falls man den Server nicht selbst betreiben möchte bzw. kann, wird außerdem noch der Internet Service Provider (ISP) bestimmt.

## 5.4 Programmierung

Bei der programmiertechnischen Umsetzung geht es um die Verwirklichung der Konzeption. Die Gestaltung, Funktionalität und das Rechte-Management wird durch die eingeteilten Personen anhand des Inhalts-, Informationsmanagement-, Redaktions-, Rechte- und Media-Konzeptes realisiert. Dabei agieren die Personen in dem für sie bestimmten Bereich. Durch die Erstellung eines Prototyps kann geprüft werden, ob Änderungen erforderlich sind. Er zeigt alle Ausprägungen des Designs und enthält die komplette Navigation und alle Funktionen. Es werden aber nur wenige Menüpunkte verlinkt. [vgl. Krüger/ Kopp 2002, S. 103] Anhand dieses Prototyps kann man die Vorstellungen des Auftraggebers visualisieren und mit ihm die einzelnen Punkte diskutieren und gegebenenfalls abändern. Nachdem der Prototyp befürwortet wurde, wird er komplettiert.

## 5.5 Testphase

Nach der Konzeption mit anschließender Umsetzung folgen intensive Tests. Dabei werden nicht nur die Browserunabhängigkeit und die Performance geprüft. Auch die korrekte Umsetzung des Rechtekonzeptes und ein Funktionscheck gehören dazu. Es darf nicht vorkommen, dass beispielsweise der anonyme Nutzer administrative Rechte bekommt. Diese Tests müssen unbedingt vor dem ersten Onlinebetrieb geschehen. Sie sollten von Betatestern, also Personen die nicht direkt an der Umsetzung des Projektes teilgenommen haben, durchgeführt werden. Somit werden Fehler gefunden, die häufig wegen einer so genannten „Programmierblindheit“ unentdeckt blieben.

## 5.6 Abnahme bzw. Freigabe

Die Abnahme erfolgt durch den Auftraggeber. Nach der Testphase entscheidet er anhand seiner im Briefing festgelegten Vorstellungen, ob das Projekt freigegeben wird oder ob es abgeändert bzw. verbessert werden muss. Wenn der Auftraggeber keine Einwände mehr hat, wird durch ihn das WCMS zum Onlinebetrieb freigegeben.

## 5.7 Onlinebetrieb

Vor der ersten offiziellen Veröffentlichung müssen die Grunddaten in das WCMS eingegeben werden. Dabei sollte das System vorerst nur ausgewählten Personen zur Verfügung stehen, welche die Seite mit den wichtigsten Informationen füllen. Die Dateneingabe erfolgt dabei über die programmierten Eingabemasken und Arbeitsabläufe. Direkte Eintragungen in die Datenbank mittels SQL sollten vermieden werden, da auftretende Eingabefehler dadurch in den meisten Fällen übersehen werden. Von Anfang an sollte die Qualität der Inhalte ein hohes Niveau besitzen. Denn eine Überarbeitung dieser Inhalte wäre zeit- und kostenintensiv und wird in den meisten Fällen nicht durchgeführt. Außerdem würde das bedeuten, dass man bis zur eventuellen Bearbeitung eine qualitativ suboptimale Webseite anbieten würde.

Nachdem das WCMS online gestellt wurde, wird erst die wirkliche Usability auf die Probe gestellt. Es kommt auf die Resonanz bei den Nutzern an, ob die Erstellung erfolgreich war oder Verbesserungen vorgenommen werden müssen. Durch Rücksprachen mit den Benutzern werden das Userinterface und Fehler, die in der Testphase unentdeckt blieben, mit Hilfe von Updates ausgebessert. Ein WCMS wird trotz intensiver Prüfungen und Verbesserungen immer wieder bearbeitet. Denn je besser es funktioniert, desto mehr Anregungen und Anforderungen werden von den Nutzern vorgeschlagen.

## 6 Die Erstellung

### 6.1 Kompetenzen

Die Suche nach Kompetenzen gestaltet sich relativ einfach. An einem Fachbereich, welcher Informatiker und Medieninformatiker ausbildet, ist ausreichend Fachpersonal für Programmierung, Verwaltung und Gestaltung vorhanden.

Die Tätigkeiten des CMS Administrators, Content- und Production Managers sollten Professoren übernehmen. Sie haben die erforderlichen Fähigkeiten und Berechtigungen, um Organisationen durchzuführen. Für die Arbeiten der Software Developer und Content Processing Staff können Professoren und Studenten des Studienganges Informatik eingesetzt werden. Durch ihre Ausbildung in diversen Programmiersprachen, Anwendungen und Konzeptionen sind sie befähigt, komplexe Internetapplikationen zu erstellen. Die Aufgaben des Publications Staff können die Studenten des Studiengangs Medieninformatik übernehmen. Sie haben die Qualifikationen für die gestalterischen Maßnahmen der Webseiten und Benutzeroberflächen. Zum Content Creation Staff, den Information Architects, den Business Analysts und dem Metator zähle ich alle Angestellten und Studenten der Hochschule, Außenstehende und Fremdfirmen. Dabei liegen die Hauptlasten bei dem Personal und den Studenten des Fachbereiches Informatik, Mathematik und Naturwissenschaften. Die genauen Tätigkeiten und Verantwortlichkeiten werden im Kapitel *6.3.4 Redaktions- und Rechtekonzept* genannt.

Durch diese große Trennung von Verantwortlichkeiten und somit Verringerung des Arbeitsaufwandes für jeden einzelnen wird die Bereitschaft der Erledigung der Aufgaben erhöht. Wegen der Separierung von großen Aufgaben in viele kleine Teilaufgaben werden diese durch ihre geringfügige Größe qualitativ besser bearbeitet. Die Aufgaben, welche auf die Studenten aufgeteilt werden, können in den jeweiligen Fächern behandelt und bearbeitet werden. Zum Beispiel das Layout im Fach Multimediales Gestalten. Weiterführende Tätigkeiten, wozu unter anderem die Grundinhaltenstellung, Erweiterung und Pflege zählt, können in einem gesonderten Fach bzw. durch wissenschaftliche Hilfskräfte und Mitarbeiter durchgeführt werden.

## 6.2 Briefing

Was soll eine Seite für den Fachbereich Informatik, Mathematik und Naturwissenschaften der HTWK Leipzig leisten? Um diese Frage zu klären, wurden in mehreren Projektgruppen des Faches Multimedia-Technologie (4. Semester des Matrikel 99I) Evaluationen von Fachbereichsseiten durchgeführt. Außerdem befragte ich Studenten, Mitarbeitern und Professoren der Hochschule. Zusammengefasst soll eine Fachbereichsseite nicht nur den Fachbereich repräsentieren, sondern auch der Informationsvermittlung dienen und kommunikationsfördernd sein. Dabei sollten folgende grobe inhaltliche Schwerpunkte<sup>12</sup> vorhanden sein:

- Informationen zu
  - Fachbereich
  - Personen
  - Gebäuden und Laboren
  - Technik
  - Studiengängen
  - Lehrangebot
- Projektarbeiten
- Neuigkeiten
- Noten
- Einschreibungen
- Statistiken und wichtige Verlinkungen zu anderen Seiten
- Kommunikations- und Interaktionsmittel
- Hilfsfunktionen

Ebenfalls soll die Seite nicht nur den Personen (Professoren, Mitarbeiter, Studenten) des Fachbereichs Informatik, Mathematik und Naturwissenschaften zur Verfügung stehen, sondern auch anderen hochschulinternen Personen, Fremdfirmen, Abiturienten, anderen Schulen und Interessierten. Die genaue Gliederung ist im → Kapitel 6.3.4 *Redaktions- und Rechtekonzept* aufgeführt.

---

<sup>12</sup> Auszug aus einer Evaluation diverser Fachbereichsseiten (Multimedia-Technologie, 4. Semester, 99I Fachbereich IMN der HTWK Leipzig)



Da keine finanzielle Förderung vorgesehen ist, müssen das System kostenfrei erstellt und betrieben und die vorhandenen Ressourcen (Personal, Software und Hardware) genutzt werden.

Die Einbeziehung von Contentquellen ist derzeit nur hochschulintern geplant. Dabei sind Schnittstellen zum E-Learning-Portal<sup>13</sup> der HTWK Leipzig, zu rOCIS/ HTWK3D<sup>14</sup> und zu VIS\_MAP<sup>15</sup> geplant. Denkbar ist auch der Contentaustausch mit anderen Hochschulen. Zum Beispiel in den Bereichen Lehrveranstaltungsscripte, Arbeiten, Publikationen, Umfragen und Statistiken.

## 6.3 Konzeption

### 6.3.1 Zielsetzung – Vorbereitungsphase

Es soll ein WCMS für den Fachbereich geschaffen werden, welches unter anderem folgende Kriterien erfüllt:

- Präsentation des Fachbereichs IMN
- Lehr- und Lernunterstützung für Professoren, Mitarbeiter und Studenten
- Gewährleistung von Aktualität und einheitlichem Design
- Integration von Kommunikationsplattformen
- leichte Pflege auch durch technisch nicht versiertes Personal
- Akzeptanz bei den Benutzergruppen
- Einbeziehung von Firmen in den Publishingprozess
- Erweiterbarkeit und Integrierbarkeit
- Lerneffekt für Studenten infolge der Erstellung

Anhand des Briefings gliedere ich den Inhalt in vier Kategorien:

- Fachbereich (fachbereichs- bzw. hochschulinterne Informationen)
- Studium (Lehrangebot)

---

<sup>13</sup> <http://elearning.htwk-leipzig.de>

<sup>14</sup> Diplomarbeit „Konzeption und Implementierung eines räumlichen Online-Campus-Informationssystem – rOCIS/HTWK3D“ von Michael Frankfurter, 2003

<sup>15</sup> Diplomarbeit „Webbasierte, Grafische Visualisierung von Datenbankinhalten unter besonderer Berücksichtigung von SVG“ von Thomas Lindner, 2003

- Mitgliederbereich (personalisierte Informationen, Kommunikationsplattformen wie Chat, Forum und Schwarzes Brett)
- Service (Suche, Impressum etc.)

### 6.3.2 Inhaltliches Konzept

Es existieren keine inhaltlichen Vorgaben für eine Fachbereichsseite. Daher habe ich intensive Recherchen und Befragungen von Professoren, Mitarbeitern und Studenten durchgeführt. Aus den Resultaten und den in der Zielsetzung definierten Kategorien habe ich einen Navigationsbaum mit einer groben Beschreibung des vorgesehenen Inhaltes und der verwendeten Assets erstellt. Bei der Erstellung wurde darauf geachtet, dass die Tiefe so gering wie möglich ist. Somit ist gewährleistet, dass die gewünschten Informationen mit nur wenigen „Klicks“ erreichbar sind. Zu beachten ist, dass nicht jede aufgeführte Kategorie bzw. Unterkategorie für jeden Nutzer zu jeder Zeit sichtbar bzw. nutzbar ist. Auf allen Seiten (außer PopUps) sind die Menüs, der Seitenkopf, die jeweilige Überschrift und der Navigationspfad sichtbar. In jeder Kategorie bzw. Unterkategorie sind die Funktionen Erstellen, Bearbeiten, Löschen und Archivieren möglich.

Legende:

- [xxx]..... jeweilige ID der Rubrik
- Link zu [xxx] ..... interner Link zur entsprechenden ID
- \* ..... interne bzw. externe Verlinkung

<b>[1] Fachbereich</b>	
	Inhalt: allgemeiner Einleitungstext zum Fachbereich mit grafischer Veranschaulichung Assets: Text; Bild
<b>[101] Informationen</b>	
	Inhalt: Informationen zum Fachbereich; Entstehungsgeschichte; textinterne Verlinkungen zu *Ämter, *Personen, ... Assets: Informationen; Geschichte; Fachbereichsstruktur; Bilder

Fachbereich	[102] Ämter	
		Inhalt: Ämter mit Zuständigkeitsbereich/ Aufgaben, Informationen, Wahltermin, Kontaktinformationen (Verlinkung zu den betreffenden Personen und Seiten) mit Anzeige der Person + Telefon + Link, Zusammenhangserklärung; Organigramme; Link zu HTWK-Ämtern Assets: Amtsname; Aufgabe; *Person(-en); Telefon; URL; Informationen; *Wahltermin; Bild
	[103] Personen	
		Inhalt: Verantwortungsabgabe über Inhalte; Auswahl zwischen Professoren, Mitarbeiter, Studenten; Suche nach Namen und Aufgabe Assets: Text; Link zu [10301], [10302], [10303]; Suche
	[10301] Professoren	
		Inhalt: Liste mit Titel, Name, grobe Kontaktdaten Assets: Titel; Vorname; Nachname; *Raum; *E-Mail; *Homepage
	[1030101] Details	
		Inhalt: komplettes Personendatenblatt mit Titel, Name, Kontaktdaten, Foto, Arbeiten, Downloads, ... Assets: Titel; Anrede; Vorname; Nachname; Professur; *Raum; *zugeteilte Personen; Ämter; Telefon; Fax; *E-Mail; *Homepage; Bild (Person); Personenstatus; Bemerkungen; *Publikationen; *Lehrveranstaltungen; *Diplomarbeiten; *Projektbeteiligungen; Lebenslauf
	[10302] Mitarbeiter	
		Inhalt: Liste mit Titel, Name, grobe Kontaktdaten Assets: Titel; Vorname; Nachname; *Raum; *E-Mail; *Homepage
	[1030201] Details	
		Inhalt: komplettes Personendatenblatt mit Titel, Name, Kontaktdaten, Foto, Arbeiten, Downloads, ... Assets: Titel; Anrede; Vorname; Nachname; Aufgaben; *Raum; *zugeteilte Personen; Ämter; Telefon; Fax; *E-Mail; *Homepage; Bild (Person); Personenstatus; Bemerkungen; *Publikationen; *Lehrveranstaltungen; *Projektbeteiligungen; Lebenslauf
[10303] Studenten		
	Inhalt: Liste mit Name, Matrikel, grobe Kontaktdaten Assets: Vorname; Nachname; Matrikel; *E-Mail; *Homepage	
[1030301] Details		
	Inhalt: komplettes Personendatenblatt mit Name, Kontaktdaten, Foto, ... Assets: Anrede; Vorname; Nachname; Matrikel; Aufgaben; *ist HiWi von; Ämter; Telefon; Fax; *E-Mail; *Homepage; Bild (Person); ausgeliehene Bücher; Bücherausleihfristende; Bemerkungen; *Projektbeteiligungen; Rollen und Rechte	

Fachbereich	Personen	Studenten	[1030302] Neue Studenten anlegen	
			Inhalt: Matrikel, vereinfachte Rollen und Rechtevergabe, Namen und generierte E-Mail Assets: Matrikel; Rolle/Recht; Nachnamen; Vornamen; generierte E-Mail	
				[104] Gebäude und Labore
				Inhalt: kurze Beschreibung; Suche Assets: Text; Suche nach nach Gebäuden, Räume, Raumnummern, Ausstattungen; Links zu [10401], [10402]
				[10401] Gebäude
				Inhalt: Liste mit Namen und Abkürzung + Einordnung auf interaktiver Karte Assets: Name; Abkürzung; Bild (Stadtplan); Link zu virtueller Ansicht
				[1040101] Details
				Inhalt: komplettes Gebäudedatenblatt Assets: Name; Abkürzung; Straße; Hausnummer; Bild (Stadtplan); Erreichbarkeit; Geschichte/ Entstehung/ Namensgebung; Bild (Geschichte); Bild (Gebäude); *Labore
				[10402] Labore
				Inhalt: Liste mit Namen, Gebäude, Raumnummer Assets: Name; *Gebäude; Raumnummer
				[1040201] Details
				Inhalt: komplettes Labordatenblatt Assets: Name; *Gebäude; Raumnummer; Bild (Raum); *Technik; Technikstatus; *Kontaktperson; Telefon; Belegungsplan; Bemerkungen (Reservierungen)
				[1040202] Reservierung
				Inhalt: Reservierungsformular Assets: Name; Raumnummer; *Reservierer; Datum; Grund
				[105] Technik
Inhalt: Kategorienanzeige (Kameras, Projektoren, ...) mit der jeweiligen Technik Assets: Kategorie; *Bezeichnung [10501]				
[10501] Details				
Inhalt: komplettes Technikdatenblatt Assets: Bezeichnung; Bild (Technik); *Kontaktperson; *Labor; Beschreibung; Verleihstatus				
			[10502] Verleih	
			Inhalt: Verleihformular Assets: Bezeichnung; Zeitraum; Abholzeit; *Ausleiher; Grund	

Fachbereich	<b>[106] Arbeiten</b>	
	Inhalt:	Suche; Auswahl der jeweiligen Rubriken oder direkte Wahl der letzten Arbeiten
	Assets:	Suche nach Stichworten, Datum, Personen, ISBN, Jahr; Rubrikname; Name der Arbeit; Datum; Status
	<b>[10601] Diplom-, Bachelor- und Masterarbeiten</b>	
	Inhalt:	Link zu [1060102], [1060103] und Fragenforum; Liste der Arbeiten mit Name (+Zeit) und Status (offen, verteidigt, frei); Newsletter
	Assets:	Link zu [1060102], [1060103], Forum; Name; Zeit; Status
	<b>[1060101] Details</b>	
	Inhalt:	komplette Beschreibung mit Zeit, Mitwirkende, Thema, Inhalt, Ort der Verteidigung; Downloadmöglichkeit
	Assets:	Thema; Ort; Datum; Zeit; *Student; *Gutachter; *Betreuer; Inhalt; Download; Status
	<b>[1060102] Ablaufschritte</b>	
	Inhalt:	Vergabe; Bearbeitungszeiten; Ziel; Abgabe; Begutachtung; Kolloquium; Bewertung; Fehlerquellen; *Prüfungsordnung; *Forum; *Liste der freien Arbeiten
	Assets:	Text; *Forum; *Links; *Antragsformular
	<b>[1060103] Anfertigungshinweise</b>	
	Inhalt:	Inhaltliche Schwerpunkte und Form der Arbeit; Referenzarbeiten
Assets:	Text; *Liste Arbeiten mit Status verteidigt	
<b>[10602] Projekte</b>		
Inhalt:	Projekt ins Leben rufen; Liste der Projekte mit Name; Newsletter	
Assets:	Link zu [1060202]; *Name	
<b>[1060201] Details</b>		
Inhalt:	komplette Beschreibung und Verlinkung	
Assets:	Projektname; Beschreibung; *Mitwirkende; jeweilige Aufgabe; Screenshot; Logo; URL; Download;	
<b>[1060202] Neues Projekt</b>		
Inhalt:	komplette Projektbeschreibung und Ausschreibung; Benachrichtigung an erwünschte Personen	
Assets:	Projektname; Beschreibung; Ausschreibung; Personenliste	
<b>[10603] Publikationen</b>		
Inhalt:	Personenliste; Publikationsrubrikliste; Suche	
Assets:	Personen (Name); Publikationen (Rubrikname); Suche nach Personen, ISBN, Titel, in Beschreibung, ...	
<b>[1060301] Details</b>		
Inhalt:	komplette Beschreibung und Verlinkung	
Assets:	Titel; Beschreibung; *Autor; *Co-Autoren; ISBN; URL; Jahr; Download; Erwerbblink	

Fachbereich	[107] Umfragen und Statistiken	Inhalt: Übersicht der Kategorien mit jeweiligen Statistiken (Name, Datum) und aktuellen Umfragen; neue Umfrage; Newsletter Assets: Kategorie; Name; Datum; Status; Link zu [10702]
	[10701] Details	Inhalt: Komplette Übersicht Assets: Name; Datum; Zeitraum; Grafik; Tabelle; Legende; *Personen; Bemerkungen; Befragte
	[10702] Umfrage erstellen	Inhalt: Fragen + Antwortvorgaben; Zeitraum und Umfragegruppe festlegen Assets: Fragen; Antwortvorgaben; Bemerkungen; Zeitraum; Umfragegruppe (Studenten, bestimmtes Matrikel, ...)
	[2] Studium	Inhalt: Einleitende Worte; Studienort Leipzig; Bild; Links zu den jeweiligen Unterpunkten Assets: Text; Bild; *Links
	[201] Studiengänge	Inhalt: Übersicht aller Studiengänge Assets: Text; Bild
	[20101] Details	Inhalt: Komplette Übersicht des Studienganges Assets: Akademischer Grad; Regelstudienzeit; Voraussetzungen; Vorpraktikum; Studieninhalt/ziel; Studienaufbau; Berufsbild/ Tätigkeitsfelder; Studienpläne (Grund-, Hauptstudium); *Labore; weitere Texte; *Studien- u. *Prüfungsordnungsdownload; Bilder
	[202] Lehrveranstaltungen	Inhalt: Informationen; Suche und Filterfunktion Assets: Text; Suche; Filter (Studiengang -> Semester oder nur Matrikel)
	[20201] Anzeige	Inhalt: Anzeige der einzelnen Lehrveranstaltungen Assets: Name; Art (Pflicht, ...); Status in Bezug auf Student/ Professor
	[20202] Details	Inhalt: Komplette Übersicht der Lehrveranstaltung Assets: Bezeichnung; Studiengang; *Lesender; Inhalt; SWS; Art; Semester; Abschluss; Zugehörigkeit zu Studienschwerpunkt; Vorkenntnisse; Literatur; *Erwerbslink; *Scripte; *URL; *Einschreibung; *Stundenplan; andere *Downloads; Status in Bezug auf Student/ Professor
	[20203] Einschreibung	Inhalt: zeitbegrenztes Einschreibeformular Assets: Fach; Semester; Matrikel; max. Anzahl Studenten; Zeitraum

Studium	[203] Praktikum	Inhalt: Link zu [20301], [20302] und Praktikumsordnung; Newsletter Assets: Text; *Links; *Praktikumsordnungsdownload
	[20301] Praktikumsstellen	Inhalt: Auflistung aller Praktikumsstellen + Detail Assets: Name; Tätigkeitsbeschreibung; Firma; *Link zu Firma; Arbeitsort; Bemerkungen; Bezahlung; Anzahl noch offener Stellen/ Gesamtstellenanzahl; Kontaktart; Telefon; Fax; *E-Mail
	[20302] Ablaufschritte	Inhalt: von der Suche über die Anmeldung bis hin zur Praktikumsarbeit Assets: Allgemeine Hinweise; Anmeldung; Zulassung; Anerkennung; *Download (Zeugnis und Tätigkeitsnachweis für Firma, Ausbildungsvertrag, Informationsblatt, Fächernachweis)
	[204] Studium generale	Inhalt: Hinweise; Liste aller Veranstaltungen Assets: Hinweise; *Liste (Vorlesung; Leitung; Status); Link zu externe Veranstaltungen
	[20401] Details	Inhalt: komplette Detailansicht Assets: Vorlesung; *Leitung; Beschreibung; Form (Art der Haltung); Termin; Ort; derzeitige/ max. Teilnehmerzahl (Status); *Einschreibung [20203]; Status im Bezug auf Student/ Professor/ andere Leitung; Warteliste
	[205] Hilfe zum Studium	Inhalt: Übersicht (HiWi-Tätigkeiten; Auslandsstudium; Semesterticket; Wohnheime; Studnet; Hochschulsport; ISIC; Mensaspiseplan; Studentenwerk; Fachbereichstage; Podium; RRZN; Fachschaft ...) Assets: Thema
	[20501] Details	Inhalt: komplette Übersicht zum Thema Assets: Thema; Beschreibung; *Link; *Download
	[3] Mitgliederbereich	Inhalt: direkter Sprung zu [301] Assets: ~
	[301] Desktop	Inhalt: Übersicht der wichtigsten Ereignisse des Mitgliederbereichs, ...; kleine Kalenderansicht; Liste der Favouriten Assets: (Datum, Zeit) oder (Zeitraum); Text; *Kalenderansicht

Mitgliederbereich	[302] Kalender	
	Inhalt: Nachrichten (Ausfall; Terminänderung; *Verteidigung; Krankheit; Einschreibung; *neue Post; Prüfung; freie Tage; Termine); Kalender mit grafischer Veranschaulichung der einzelnen Nachrichten	
	Assets: Kalender; (*)Nachricht mit (Zeit/Datum) oder (Zeitraum) und je nach Fall (*Lehrveranstaltung; Betreff; Beschreibung; *Person; (*)Raum; URL; Ort)	
	[303] Noten	
	Inhalt: direkter Sprung zu [30301] bzw. [30302] anhand von Login	
	Assets: ~	
	[30301] Student	
	Inhalt: Übersichtstabelle; Newsletter	
	Assets: Semester; *Fach; Zwischennoten; Endnote/ Bestanden;	
	[30302] Professor/ Mitarbeiter	
	Inhalt: Übersichtstabelle	
	Assets: Matrikel; *Fach	
	[3030201] Personenliste	
	Inhalt: Liste aller Personen eines Matrikels in einem Fach	
	Assets: *Name; Zwischennoten; Endnote/ Bestanden; Fachart; Prüfungsleistungen; Prüfungsdatum	
[304] Forum		
Inhalt: Rubrikenübersicht + Status (Bspw.: Freizeit [45; 2 neu]; Newsletter		
Assets: Forumsrubrik; Themen; Beiträge; letzter Beitrag (Datum, Uhrzeit, *Person)		
[30401] Rubrik		
Inhalt: Themenübersicht + Status		
Assets: Thema; Antworten; *Autor; Aufrufe; letzter Beitrag (Datum, Uhrzeit, *Person)		
[3040101] Thema		
Inhalt: alle Beiträge zum Thema		
Assets: *Autor; Titel; Nachricht; Verfassungsdatum		
[305] Schwarzes Brett		
Inhalt: Rubrikenübersicht + Status (Bspw.: Bücher [24; 4 neu; 3 Suchen]); Newsletter		
Assets: Artikelrubrik; Status (Anzahl, Neuzugänge, Suchen)		
[30501] Rubrik		
Inhalt: Artikelübersicht + Status; Such-Übersicht		
Assets: Artikelbezeichnung; Thumbnail; Einstelldatum; Preis		
[3050101] Artikeldetails		
Inhalt: komplette Übersicht zum Artikel		
Assets: Artikelbezeichnung; Beschreibung; Kontakt; Bild; Einstelldatum; Preis; *Verkäufer		



Mitgliederbereich	Schwarzes Brett	Rubrik	[3050102] Suchdetails	
			Inhalt: komplette Übersicht zur Suche Assets: *Käufer; Artikel; Beschreibung; Kontakt; Suchdatum; evt. Preis	
		[306] Chat		Inhalt: Chatraumübersicht + Status Assets: Chatraum; Status (Personenanzahl)
		[30601] Chatraum		Inhalt: Benutzerübersicht + Status; Chatfenster Assets: Benutzer*; Status; Chatfenster; Eingabezeile
		[307] Karriere		Inhalt: Verlinkung zu den Unterkategorien Assets: Links zu [30701], [30702] und [30703]; Bild; Text
		[30701] Firmenporträts		Inhalt: Vorstellung der Firma Assets: Name; Portofolio; Ansprechpartner; Adresse; Telefon; *E-Mail; *Homepage
		[30702] Praktikumsstellen		Inhalt: Verweis auf Praktikumsstellen Assets: direkt zu [20301]
		[30703] Stellenangebote		Inhalt: Stellendetails, Firma + Kontakt Assets: Stellenbezeichnung; Beschreibung; Anzahl Stellen; Einstelldatum Firma*
		[308] Newsletter		Inhalt: Übersicht über alle bestellten Themen Assets: Thema
		[30801] Hinzufügen neuer Themen		Inhalt: Übersicht aller Themen (Bspw. Anzeige der Sitemap + extra Verteilerlisten) Assets: Thema; Status
		[309] Post		Inhalt: Liste Posteingang; Link zu NOCC Assets: Betreff; *Absender; Datum; Status
		[30901] Nachricht lesen		Inhalt: Details Assets: Betreff; Nachricht; *Absender; Datum; Anhang
		[30902] Neue Nachricht		Inhalt: Nachricht und Empfänger + Anhang Assets: Betreff; Nachricht; Empfänger; Anhang

Mitgliederbereich	[310] Rechtevergabe	Inhalt: Zusammengehörigkeit über Gruppen, Rollen und Rechte Assets: Gruppen; Rollen; Rechte
	[311] Admin-Panel	Inhalt: Seite unter bestimmten Bedingungen anzeigen (z.B. anderes Datum, andere Benutzer); Zeiten der Elemente anzeigen; Datenbankanknüpfung; Konfigurationen, neue Strukturpunkte anlegen, ...  Assets: Funktionsverlinkungen
	[4] Service	Inhalt: ~ Assets: ~
	[401] Suche	Inhalt: Suche mit Filterfunktion Assets: Suchfeld; Filterauswahl
	[402] Impressum	Inhalt: Kontaktinformationen zur Homepage; Liste der Bearbeiter* Assets: Adresse; Telefon; Telefax; *E-Mail; Personen*
	[403] Hilfe	Inhalt: Hilfeformular; *Ansprechpartner; Adressen Assets: Grund; Beschreibung; *Personen; Adressen

Tabelle 6-1 Navigationsbaum der IMN-Seite

### 6.3.3 Informationsmanagement-Konzept

Die möglichen Formate, die im Content Repository vorhanden sein können, sind in Tabelle 6-2 veranschaulicht. In Tabelle 6-3 sind die Im- und Exportarten aufgeführt, mit denen das System mit der „Außenwelt“ kommuniziert.

Formatart	Format
Text	HTML, XLS, PPS, Textverarbeitungs-Dokumente (TXT, DOC, PDF, PS)
Grafik	GIF, BMP, JPEG, PNG
Video	Flash, Shockwave, RealVideo, AVI, MPEG, Quicktime
Audio	MP3, WAV, MIDI, RealAudio

Tabelle 6-2 Anlieferungsformate

Anlieferungsarten	Auslieferungsarten
MySQL-/ Oracle-Datenbanken	Webseite
Webseiten-Formulare	E-Mail-Newsletter
FTP-Upload	Playerroutinen für Audio und Video
Datenscripte	Offline-Medien (CDs, Broschüren, Listen)
	Messenger-Nachrichten

Tabelle 6-3 An- und Auslieferungsarten

### 6.3.4 Redaktions- und Rechtekonzept

In Anlehnung an die derzeitige Personalstruktur des Fachbereiches IMN der HTWK Leipzig habe ich die benötigten Gruppen und Rollen abgebildet. Ebenfalls wurde gesondert die Personalstruktur eines WCMS erstellt. Durch diese Umsetzung entstanden die Gruppen *Angestellte*, *Fachbereichsrat*, *Studierende*, *Gäste*, *WCMS-Personal* und *Anonyme*. In der Gruppe *WCMS-Personal* wurde bewusst nicht die Rolle der Content Creators explizit erfasst, weil jede Person diese Rolle übernehmen kann. Dennoch gibt es einzelne Creator-Verantwortlichkeiten für spezielle Bereiche. Fast alle Gruppen besitzen Rollen. Jede Rolle erbt die Rechte von seiner Gruppe bzw. übergeordneten Rolle. Zum Beispiel hat ein *Dekan* die gleichen Rechte wie *Angestellte* und zusätzlich noch spezielle Rechte. In der folgenden Tabelle werden die Gruppen- und Rollenbezeichnungen mit einer jeweiligen ID dargestellt. Eine Person kann mehreren Gruppen und Rollen angehören.

1 Angestellte	
	101 Dekan
	102 Dekansekretärin
	103 Prodekan
	104 Prodekansekretärin
	105 Studiendekan
	10501 Informatik
	10502 Mathematik
	106 Praktikumsbeauftragter
	10601 Informatik
	10602 Medieninformatik
	10603 Mathematik
	107 Vorsitzender Prüfungsausschuss
	10701 Informatik
	10702 Mathematik
	108 Studienfachberater
	10801 Informatik
	10802 Medieninformatik
	10803 Mathematik

	109 Studienorganisation 110 Prüfungsorganisation
<b>2 Fachbereichsrat</b>	
<b>3 Studierende</b>	
	301 Informatiker
	30101 Diplom FH 30102 Bachelor 30103 Master
	302 Medieninformatiker
	30201 Bachelor 30202 Master
	303 Wirtschaftsmathematiker
	30301 Diplom FH
	304 Angewandte Mathematiker
	30401 Bachelor
<b>4 Gäste</b>	
	401 Firmen 402 Schulen
<b>5 WCMS-Personal</b>	
	501 Manager
	50101 Content Manager 50102 Production Manager
	502 Business Analyst
	503 Information Architect
	50301 Metator
	504 Infrastructure Stuff
	50401 CMS Administrator
	505 Software Developer
	50501 Template Developer 50502 Test Engineer
	506 Publication Staff
	50601 Publication Designer 50602 Page Developer
	507 Content Processing Staff
	50701 Content Processor 50702 Content QA Specialist
	508 Content Creation Staff
	50801 Acquisitions Manager 50802 Traffic Cop 50803 Editor 50804 Graphic-Editor 50805 Special Creator 50806 Equipment Creator 50807 Person Creator 50808 Lab Creator 50809 Studium Generale Creator
<b>6 Anonyme</b>	

Tabelle 6-4 Gruppen- und Rollenbezeichnungen

Die Rechte, welche jede Gruppe und Rolle besitzen kann, sind:

- view (betrachten)
- create (erstellen)
  - 1. Phase: Content wird erstellt; bei unkorrektem Content darf er auch bearbeitet werden
- edit (bearbeiten)
  - 2. Phase: Content wird bearbeitet; bei fehlendem Content darf er auch neu erstellt werden
- release (veröffentlichen)
- delete/ archive (löschen/ archivieren)
  - inhaltsabhängig

Im folgenden Schritt verbinde ich das Inhaltliche Konzept mit den Gruppen und Rollen. Dabei wird nicht nur die Kategorie mit einem Workflow verbunden, sondern jedes Asset dieser Kategorie. Dieser Schritt ist zwar in der Konzeption aufwendig, aber durch die Rechtevererbung in der späteren Nutzung einfach zu bedienen. Daneben wird eine vollständige Kontrolle gewährleistet. Zusätzlich werden weitere Angaben zu dem Content gemacht, um spätere Phasen der Konzeption zu vereinfachen.

**Anmerkung:**

- Links zu Bereichen, auf denen man das Recht „view“ nicht hat, gelten automatisch als inaktiv bzw. nicht sichtbar für die betroffenen Personen.
- die Spalte *Kategorie* beinhaltet die IDs der Tabelle 6-1
- die Spalten *view*, *create*, *edit*, *release*, *delete* und *archive* beinhalten die IDs der Tabelle 6-4

**Legende:**

autom. automatisch  
n. nicht  
n.def. nicht definiert  
semidyn. semidynamisch  
del/arch delete/ archive

Kategorie	Content	view	create	edit	release	del/arch	Quelle	Info-Typ	Aktualisierung	Schnittstelle
[1]	Text	alle	[105]	[50803]	[101]	[50101]	intern	statisch	n.def.	n.def.
	Bild	alle	[105]	[50804]			intern	statisch	n.def.	n.def.
[101]	Bilder	alle	[102], [105]	[50804]	[102]	[50101]	intern	statisch	n.def.	n.def.
	Rest	alle	[102], [105]	[50803]			intern	statisch	n.def.	n.def.
[102]	Bild	alle	[102]	[50804]	[102]	[102]	intern	statisch	n.def.	n.def.
	Wahltermin	n. [4], [6]	[2], [102]	[50803]		autom.	intern	dynamisch	semesterweise (bei Neuwahl)	PDF, E-Mail
	Rest	alle	[102]	[50803]		[102]	intern	semidyn.	semesterweise (bei Neuwahl)	PDF
[103]	alles	alle	[50805]	[50803]	[50805]	[50101]	intern	statisch	n.def.	n.def.
[10301]	alles	n. [6]	autom.	autom.	autom.	autom.	DB	dynamisch	bei Aufruf	n.def.
[1030101]	Foto	alle	jew. [1], [50807]	[50804]	jew. [1]	[50807]	intern	semidyn.	jährlich	n.def.
	zug. Personen, Publi- kationen, Diplomarbeiten	n. [6]	jew. [1], [50807]	[50803]	jew. [1]	[50807]	intern	semidyn.	n.def.	n.def.
	Rest	alle	jew. [1], [50807]	[50803]	jew. [1]	[50807]	intern	semidyn.	n.def.	n.def.
[10302]	alles	n. [6]	autom.	autom.	autom.	autom.	DB	dynamisch	bei Aufruf	n.def.
[1030201]	Foto	alle	jew. [1], [50807]	[50804]	jew. [1]	[50807]	intern	semidyn.	jährlich	n.def.

Kategorie	Content	view	create	edit	release	del/arch	Quelle	Info-Typ	Aktualisierung	Schnittstelle
	zug. Personen, Publikationen	n. [6]	jew. [1], [50807]	[50803]	jew. [1]	[50807]	intern	semidyn.	n.def.	n.def.
	Rest	alle	jew. [1], [50807]	[50803]	jew. [1]	[50807]	intern	semidyn.	n.def.	n.def.
[10303]	alles	n. [6]	autom.	autom.	autom.	autom.	DB	dynamisch	bei Aufruf	n.def.
[1030301]	Foto	n. [4], [6]	jew. [3], [50807]	[50804]	[50807]	[50807], autom.	intern	semidyn.	n.def.	n.def.
	Rest	n. [4], [6]	jew. [3], [50807]	[50803]	[50807]	[50807], autom.	intern	semidyn.	n.def.	n.def.
[1030302]	alles	[50807]	[50807]	[50803]	[50807]	-	Script, intern	semidyn.	semesterweise	Script
[104]	alles	n. [4], [6]	[50805]	[50803]	[50805]	[50101]	intern	statisch	n.def.	n.def.
	alles (eingeschränkte Suche)	[4], [6]								
[10401]	Bild	alle	[50805]	[50804]	[50805]	[50101]	intern	statisch	n.def.	n.def.
	Rest	alle	[50805]	[50803]		[50101]	intern	statisch	n.def.	n.def.
[1040101]	Bilder	alle	[50805]	[50804]	[50805]	[50101]	intern	statisch	n.def.	n.def.
	Rest	alle	[50805]	[50803]		[50101]	intern	statisch	n.def.	n.def.
[10402]	alles	alle	autom.	autom.	autom.	autom.	DB	dynamisch	bei Aufruf	n.def.
[1040201]	Bild	alle	[50808]	[50804]	[50808]	[50808]	intern	semidyn.	n.def.	n.def.
	Technik, Technikstatus, Belegungsplan, Bemerkungen	n. [6]	[50808]		[50808]	[50808]	intern	semidyn.	n.def.	n.def.

Kategorie	Content	view	create	edit	release	del/arch	Quelle	Info-Typ	Aktualisierung	Schnittstelle
	Rest	alle	[50808]	[50803]	[50808]	[50808]	intern	semidyn.	n.def.	n.def.
[1040202]	alles	n. [4], [6]	[1]	-	-	[50808]	intern	semidyn.	n.def.	n.def.
[105]	alles	alle	autom.	autom.	autom.	autom.	DB	dynamisch	bei Aufruf	n.def.
[10501]	Bild	alle	[50806]	[50804]	[50806]	[50806]	intern	semidyn.	n.def.	n.def.
	Labor, Verleihstatus	n. [4], [6]	[50806]	[50803]	[50806]	[50806]	intern	semidyn.	n.def.	n.def.
	Rest	alle	[50806]	[50803]	[50806]	[50806]	intern	semidyn.	n.def.	n.def.
[10502]	alles	n. [4], [6]	[1], [3]	-	-	[50806]	intern	semidyn.	n.def.	n.def.
[106]	alles	alle	autom.	autom.	autom.	autom.	DB	dynamisch	bei Aufruf	n.def.
	alles (eingeschränkte Suche)	[4], [6]	autom.	autom.	autom.	autom.	DB	dynamisch	bei Aufruf	n.def.
[10601]	alles	alle	autom.	autom.	autom.	autom.	DB	dynamisch	bei Aufruf	n.def.
[1060101]	alles	alle	[1]	[50803]	[1]	autom.	intern	semidyn.	n.def.	n.def.
[1060102]	alles	n. [4], [6]	[105]	[50803]	[105]	[50101]	intern	statisch	n.def.	n.def.
[1060103]	alles	n. [4], [6]	[105]	[50803]	[105]	[50101]	intern	semidyn.	n.def.	n.def.
[10602]	alles	alle	autom.	autom.	autom.	autom.	DB	dynamisch	bei Aufruf	n.def.
[1060201]	Screenshot, Logo	alle	[1]	[50804]	[1]	[1]	intern	statisch	n.def.	n.def.
	Rest	alle	[1]	[50803]	[1]	[1]	intern	statisch	n.def.	n.def.
[1060202]	alles	n. [4], [6]	[1]	[50803]	[1]	-	intern	dynamisch	n.def.	n.def.
[10603]	alles	n. [6]	autom.	autom.	autom.	autom.	DB	dynamisch	bei Aufruf	n.def.
[1060301]	alles	n. [6]	[1]	[50803]	[1]	[50101]	intern	statisch	n.def.	n.def.
[107]	alles	alle	autom.	autom.	autom.	autom.	DB	dynamisch	bei Aufruf	n.def.
[10701]	Grafiken	alle	[50805]	[50804]	[50805]	[50101]	intern			PDF



Kategorie	Content	view	create	edit	release	del/arch	Quelle	Info-Typ	Aktualisierung	Schnittstelle
	Rest	alle	[50805]	[50803]	[50805]	[50101]	intern			
[10702]	alles	n. [4], [6]	[1], [2], [3]	[50805]	autom.	[50101]	intern	dynamisch	n.def.	n.def.
[2]	Bild	alle	[108]	[50804]	[108]	[50101]	intern	statisch	n.def.	n.def.
	Rest	alle	[108]	[50803]	[108]	[50101]	intern	statisch	n.def.	n.def.
[201]	Bild	alle	[108]	[50804]	[108]	[50101]	intern	statisch	n.def.	n.def.
	Rest	alle	[108]	[50803]	[108]	[50101]	intern	statisch	n.def.	n.def.
[20101]	Bild	alle	[108]	[50804]	[108]	[50101]	intern	semidyn.	n.def.	PDF
	Rest	alle	[108]	[50803]	[108]	[50101]	intern	semidyn.	semesterweise	
[202]	alles	alle	autom.	autom.	autom.	autom.	DB	dynamisch	bei Aufruf	n.def.
[20201]	alles	alle	autom.	autom.	autom.	autom.	DB	dynamisch	bei Aufruf	n.def.
[20202]	Scripte, URL, Stundenplan, Downloads	n. [4], [6]	jew. [1]	[50803]	jew. [1]	[50101]	intern	semidyn.	min. semesterweise	PDF
	Rest	alle	jew. [1]	[50803]	jew. [1]	[50101]	intern	semidyn.	semesterweise	PDF
[20203]	alles	jew. [1], [3]	jew. [1], [3]	jew. [1]	jew. [1], autom.	jew. [1], [3]	intern	semidyn.	n.def.	PDF
[203]	alles	alle	[106]	[50803]	[106]	[106]	intern	statisch	n.def.	n.def.
[20301]	alles	n. [6]	[106], [401]	[50803]	[106], autom.	[106], [401], autom.	intern, extern	semidyn.	n.def.	n.def.
[20302]	alles	n. [4], [6]	[106]	[50803]	[106]	[50101]	intern	semidyn.	n.def.	PDF
[204]	alles	alle	[402], [50809]	[50803]	[50809]	[50101]	intern, extern	semidyn.	n.def.	n.def.
[20401]	alles	alle	[402],	[50803]	[50809]	[50101]	intern,	semidyn.	n.def.	n.def.

Kategorie	Content	view	create	edit	release	del/arch	Quelle	Info-Typ	Aktualisierung	Schnittstelle
			[50809]				extern			
[205]	alles	alle	[50805]	[50803]	[50805]	[50101]	intern	semidyn.	n.def.	n.def.
[20501]	alles	alle	[105], [50805]	[50803]	[50805]	[50101]	intern	semidyn.	n.def.	n.def.
[3]	-	-	-	-	-	-	-	-	-	-
[301]	angepasst	n. [6]	autom.	autom.	autom.	autom.	DB	dynamisch	bei Aufruf	n.def.
[302]	alles	[1], [3]	[1], [3]	[104]	[104]	[50101]	intern	dynamisch	n.def.	n.def.
[303]	alles	-	-	-	-	-	-	-	-	-
[30301]	alles	[3]	autom.	autom.	autom.	autom.	DB	dynamisch	bei Aufruf	n.def.
[30302]	alles	jew. [1]	autom.	autom.	autom.	autom.	DB	dynamisch	bei Aufruf	n.def.
[3030201]	alles	jew. [1]	jew. [1]	[103]	jew. [1]	jew. [1], [103]	intern	semidyn.	n.def.	PDF
[304]	alles	n. [6]	autom.	autom.	autom.	autom.	DB	dynamisch	bei Aufruf	n.def.
[30401]	alles	n. [6]	autom.	autom.	autom.	autom.	DB	dynamisch	bei Aufruf	n.def.
[3040101]	alles	n. [6]	n. [6]	[50803]	autom.	[50101]	intern	semidyn.	n.def.	n.def.
[305]	alles	n. [6]	autom.	autom.	autom.	autom.	DB	dynamisch	bei Aufruf	n.def.
[30501]	alles	n. [6]	autom.	autom.	autom.	autom.	DB	dynamisch	bei Aufruf	n.def.
[3050101]	Bild	n. [6]	n. [6]	[50804]	autom.	jew. Creator, n. [6]	intern	semidyn.	n.def.	n.def.
	Rest	n. [6]	n. [6]	[50803]	autom.	jew. Creator, n. [6]	intern	semidyn.	n.def.	n.def.
[3050102]	alles	n. [6]	autom.	autom.	autom.	autom.	DB	dynamisch	bei Aufruf	n.def.
[306]	alles	n. [6]	autom.	autom.	autom.	autom.	DB	dynamisch	bei Aufruf	n.def.

Kategorie	Content	view	create	edit	release	del/arch	Quelle	Info-Typ	Aktualisierung	Schnittstelle
[30601]	alles	n. [6]	[50805]	[50803]	[50805]	[50101]	intern	semidyn.	n.def.	n.def.
[307]	Bild	n. [6]	[50805]	[50804]	[50805]	[50101]	intern	statisch	n.def.	n.def.
	Rest	n. [6]	[50805]	[50803]	[50805]	[50101]	intern	statisch	n.def.	n.def.
[30701]	alles	n. [6]	[401]	[50803]	[401]	[401]	extern	semidyn.	n.def.	n.def.
[30702]	alles	n. [6]	-	-	-	-	-	-	-	-
[30703]	alles	n. [6]	[401]	[50803]	[401]	[401], autom.	extern	semidyn.	n.def.	n.def.
[308]	alles	n. [6]	autom.	autom.	autom.	autom.	DB	dynamisch	bei Aufruf	n.def.
[30801]	alles	n. [6]	[50805]	[50803]	[50805]	[50101]	intern	semidyn.	n.def.	n.def.
[309]	alles	n. [6]	autom.	autom.	autom.	autom.	DB	dynamisch	bei Aufruf	n.def.
[30901]	alles	n. [6]	autom.	autom.	autom.	autom.	DB	dynamisch	bei Aufruf	n.def.
[30902]	alles	n. [6]	n. [6]	-	-	-	intern	dynamisch	n.def.	n.def.
[310]	alles	[504], [510]	[504], [510]	-	-	[504], [510]	intern	semidyn.	n.def.	n.def.
[311]	alles	[504], [510]	[504], [510]	-	-	[504], [510]	intern	semidyn.	n.def.	n.def.
[4]	-	-	-	-	-	-	-	-	-	-
[401]	alles	alle	autom.	autom.	autom.	autom.	intern	dynamisch	n.def.	n.def.
[402]	alles	alle	[50805]	[50803]	[50805]	[50101]				
[403]	alles	alle	[50805]	[50803]	[50805]	[50101]	intern	semidyn.	n.def.	n.def.

Tabelle 6-5 Verhältnistabelle von Kategorie, Content und Rechte

### 6.3.5 Media-Konzept

Beim Grundlayout muss vor allem auf die Umsetzung der Corporate Identity und des Corporate Designs der Hochschule für Technik, Wirtschaft und Kultur (FH) geachtet werden.

#### Logo

Die richtige Verwendung des Hochschullogos steht in der Anwendungsvorschrift vom 01.08.1999, welche auf den Grundlagen der "Handhabungsvorschrift zum Corporate design der HTWK Leipzig" vom 20.08.1996 und der Hausmitteilung "Verbessertes Logo der HTWK Leipzig" vom 19.05.1998 basiert. Zusammenfassend gilt, dass das Logo einfarbig - dunkelblau (Druckfarbe HKS 42) ist und einzeln oder in Komplettierung mit dem deutschen bzw. englischen Hochschulnamen angewendet werden kann. Dabei ist die Farbe des Namens schwarz und die Schrift OfficinaSanItcT<sup>16</sup>.

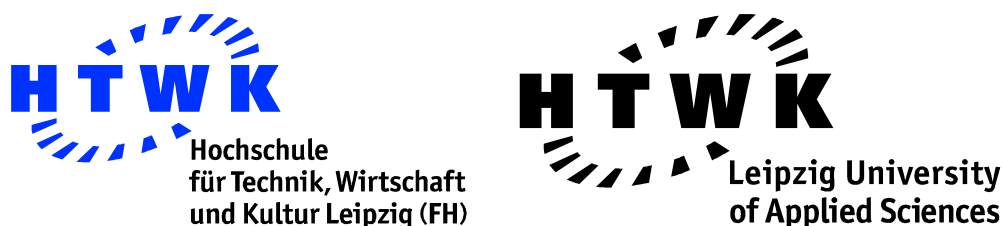


Abbildung 6-1 Beispiele für das Hochschullogo

#### Schrift<sup>17</sup>

Die Hausschrift der HTWK Leipzig ist Officina. Sie sollte immer verwendet werden. Ausnahmen sind umfangreiche Lesetexte, wobei Officina für die Überschrift, die Kopf-/Fußzeilen und den Vordruck verwendet werden soll. Als Kontrastschrift wird Times New Roman empfohlen.

#### Farben<sup>17</sup>

Die Beziehung zur Stadt stellt die HTWK Leipzig durch ihre Farben her. Bei der farblichen Gestaltung sollten die Farben dunkelblau (Druckfarbe HKS 42 - 100%) und gelb (Druckfarbe HKS 5 - 100%) dominieren.

<sup>16</sup> das T steht für normale Schrift

<sup>17</sup> Handhabungsvorschrift zum Corporate design der HTWK Leipzig, 20.08.1996

## Navigation

Die Navigation wird in vier Formen eingeteilt [vgl. Steinmetz 2000, S. 823]:

- gerichtet (Durchstöbern von Dokumenten mit einem Ziel)
- ungerichtet (Durchstöbern von Dokumenten ohne Ziel)
- systematisch (Suche nach bestimmtem Ziel)
- gesteuert (Folgen von vorgegebenen Pfaden, bspw. Guided Tours)

Das WCMS sollte alle Formen ermöglichen. Dabei darf die gerichtete und ungerichtete Navigation nicht zu tief und über einen Pfad zurückverfolgbar sein. Die systematische Navigation enthält Filter für genauere Suchergebnisse und die gesteuerte Navigation sollte interaktiv sein. Das heißt, dass der Nutzer anhand seiner Vorkenntnisse bzw. Probleme Schritte der Tour überspringen kann.

## Layout

Bei dem Layout geht es vor allem um das Zusammenspiel von Farben, Schrift, Inhalt und dessen Anordnung und die richtigen Visualisierungen. „Visualisierungen ... dienen der Abbildung eines Sachverhalts: Meistens soll dabei eine Vielfalt von Informationen in eine Form gebracht werden, so daß einem Betrachter entsprechende Interpretationen, Such-, und Vergleichsoperationen wesentlich erleichtert werden.“ [Steinmetz 2000, S. 776] Durch die korrekte Anwendung entsteht eine übersichtliche Webseite, die nicht mit Inhalten überlastet ist und der CI und dem CD der Hochschule entspricht. Eine genaue Evaluation von anderen Content Management Systemen (→ Kapitel 4.3 *Fremdsystemanschaffung oder Eigenentwicklung*) ist in diesem Fall von Nöten. Nur so kann eine übersichtliche und leicht bedienbare Oberfläche für die Bearbeiter des Systems geschaffen werden.

## Beispiel

Im folgenden Schritt habe ich eine Dummyseite (siehe Abbildung 6-2) entwickelt. Das Hochschullogo wurde für die Wiedererkennung der Seitenzugehörigkeit global sichtbar dargestellt. Die Schriftart kann leider nicht umgesetzt werden, da die meisten Benutzer diese nicht standardmäßig installiert haben. Ich habe mich für die serifenlose Schriftart Arial entschieden. Bei der Farbumsetzung habe ich Blau für den allgemeinen Bereich und Gelb für den Mitgliederbereich verwendet. Eine Mischung der beiden Farben, z. B. gelbe Balken auf blauem Hintergrund im Menübereich, lässt die Webseite nicht seriös erscheinen. Die allgemeine Navigation erfolgt wie bei den meisten Seiten im linken Bereich. Durch die Menüpunktiefe wurde eine Menü-/ Navigationsleiste im oberen

Bereich der Seite erstellt. Den Mitgliederbereich habe ich gesondert von der allgemeinen Navigation auf der rechten Seite erstellt. Diese klare Trennung ermöglicht eine leicht erkennbare Übersichtlichkeit. Das gesamte Layout wirkt durch seine Anordnung, die verwendeten Farben und die Schriftart seriös und übersichtlich.



Abbildung 6-2 Screenshot

### 6.3.6 Technisches Konzept

Wie im Kapitel 4.3 *Fremdsystemanschaffung oder Eigenentwicklung* schon erwähnt, habe ich mich für eine Eigenentwicklung entschieden. Die derzeitige Homepage des Fachbereiches ist durch ihren statischen Aufbau nicht erweiterbar bzw. nicht für eine Umstellung geschaffen. Eine Neuentwicklung ist daher unabdingbar. Für die Erstellung und den Betrieb des WCMS wird voraussichtlich folgende Hard- und Software benötigt:

Hardware	
benötigt	vorhanden
Server	Sun Fire V100 (Prozessor: UltraSPARC-III 500MHz; Speicher: 1 GByte)

Tabelle 6-6 Benötigte und vorhandene Hardware

Software	
benötigt	vorhanden
Microsoft Windows	MS Windows XP, MS Windows NT
Apache Webserver	Apache 2.0.47
MySQL-Datenbanksoftware	MySQL 4.0.14.
PHP	PHP 4.3.3.
Microsoft Office	MS Office XP
Adobe Acrobat	Adobe Acrobat Reader 6

**Tabelle 6-7** Benötigte und vorhandene Software

Alle Hard- und Softwarekomponenten sind am Fachbereich vorhanden. Die anfallenden Daten werden weitestgehend in einer Datenbank abgespeichert. Ein Datenbankschema und eine ausführliche Tabellenbeschreibung befinden sich im Anhang.

### 6.3.7 Deployment-Konzept

Das Konzept des Dynamic Publishing ist für die WCMS-Umsetzung eines Fachbereiches ungenügend. Die hohe Anzahl an Nutzern würde zu Performanceproblemen führen. Ebenso ist die Sicherheit durch Firewalls nicht gewährleistet. Aber auch ein separater Qualitätssicherungsserver, wie es beim Publishing-/ QA-/ Staging-Server der Fall ist, wird bei der Relevanz und Menge der Daten nicht benötigt. Der Einsatz eines Publishing-/ Staging-Servers ist daher der zweckmäßigste. Nach Absprache mit den Administratoren des Fachbereiches, kann dieses Prinzip softwaretechnisch auf einem Server realisiert werden. Somit wird nur ein Server benötigt.

### 6.3.8 Schulungskonzept

Die am Projekt beteiligten Studenten werden in den jeweiligen Lehrveranstaltungen informiert und geschult. Dabei lernen sie die Grundlagen und den Aufbau des WCMS, wie man es benutzt, pflegt und es erweitert. Eine Einarbeitung in das fertige System erfolgt für zwei Personengruppen. Die erste und wichtigste Gruppe besteht aus dem Personal, welches verwaltende Tätigkeiten im WCMS besitzt. Dazu zähle ich die folgenden Rollen: Content Manager, Production Manager, Business Analysts, Metator, CMS Administrator und die technischen Administratoren des Systems. Sie werden durch das Erstellungspersonal und

deren angefertigte Dokumentationen und Handouts (deutsch: Handzettel) mit den wichtigsten Fragen und Antworten geschult. Die zweite Personengruppe besteht aus den Template and CMS Developer, Test Engineer, Publication Staff, Content Processing Staff und dem speziellen Content Creation Staff. Bei der Schulung wird der Sinn und die Handhabung des Systems erklärt. Dabei wird besonderer Wert auf das Erstellen, Bearbeiten, Veröffentlichen, Archivieren und Löschen von Inhalten gelegt. Für alle anderen Nutzer sollte das System für jede Tätigkeit selbsterklärend sein bzw. die intern umgesetzte Hilfe weiterhelfen.

### **6.3.9 Betriebskonzept**

Falls es zu Systemausfällen oder -fehlern kommt, wird der Systemadministrator des Fachbereichs darüber informiert. Bei Software- und inhaltlichen Fehlern wird man über ein Hilfeformular des WCMS die verantwortlichen Personen kontaktieren können. Zu ihnen zähle ich die Rolle des Content Managers, welcher als Feedback-Agent agiert.

## **6.4 Programmierung**

Im ersten Schritt wird das Datenbankschema (→ *Anhang A*) umgesetzt. Nach der Erstellung der Tabellen, Constrains und Trigger wird das Basissystem programmiert. Es beinhaltet Grundlagen, damit Templates, Funktionen und Inhalte nach dem Rechtekonzept erstellt und eingepasst werden können. Die Programmierung erfolgt im Wesentlichen anhand des Inhaltlichen Konzeptes (→ Kapitel 6.3.2 *Inhaltliches Konzept*). Ebenfalls werden administrative Funktionen für die Erstellung und Zuweisung von Rollen und Rechte angefertigt. Diese Prozesse werden ungefähr 250 Mannstunden dauern. Danach werden die Templates und Funktionen erstellt und zusammengeführt. Im Anschluss wird das System für verschiedene Testphasen mit Daten gefüllt. Die Befüllung sollte über die Benutzeroberfläche und nicht über direktes Einfügen in die Datenbank erfolgen. Für diese Prozesse sind ca. 300 Mannstunden einzurechnen. In allen Phasen der Erstellung ist eine ausführliche Dokumentation des Quellcodes und der Entwicklung für die spätere Erweiterung und Anlernung anderer Fachkräfte erforderlich.



## 6.5 Testphase

Als erstes prüfen die Programmierer und Designer ihre Arbeit. Sie schauen unter anderem, ob die Funktionen arbeiten und Performanter programmiert werden können, ob das WCMS auf verschiedenen Browsern in gleicher Weise dargestellt wird und ob die Anordnung der Templates die Übersichtlichkeit garantiert. Der Test der Browserunabhängigkeit muss auf den meistgenutzten Browsern geschehen. Dazu zählt in erster Linie der Microsoft Internet Explorer, welcher von 90,8%<sup>18</sup> der Internetsurfer genutzt wird. Trotzdem muss auch für den Netscape Navigator bzw. Mozilla (7,7%)<sup>18</sup> und Opera das System optimiert werden. Mit der Hilfe der Programmierer können Worst-Cases simuliert werden, um Funktionen und Fehlerausgaben zu untersuchen. Für einen unabhängigen Usability-Test werden Betatester (Studenten, Mitarbeiter, Professoren) das erstellte WCMS prüfen. Sie werden dabei nicht den Quelltext nach Fehlern durchsuchen, sondern das System durch Beispiele auf der Anwenderseite testen. Dabei achten sie auf Navigationsfreundlichkeit, Übersichtlichkeit, korrekte Resultate von Funktionen und die richtige Umsetzung des Rechtekonzeptes. Bei der Überprüfung der Rechte werden die unterschiedlichen Logins und Sichtweisen der Gruppen und Rollen getestet. Nach der Prüfung werden alle Betatester befragt. Das kann durch Interviews oder auch Fragebögen geschehen. [Werner et al. 1995]

Kriterien in der Testphase [vgl. Zschau et al. 2002, S. 304-312]:

- sind Möglichkeiten für Aktualisierungen gegeben
- ist die Mehrsprachenfähigkeit konsequent umgesetzt
- werden die vorgegebenen Medienformate vom System akzeptiert und umgewandelt
- bleibt das Layout erhalten, wenn längere Texte eingefügt werden oder die Seitengröße verändert wird
- ist eine einfache Navigation gewährleistet
- werden die gewünschten Suchergebnisse gefunden und übersichtlich dargestellt
- sind Formulare logisch aufgebaut
- haben alle Seiten kurze Ladezeiten
- ist das System Skalierbar
- treten Fehler auf

---

<sup>18</sup> Quelle: [N.N. 2003b]

- existieren ausführliche Dokumentationen
- ist das System vor Hackerangriffen geschützt
- erfolgt eine sichere Datenhaltung bzw. sicherer Datentransfer
- sind alle Webseiten browserunabhängig
- existiert ein übersichtliches und fehlerfreies Layout

## 6.6 Abnahme bzw. Freigabe

Nach ausführlichen Prüfungen wird das System dem Dekan vorgestellt. Durch die im Briefing aufgeführten Schwerpunkte ist eine vereinfachte Kontrolle für ihn möglich. An dieser Stelle ist die Umsetzung seiner Verbesserungsvorschläge relativ einfach, da noch kein Inhalt bzw. keine Testdaten gespeichert sind. Erfolgt die Abnahme fehlerfrei kann das System für den Onlinebetrieb freigegeben werden.

## 6.7 Onlinebetrieb

Nach der Freigabe werden alle Testdaten aus dem System gelöscht und mit Grunddaten gefüllt. Dazu gehören vorerst die statischen Inhalte wie Fachbereichsinformationen, Ämter, Gebäude und Labore, Technik, Hilfe zum Studium und Studiengänge. Die Inhaltserstellung wird ungefähr einen Monat dauern. Danach wird das System ca. drei Monate intensiv überwacht. Treten Fehler auf, kommt es zu Serverüberlastungen und wie ist die Resonanz bei den Benutzern? Diese und andere Fragen werden in dieser Zeit intensiv überdacht, beraten und gelöst, damit das System einen Grundzustand erreicht und dieser gefestigt wird. Es kann im Anschluss eine nochmalige Testphase (→ Kapitel 6.5 *Testphase*) eingeleitet werden.

## 7 Programmierbeispiel

### 7.1 Vorwort

Für die Veranschaulichung des WCMS für den Fachbereich IMN habe ich eine Dummyseite entwickelt. Sie soll nicht nur die Wirkungsweise des Gesamtlayouts zeigen, sondern auch die Funktionsweise des konzipierten WCMS demonstrieren. Im speziellen habe ich mich auf die Umsetzung der folgenden Punkte konzentriert:

- vollständige Datenbank
- serverseitige Indexdatei (verantwortlich für Datenbankverbindung und Zusammenführung von Templates, Dateien, Daten, Modulen, Styles und Farben)
- Indextemplate (bestehend aus Kopfteil, Menübereich, Untermenübereich, Subtemplatebereich, Login- bzw. Mitgliederbereich)
- Strukturmodule (Anzeige der Menü- und Untermenüpunkte)
- Loginmodul (Authentifizierung der Nutzer)
- Berechtigungsmodul (Bearbeitungsrecht der Nutzer auf Inhaltselemente)
- Sprachmodul (Wechsel der Sprache)
- Bearbeitungsmodul (Inhaltsbearbeitung der Templates)
- vereinfachtes Rechtevergabemplate und -module (Rechtezuweisung für Rollen und Strukturpunkte)
- Personentemplate (Einleitungsseite der Personen)
- Studentenlistemplate und -module (Anzeige aller Studenten in einer Liste)
- Studentendetailstemplate und -module (Anzeige aller Details zu einem Studenten)

### 7.2 Aufbau des Grundsystems

#### Datenbank

Das Datenbankschema wurde mit Microsoft Visio Professional 2002 erstellt und befindet sich im → *Anhang A*. Darin sind das

- Rightmanagement
- Personmanagement
- Layoutmanagement
- Modulemanagement
- Medialibrary
- Contentmanagement

farblich veranschaulicht. Es ist zu erkennen, dass es sehr allgemeingültig erstellt wurde. Dadurch ist eine individuelle Seitenerstellung möglich, ohne das Datenbankschema zu modifizieren. Die einzelnen Tabellen werden im → *Anhang B* beschrieben. Dabei werden die Aufgaben und die einzelnen Datenfelder mit ihren Bedeutungen aufgelistet.

### **Initialdaten**

Nachdem die Datenbank angelegt wurde, werden die Initialdaten eingefügt. Für dieses Beispiel habe ich alle Farben, einige Styles, die komplette Seitenstruktur (→ Kapitel 6.3.2 *Inhaltliches Konzept*), alle Rechte und Rollen (→ Kapitel 6.3.4 *Redaktions- und Rechtekonzept*), einige Testpersonen und die Verbindungen untereinander in die Datenbank eingefügt.

### **Indexseite**

Die serverseitige Indexdatei ist für die Datenbankverbindung und die Zusammenführung von Templates, Dateien, Daten, Modulen, Styles und Farben zu einer Webseite verantwortlich. Im Einzelnen bedeutet das, dass nachdem die Verbindung hergestellt wurde, die einzelnen Styles aus der Datenbank ausgelesen werden. Danach wird das Indextemplate geladen und das entsprechende Subtemplate darin eingefügt. Im nächsten Schritt werden die einzelnen Datenvariablen dynamisch generiert und mit den jeweils übersetzten Daten der gewählten Sprache belegt. Nachdem die Platzhalter der Module, Dateien und Farben in den beiden Templates durch den entsprechenden Quellcode ersetzt wurden, wird die Webseite für die Anzeige generiert.

### **Indextemplate**

Das Indextemplate beinhaltet die Platzhalter für alles, was man zu jeder Zeit auf der Webseite sehen soll. Dazu gehören der Kopfteil, der Menübereich, der Untermenübereich, der Subtemplatebereich und der Login- bzw. Mitgliederbereich.

## Templates

Ein Template besteht aus Tabellen und Styles, welche die Contentbauteile auf der Webseite positionieren. Jedem Strukturpunkt wird direkt ein Template zugeordnet. Das folgende Beispiel zeigt den charakteristischen Aufbau: eine Tabelle, welche die einzelnen Dateien, Daten, Module, Styles und Farben beinhaltet.

Beispiel eines Templates:

```
<table>
  <tr bgcolor="gelb">
    <td width="1%">file_spacer10_20</td>
    <td width="49%" align="right">
      <span class="style_text">data_willkommen</span>
      modul_hilfebutton
    </td>
    <td width="1%">spacer10_20</td>
    <td width="49%" align="left">file_imnbild</td>
  </tr>
</table>
```

## Dateien

Die einzelnen Verknüpfungen zu allen Dateien werden für eine schnellere Verarbeitung kategorisiert und medienneutral abgespeichert. Alle Eigenschaften der einzelnen Dateien werden in der separaten Tabelle *filefact* gespeichert und mit der jeweiligen Templateverknüpfung der Tabelle *templatefile* verbunden. Diese Methode ist vorteilhaft für die Mehrfachverwendbarkeit der einzelnen Dateien in verschiedenen Kontexten. Zum Beispiel kann dadurch ein 1x1-Pixel durch eine Höhen- und Breitenangabe als Spacer (Platzhalter) oder ein Foto mit verschiedenen Bildunterschriften genutzt werden.

## Daten

Alle textbasierten Daten, wie beispielsweise Überschriften und Menüpunkte, werden medienneutral und multilingual in der Tabelle *data* abgespeichert. Über die Tabelle *templatedatatype* werden die Templates mit den Datentypen verbunden. Mittels dieser Methode wird eine individualisierte Rechtevergabe realisiert. Zum Beispiel können dadurch Rechte auf alle Überschriften eines Templates gegeben werden. Mit Hilfe der Tabelle *dataassign* werden die Daten (z.B.: Hilfeseite, Paul, Hauptstraße) den Datentypen (z.B.: Überschrift, Name, Straße etc.) zugeordnet.

## Module

Die Module sind für die Funktionalität der Webseiten verantwortlich. Sie enthalten Quellcode, der beispielsweise Formulare erzeugt, die Seitenstruktur ausliest, für die Rechteverteilung zuständig ist oder Schnittstellen zu Scripten beinhaltet. Jedes Modul wird kategorisiert abgespeichert.

## Allgemein

Alle Datensätze der Tabellen *person*, *structure*, *template*, *file*, *data* und *module* enthalten einen Zeitstempel für den Erstellungszeitpunkt (*timest*), für die Aktivierung (*activate*) und für die virtuelle Löschung (*deleted*). Zusätzlich existiert für das Check-in/ Check-out-Verfahren bei der Bearbeitung ein Flag (*checkout*). Für die genaue Zuordnung eines Contents zu seinem Ersteller wird ebenfalls die Identifikationsnummer der Person (*personid*) zu jedem Datensatz der oben genannten Tabellen (nicht *person*) abgespeichert.

## 7.3 Beispiel

Für die Veranschaulichung eines Inhaltsbereiches des WCMSs habe ich den Personenbereich (speziell Studenten) teilweise umgesetzt. Die realisierten Inhalte mit den verwendeten Modulen sind in Tabelle 7-1 aufgelistet.

Rubrik-ID	Rubrik	realisierte Inhalte	Module
[103]	Personen	Einleitungstext	-
[10303]	Studenten	Tabelle mit Nachname, Vorname und Matrikel	listeallerstudenten (Datenbankabfrage der jeweils 3 Daten pro Person)
[1030301]	Details	Bild und alle Daten einer Person	personendetail (Datenbankabfrage des Bildes und aller Daten) bearbeiten (Buttonanzeige, falls Rolle über edit-Recht verfügt)

**Tabelle 7-1** Inhaltsrealisierung am Beispiel Personen

## 7.4 Ausblick

Das Grundsystem mit dem Beispiel der Studenten soll einen kleinen Ausschnitt eines kompletten WCMSs zeigen. Leider war die vollständige Umsetzung im Rahmen meiner Diplomarbeit zeitlich nicht möglich. Daher fehlen auch die restlichen Inhaltsbereiche (→ Kapitel 6.3.2 *Inhaltliches Konzept*) mit den entsprechenden Dateien, Daten, Modulen und Styles. Anhand der Komponenten eines WCMSs (→ Kapitel 3 *Komponenten und Aufbau eines WCMSs*) führe ich kurz auf, welche Systemteile umgesetzt bzw. welche noch verwirklicht werden müssen.

### Assetmanagement

#### Abspeicherung des Inhaltes

Die Datenbank wurde vollständig und allgemeingültig für die Speicherung aller Inhalte erstellt. Eine Veränderung bzw. Erweiterung ist nicht nötig.

#### Bearbeiten

Die dezentralisierte Bearbeitung der Inhalte wird durch den Browser-Client ermöglicht. Für diese Variante werden nur die fertigen Templates benötigt. Die neuen Inhalte werden dann direkt an der dafür vorgesehenen Stelle eingefügt. Die Verwaltungsoberflächen für einen Teil des WCMS-Personals sind noch nicht vorhanden. Zum Beispiel:

- Template-Management (Erstellung, Bearbeitung und Löschung von Templates mit Überblick über Module, Dateien, Daten, Styles und Farben mit Vorschaufunktion)
- Modul-Management (Erstellung, Bearbeitung und Löschung von Modulquellcode)
- Datei-Management (Erstellung, Bearbeitung und Löschung von Dateien mit Vorschaufunktion)

#### Automatisierungen

Es wurden keine Content-Authoring-Methoden erstellt. Die Möglichkeit besteht durch die Realisierung mittels Modulen.

### Versionierung

Die Versionierung ist mit Hilfe von mehreren Zeitstempeln für die Erstellung, Aktivierung und Löschung ermöglicht worden.

### Länder- und Sprachabhängigkeit

Die Multilingualität der Webseite wurde vollständig umgesetzt. Alle Daten können mit Hilfe der Datenbank übersetzt gespeichert werden. Das Modul *sprache* wandelt die Daten in die jeweils ausgewählte Sprache.

## **Workflowkomponente**

### Workflow

Die Grundlagen für die Komplettierung sind geschaffen. In verschiedenen Modulen werden Funktionen für das Erstellen, Bearbeiten, Löschen und Archivieren der Inhalte erstellt und durch das interne Messagesystem an die jeweils berechtigte Rolle weitergereicht. Zum Beispiel soll ein neues Gebäude angelegt werden. Bei alle Rollen, die das Recht zum erstellen von Gebäuden haben, erscheint ein Erstellbutton. Nach der Erstellung werden die Rollen mit dem edit-Recht abgefragt und an diese Personen eine Nachricht geschickt, dass eine Bearbeitung anliegt. Anschließend wird das neue Gebäude eventuell bearbeitet und mit Hilfe des Messagesystems an die Freigabeberechtigten Personen weitergeleitet, die den neuen Inhalt publizieren.

### Rechtekonzept

Jede Person kann Rollen zugeordnet werden. Durch die Verbindung der Rechte, IDs der Contentbauteile und der Rollen ist eine individualisierte Rechtevergabe gewährleistet.

### Geschütztes Editieren

Damit es zu keinen Datenkollisionen kommt, wurde allen Contentbausteinen ein Flag für das Check-in/ Check-out-Verfahren hinzugefügt. Sowie ein Nutzer einen Inhalt bearbeitet, wird das Flag gesetzt und allen anderen Nutzern der Zugriff verwehrt.



### Werkzeuge des Workflows

Benachrichtigungen der Nutzer sind über ein internes Messagesystem möglich. Die Voraussetzungen dafür sind mit der Datenbank gegeben. Auch Statusanzeigen, Filter, Aufgabenverteilungen und -listen, Protokollierungen, Previews und Vertreterregeln sind vorgesehen.

### Freigabeverfahren

Die Freigabeverfahren sind später mit Hilfe des internen Messagesystem und den Modulen möglich.

### **Zugriffs- und Benutzerverwaltung**

Jeder Nutzer wird Gruppen und Rollen zugeordnet, die wiederum Rechte auf bestimmte Bereiche besitzen. Meine Umsetzung bezieht sich auf die Zugriffsverwaltung der einzelnen Strukturpunkte der Webseite. Die Rechteverteilung für Daten, Module und Dateien wurde nur datenbanktechnisch realisiert. Mit Hilfe von Modulen wird die Anwendung der Rechteverteilung verwirklicht. Für eine Veranschaulichung habe ich das Modul zur Bearbeitungsberechtigung von Daten programmiert.

### **Im- und Exportschnittstellen für Content**

Schnittstellen zu Scripten, Datenbanken etc. sind durch die Programmierung von Modulen gewährleistet.

### **Programmierschnittstellen**

Schnittstellen für funktionale Erweiterungen wurden nicht umgesetzt. Prinzipiell ist die Erstellung von Programmierschnittstellen möglich.

## 8 Resümee

Ein Content Management System bietet viele Möglichkeiten, Inhalte effektiv zu verwalten. Durch die Trennung von Inhalt, Gestaltung und Funktionalität werden eine einheitliche CI und ein einheitliches CD, Multilingualität, problemlose Designveränderungen, Mehrfachverwendbarkeit der Inhalte, effektive Mitarbeiterbeziehung und Cross-Media-Publishing gewährleistet. Der integrierte Freigabezyklus garantiert die Aktualität und Qualität der erstellten Inhalte. Durch Authoring-Möglichkeiten, wie zum Beispiel Linkkontrolle, Benachrichtigungen, Versionierung und Sitemap-Generierungen, werden die Mitarbeiter entlastet. Durch diese ausschlaggebenden Argumente habe ich festgestellt, dass der Einsatz eines Web Content Management Systems für den Fachbereich Informatik, Mathematik und Naturwissenschaften der HTWK Leipzig sinnvoll erscheint. „Content Management-Projekte sind komplexe Aufgaben, die solide durchdacht werden müssen. Käufliche Wunderwaffen gibt es hier nicht.“ [Knobloch 2003, S. 37] Daher habe ich mich auf eine Eigenentwicklung konzentriert und anhand der Bedürfnisse der Zielgruppen ein Konzept entwickelt, welches den Anforderungen an eine Fachbereichsseite in vollem Maße gerecht wird. Ich klärte dabei unter anderem die Kompetenzen, den Inhalt, die Gestaltung, die Verwaltung und die Verwirklichung des Konzeptes.

Die programmiertechnische Umsetzung eines Beispiel-WCMS hat gezeigt, dass eine Verwirklichung für den Fachbereich IMN der HTWK Leipzig relativ einfach und schnell erfolgen kann. Trotz des hohen Initialaufwandes meinerseits sollten intensive Evaluationen von bestehenden Content Management Systemen durchgeführt werden. Eine vollständige Umsetzung des gesamten WCMSs war im zeitlich begrenzten Rahmen meiner Diplomarbeit nicht möglich.

Während meiner Anfertigung sprach ich mit vielen Professoren und Mitarbeitern, die sich für eine Realisierung dieses Projektes einsetzen möchten. Bei einer Verwirklichung des WCMSs würde ausreichend Lehrmaterial für verschiedene Lehrveranstaltungen entstehen. Aber auch ein Mehraufwand für Mitarbeiter und Professoren, welche derzeit nicht am Publishing-Prozess der aktuellen Webseite beteiligt waren. Daher muss eine Akzeptanz und das Verständnis mit dem Umgang bei allen Beteiligten erreicht werden.

## Akronyme

<b>A</b>	API	Application Programming Interface
<b>C</b>	CD	Corporate Design
	CI	Corporate Identity
	CMS	Content Management System
<b>D</b>	DBMS	DataBase Management System
	DMS	Document Management System
	DTD	Document Type Definition
	DTML	Document Template Markup Language
<b>E</b>	EAI	Enterprise Applications Integration
<b>G</b>	GUI	Graphical User Interface
<b>H</b>	HTWK	Hochschule für Technik, Wirtschaft und Kultur
<b>I</b>	ICE	Information and Content Exchange Protocol
	IMN	Informatik, Mathematik und Naturwissenschaften
	IT	Information Technology
<b>J</b>	JDBC	Java DataBase Connectivity
<b>K</b>	KMS	Knowledge Management System
<b>L</b>	LDAP	Lightweight Directory Application Protocol
	LPE	Live Production Environment
<b>M</b>	MAMS	Media Asset Management System
	MIME	Multipurpose Internet Mail Extensions

<b>N</b>	NIS	Network Information System
<b>O</b>	ODBC	Open DataBase Connectivity
<b>P</b>	PDA	Personal Digital Assistant
	PDF	Portable Document Format
	PHP	Personal Home Page bzw. PHP HyperText Preprocessor
<b>S</b>	SGML	Standard Generalized Markup Language
	SMS	Short Message Service
<b>U</b>	URL	Uniform Resource Locator
<b>W</b>	WAP	Wireless Application Protocol
	WCMS	Web Content Management System
	WML	Wireless Markup Language
	WYSIWYG	What You See Is What You Get
<b>X</b>	XML	eXtensible Markup Language
	XSL	eXtensible Stylesheet Language
<b>Z</b>	ZODB	Zope Object Database
	ZOPE	Z Object Publishing Environment

## Glossar

**Asset, digitales:** Assets sind kleine Einheiten bzw. Inhaltsbausteine wie zum Beispiel Text, Bilder, Videos und  $\rightarrow$ Module. Aus ihnen wird der Content der Seite zusammengesetzt.

**Authoring:** Die automatisierte Darstellung und Auswahl von Inhalten. Zu Authoring-Methoden zählen beispielsweise die Sitemapgenerierung und das Link-Checking.

**Content Repository:** Die Ablage für den Inhalt und den dazugehörigen  $\rightarrow$ Metadaten. Technologisch setzt dieses auf ein Dateisystem, einer Datenbank oder beides auf.

**Content-Syndication:** Eine Inhaltsbereitstellung durch Fremdanbietern.

**Cross-Media-Publishing:** Die Verwendung von Inhalten aus einem Informationsbestand für verschiedene Ausgabemedien wie zum Beispiel CD-ROM, Handy und Printmedien.

**Guided Tours:** Eine Hilfefunktion, welche den Nutzer entlang eines Pfades zum Ziel führt.

**LDAP:** Das Lightweight Directory Application Protocol ist ein auf X.500 aufbauendes Protokoll. Es ermöglicht ein „Single-Sign-On“ (einmaliges anmelden) durch die zentrale Speicherung der Nutzerdaten in einem Unternehmen.

**Metadaten:** Auch Metainformationen: nennt man die zusätzliche Informationen über den Inhalt und die Beschaffenheit eines  $\rightarrow$ Assets.

**Modul:** Eine kleine Anwendung (Applikation), welche in  $\rightarrow$ Templates eingebunden werden. Beispiele hierfür sind Chat, Forum und die Suche.

**Outsourcing:** Die Übertragung von Aufgaben an externe Firmen oder Dienstleister.

**Redesign:** Eine Veränderung der bestehenden Gestaltung nennt man Redesign.

**Related Links:** Links, welche zu anderen Links themenverwandt sind.

**reuse:** Die Wieder- bzw. Mehrfachverwendbarkeit von Inhalten.

**Sitemap:** Bezeichnet die Übersicht über die Struktur der kompletten Webseite.

**Spider:** Auch Robot: Automatische Hilfsprogramme, welche Webserver nach Inhalten durchsuchen und diese den Suchmaschinen zuleitet, damit sie indiziert werden können.

**Template:** Auch Vorlage: Ein Gerüst von Platzhaltern für die Zusammenführung und Darstellung der einzelnen  $\rightarrow$ Assets.

**Time-to-Web:** Zeit, die der Inhalt von der Konzeption bis zur Online-Veröffentlichung benötigt. Sie bestimmt die Aktualisierungsfrequenz der Webseite.

**Track:**  $\rightarrow$  Modul

**Workflow:** Die Arbeitsschritte, deren zeitliche Einteilung und die Verantwortlichkeiten, welche bei der Erstellung bzw. Pflege einer Webseite benötigt werden.

**WYSIWYG-Editor:** Editoren, welche bei der Programmierung bzw. das Scripten von Webinhalten die Darstellung im Web vorab simulieren. Daher erübrigt sich die Arbeit im Quellcode.

## Abbildungs- und Tabellenverzeichnis

Abbildung 2-1 Grundprinzip eines Web Content Management Systems.....	11
Abbildung 2-2 Content Life Cycle [vgl. Zschau et al. 2002, S. 56].....	13
Abbildung 2-3 Ursachen für Webseitenmeidung [Versteegen 2001, S. 20] .....	17
Abbildung 2-4 Kostenentwicklung einer Webseite [vgl. Zschau et al. 2002, S. 79] .....	17
Abbildung 3-1 Komponenten eines WCMS.....	19
Abbildung 3-2 Lineares Freigabeverfahren.....	27
Abbildung 3-3 Paralleles Freigabeverfahren.....	27
Abbildung 3-4 Prinzip Dynamic Publishing [vgl. Zschau et al. 2002, S. 244] .....	30
Abbildung 3-5 Prinzip Publishing-/ Staging-Server [vgl. Zschau et al. 2002, S. 246] .....	31
Abbildung 3-6 Prinzip Publishing-/ QA-/ Staging-Server [vgl. Zschau et al. 2002, S. 247].....	31
Abbildung 3-7 Basiskomponenten eines Applikationsservers [vgl. Busch 2000] .....	32
Abbildung 6-1 Beispiele für das Hochschullogo.....	76
Abbildung 6-2 Screenshot .....	78
Tabelle 5-1 CMS-Jobs und ihre Einsatzzeiten [Boiko 2001, S. 178 f.] .....	47
Tabelle 6-1 Navigationsbaum der IMN-Seite .....	66
Tabelle 6-2 Anlieferungsformate.....	66
Tabelle 6-3 An- und Auslieferungsarten .....	67
Tabelle 6-4 Gruppen- und Rollenbezeichnungen.....	68
Tabelle 6-5 Verhältnistabelle von Kategorie, Content und Rechte .....	75
Tabelle 6-6 Benötigte und vorhandene Hardware.....	78
Tabelle 6-7 Benötigte und vorhandene Software .....	79
Tabelle 7-1 Inhaltsrealisierung am Beispiel Personen .....	86

## Literaturverzeichnis

### [Boiko 2001]

Boiko, Bob, Content Management Bible, Indianapolis, Indiana: Wiley Publishing, Inc., 2001

### [Busch 2000]

Busch, D., Applikationsserver für die Webentwicklung, in: Internet Professionell, Nr. 12, 2000

### [Christ 2003]

Christ, Oliver, Content-Management in der Praxis – Erfolgreicher Aufbau und Betrieb unternehmensweiter Portale, Berlin, 2003

### [Christian 2003]

**Wolff, Christian**, Web Services im e-Learning und e-Publishing, Leipziger Informatik Tage '03; 24.09.2003

### [Güttler 2003]

Güttler, Thomas, Zope – Ein Web Application Server, 2003, <http://www.thomas-guettler.de/vortraege/zope/einfuehrung.html#5>, 15.09.2003

### [Schwarze 2001]

Schwarze, Frank, In der durchdachten Lokalisierung liegt der Schlüssel zum erfolgreichen internationalen Webauftritt, in: swldevelopment, Mai/Juni 2001, S.52-54

### [Knobloch 2003]

Knobloch, Manfred, Anforderungen an Content Management, in: Javamagazin, 5.03, S. 32-51



**[Krüger/ Kopp 2002]**

Krüger, Jörg D., Kopp, Matthias, Web Content managen: professioneller Einsatz von Content-Management-Sytemen, München, 2002

**[N.N. 2002]**

N.N., Zope: Übersicht, 2002, <http://www.easyleading.org/zope-folien/satz1/>, 24.10.2002

**[N.N. 2003a]**

N.N., Applikationsserver-Architektur, [http://www.brockhaus-ag.de/bag\\_live/bag/De/Unternehmen/knowHow/A-F/Applikationsserver/Applikationsserver3.jsp](http://www.brockhaus-ag.de/bag_live/bag/De/Unternehmen/knowHow/A-F/Applikationsserver/Applikationsserver3.jsp), 09.10.2003

**[N.N. 2003b]**

W3B Browserwatch, <http://www.fittkaumaass.de>, 09.10.2003

**[Reitmaier 2003]**

Reitmaier, A., Hallo CMS: Content Management-Systeme auf Java- und Zope-Basis, in: Linux Enterprise, 3.03, S. 27-31

**[Steinmetz 2000]**

Steinmetz, R., Multimedia-Technologie: Grundlagen, Komponenten und Systeme, 3. Aufl., Berlin; Heidelberg; New York, 2000

**[Stepken 1999]**

Stepken, Guido, MySQL Datenbankhandbuch, 1999, <http://www.little-idiot.de/mysql/>, 12.11.2003

**[Versteegen 2001]**

Versteegen, Gerhard, Web Content Management – Das Web kriegt Hilfe, in: swldevelopment, Mai/Juni 2001, S.20-24

**[Weber 2002]**

Weber, Michael, Web Content Management - Studie: Aktuelle, [http://www.contentmanager.de/magazin/artikel\\_212-print\\_web\\_content\\_management\\_studie.html](http://www.contentmanager.de/magazin/artikel_212-print_web_content_management_studie.html), 24.03.2003

**[Werner et al. 1995]**

Werner, Dieter, u. a., Taschenbuch der Informatik, 2. Auflage, Leipzig, 1995

**[Wilhelm 2001]**

Wilhelm, Stephan, Content Management in der deutschen Wirtschaft, 2001, [http://www.contentmanager.de/magazin/artikel\\_77-print\\_content\\_management\\_in\\_der\\_deutschen\\_wirtschaft.html](http://www.contentmanager.de/magazin/artikel_77-print_content_management_in_der_deutschen_wirtschaft.html), 03.06.2003

**[Wilhelm 2002]**

Wilhelm, Stephan, Qualität durch Information – Content Management kann bis ans Fließband reichen, 2002, [http://www.contentmanager.de/magazin/artikel\\_250-print\\_qualitaet\\_durch\\_information.html](http://www.contentmanager.de/magazin/artikel_250-print_qualitaet_durch_information.html), 03.06.2003

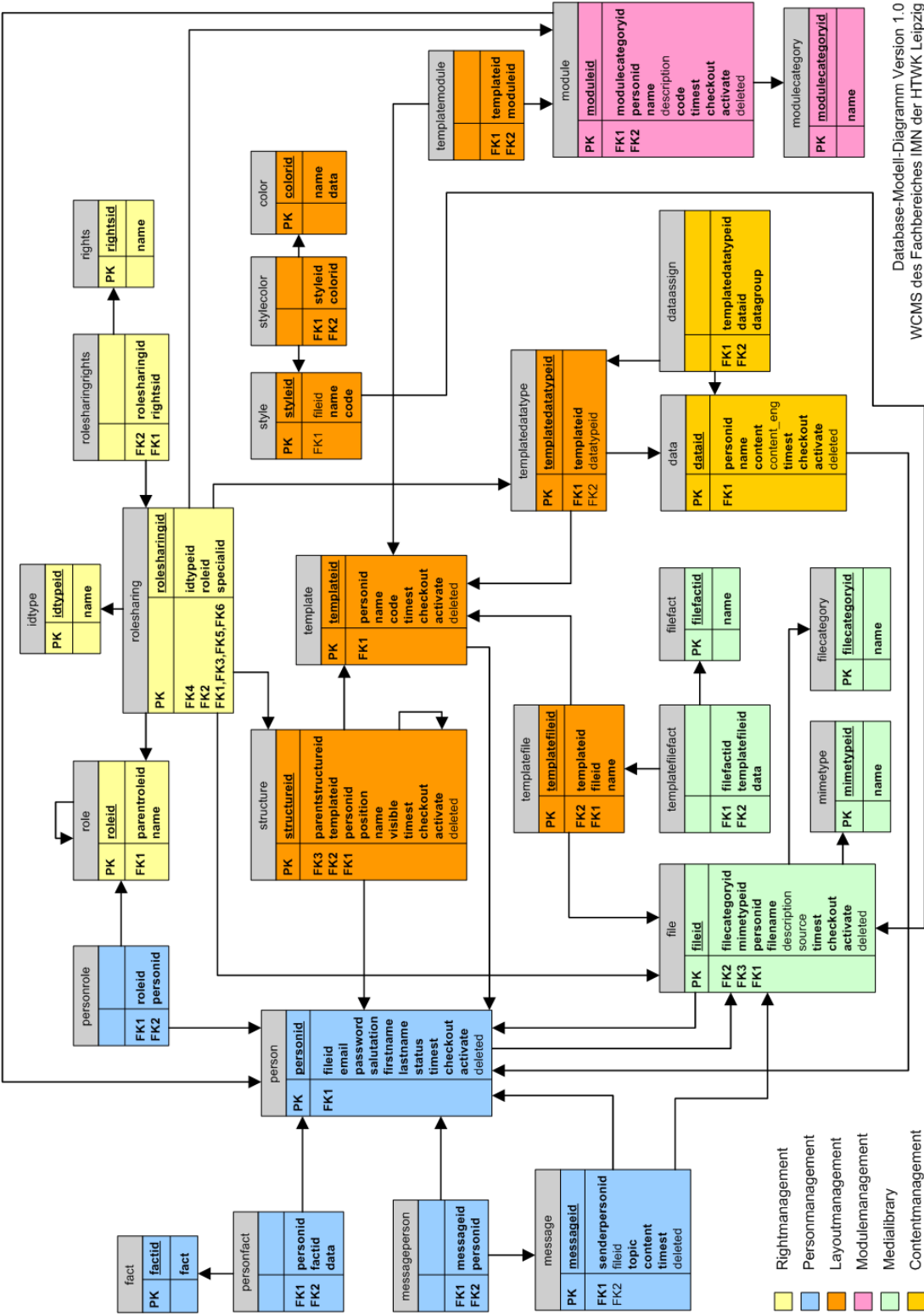
**[Wolff 2003]**

Wolff, Markus, Power-Sklaven: Vier PHP-basierte Content Management-Systeme im Überblick, in: Linux Enterprise, 3.03, S. 32-41

**[Zschau et al. 2002]**







Zschau, O., Traub, D., Zahradka, R., Web Content Management – Websites professionell planen und betreiben, 2. Aufl., Bonn, 2002

# Anhang A: Datenbankschema



Database-Modell-Diagramm Version 1.0  
WCMS des Fachbereiches IMN der HTWK Leipzig

## Anhang B: Tabellenbeschreibung

	Rightmanagement
	Personmanagement
	Layoutmanagement
	Modulemanagement
	Medialibrary
	Contentmanagement

<b>role</b>	
Hier werden die einzelnen Rollen des Systems abgebildet.	
roleid	Identifikationsnummer der Rolle
parentroleid	übergeordnete Rollen-ID
name	Name der Rolle (z.B. Dekan)

<b>idtype</b>	
ID-Typ für die spezialisierte Rollenvergabe	
idtypeid	Identifikationsnummer des ID-Types
name	Name (structurid, moduleid, fileid, templatedatatypeid)

<b>rolesharingrights</b>	
Verbindungstabelle zwischen rolesharing und rights; Vergabe von mehreren Rechten an eine Rolle	
rolesharingid	ID der Rollenvergabe
rightsid	ID des Rechtes

<b>personrole</b>	
Verbindungstabelle zwischen person und role; eine Person kann verschiedene Rollen besitzen	
roleid	ID der Rolle
personid	ID der Person

<b>fact</b>	
Fakten zur Person (z.B. Professur, Telefon,...)	
factid	Identifikationsnummer des Faktes
name	Name des Faktes

<b>personfact</b>	
Verbindungstabelle zwischen person und fact; darin werden die einzelnen Daten der Fakten gespeichert	
personid	ID der Person
factid	ID des Faktes
data	Daten

<b>rights</b>	
In dieser Tabelle werden alle Rechte erfasst.	
rightsid	Identifikationsnummer des Rechts
name	Name des Rechts (z.B. create)

<b>rolesharing</b>	
Verbindungstabelle zwischen role und structure, module, file und data; Rollen mit Rechten an Struktur, Modul, Datei und Daten binden	
rolesharingid	Identifikationsnummer der Rollenverteilung
idtypeid	ID des speziellen Rollennamens
roleid	ID der Rolle
specialid	spezielle ID des idtypes

<b>person</b>	
enthält die grundlegenden (nicht optionalen) Daten zu jeder Person	
personid	Identifikationsnummer der Person
fileid	ID des Fotos
email	E-Mail-Adresse
password	Passwort
salutation	Anrede (Herr, Frau)
firstname	Vorname
lastname	Nachname
status	Personenstatus (Student, Prof,...)
timest	Zeitstempel der Erstellung
checkout	Checkfeld für Bearbeitung
activate	Aktivierungsfeld
deleted	Zeitstempel der Löschung

<b>message</b>	
Die message-Tabelle beinhaltet alle internen Nachrichten.	
messageid	Identifikationsnummer der Nachricht
senderpersonid	ID der Absenderperson
fileid	ID einer Datei
topic	Betreff
content	Inhalt
timest	Zeitstempel der Erstellung
deleted	Zeitstempel der Löschung

<b>structure</b>	
Diese Tabelle bildet die Struktur (Menüpunkte) der Homepage ab.	
structureid	Identifikationsnummer der Strukturpunkte
parentstructureid	übergeordneten Struktur-ID
templateid	ID des Templates
personid	ID der Erstellerson
position	Position des Menüpunktes
name	Name
visible	Sichtbarkeit auf Seite
timest	Zeitstempel der Erstellung
checkout	Checkfeld für Bearbeitung
activate	Aktivierungsfeld
deleted	Zeitstempel der Löschung

<b>templatefile</b>	
Verbindungstabelle zwischen template und file	
templateid	ID des Templates
fileid	ID der Datei
name	Name für interne Mehrfachnutzung des selben files

<b>file</b>	
Hier werden alle inhaltlichen Dateien verwaltet (z.B. gif, jpg, pdf, doc, ...)	
fileid	Identifikationsnummer der Datei
filecategoryid	ID der Dateikategorie
mimetypeid	MIME-Typ der Datei
personid	ID der Erstellerson
filename	Name der Datei
description	Beschreibung
source	Quelle
timest	Zeitstempel der Erstellung
checkout	Checkfeld für Bearbeitung
activate	Aktivierungsfeld
deleted	Zeitstempel der Löschung

<b>messageperson</b>	
Verbindungstabelle zwischen message und person	
messageid	ID der Nachricht
personid	ID der Person

<b>template</b>	
Das Grundgerüst jeder einzelnen Seite wird hier gespeichert.	
templateid	Identifikationsnummer des Templates
personid	ID der Erstellerson
name	Name des Templates
code	Programmiercode
timest	Zeitstempel der Erstellung
checkout	Checkfeld für Bearbeitung
activate	Aktivierungsfeld
deleted	Zeitstempel der Löschung

<b>templatefilefact</b>	
Verbindungstabelle zwischen templatefile und filefact	
filefactid	ID des Dateifaktes
templatefileid	ID von Templatefile
data	Daten des Faktens

<b>filefact</b>	
Fakten zur Datei (z.B. Höhe, Breite, alternativer Text, Größe, ...)	
filefactid	Identifikationsnummer des Faktens
name	Name des Faktens

<b>filecategory</b>	
Dateikategorisierung für Übersichtlichkeit bei Erstellung (z.B. Icons, Gebäude, ...)	
filecategoryid	Identifikationsnummer der Dateikategorie
name	Name

<b>mimetype</b>	
MIME-Typ-Verwaltung	
mimetypeid	Identifikationsnummer des MIME-Typs
name	Name

<b>data</b>	
Hier werden alle textbasierten Inhalte gespeichert (z.B. Überschriften, ...)	
dataid	Identifikationsnummer der Daten
personid	ID der Erstellerson
name	Name
content	Inhalt
content_eng	Inhalt in Englisch
timest	Zeitstempel der Erstellung
checkout	Checkfeld für Bearbeitung
activate	Aktivierungsfeld
deleted	Zeitstempel der Löschung

<b>templatedatatype</b>	
Verbindungstabelle zwischen template und data; Typentabelle der Daten (z.B. Überschriften, Name, Matrikel, ...)	
templatedatatypeid	Identifikationsnummer des Templatedatatypes
templateid	ID des Templates
datatypeid	ID des Datentypes

<b>dataassign</b>	
Verbindungstabelle zwischen templatedatatype und data für die Zuordnung der Daten zu den Datentypen	
templatedatatypeid	ID des Templatedatatypes
dataid	ID der Daten
datagroup	Gruppe für Zugehörigkeit der Daten

<b>module</b>	
Diese Tabelle enthält den Programmcode für die einzelnen Module (z.B. Kalender, Buttons, ...)	
moduleid	Identifikationsnummer des Moduls
modulecategoryid	Modulkategorie
personid	ID der Erstellerson
name	Name des Moduls
description	Beschreibung
code	Programmiercode
timest	Zeitstempel der Erstellung
checkout	Checkfeld für Bearbeitung
activate	Aktivierungsfeld
deleted	Zeitstempel der Löschung

<b>templatemodule</b>	
Verbindungstabelle zwischen template und module	
templateid	ID des Templates
moduleid	ID des Moduls

<b>modulecategory</b>	
Modulkategorisierung für Übersichtlichkeit bei Erstellung (z.B. Elemente, Buttons, ...)	
modulecategoryid	Identifikationsnummer der Modulkategorie
name	Name

<b>style</b>	
Styleauflistung der Webseite	
styleid	Identifikationsnummer des Styles
fileid	ID der Datei (bei Backgroundimage)
name	Stylename
code	Quellcode

<b>color</b>	
Farbverwaltung	
colorid	Identifikationsnummer der Farbe
name	Name der Farbe
data	RGB-Code

<b>stylecolor</b>	
Verbindungstabelle zwischen style und color	
styleid	ID des Styles
colorid	ID der Farbe

## **Anhang C: Inhalt der CD**

Die beiliegende CD enthält das in dieser Arbeit beschriebene Beispiel eines Basis-WCMSs. Dazu gehören die serverseitige Indexdatei und ein Datenbankdump. Auf der CD befinden sich weiterhin das Datenbankschema und die Diplomarbeit selbst.