

# **Wiki und WCMS: Ein Vergleich**

Richard Cyganiak

19. Mai 2002

Hausarbeit im Rahmen des Seminars

**Content Management – Einführung in Theorie und Praxis**

von Tobias Müller-Prothmann im SS2001

Freie Universität Berlin, Institut für Publizistik- und Kommunikationswissenschaft

# Wiki und WCMS: Ein Vergleich

Richard Cyganiak

## Abstract

Wikis sind eine Form von Hypertext-Dokumenten. Diese Arbeit untersucht ihr Konzept aus der Perspektive des Content Management. Es werden Gemeinsamkeiten und Unterschiede aufgezeigt, und untersucht, was für Möglichkeiten die Anwendung von Wiki-Konzepten dem Content Management eröffnet.

## Inhalt

Einleitung.....	1
Was ist ein Wiki? .....	2
Wiki und Content Management .....	3
Jeder ist ein Autor .....	7
Der Content ist die Struktur.....	7
Lernende Texte .....	9
Fazit.....	10

## Einleitung

Informationen aus dem WWW zu holen, ist ein einfacher und von vielen Menschen ausgeführter Vorgang. Informationen ins WWW hineinzukriegen, ist hingegen deutlich schwieriger. Mit dem Anfänger-Dreigespann aus Texteditor, HTML-Buch und FTP-Programm ist es schwer, Ergebnisse zu erzielen, die den Aufwand wert sind. WYSIWYG-Editoren sind eine gewisse Hilfe, aber auch nur bis zu einer gewissen Grenze an zu verwaltenden Informationen. Jenseits dieser Grenze regiert das Content Management System. Ein praktisches Webinterface, leicht änderbare Templates und nützliche Tools können den Verwaltungs-Overhead massiv reduzieren. Dieser Spaß kostet jedoch. Wenn nicht Geld, dann wenigstens Zeit für das Überwinden der Lernkurve. Und ohne eigenen Server läuft meist gar nichts. Kleinere Lösungen existieren zwar, vor allem im Open-Source-Bereich, aber sie lassen besonders an Flexibilität einige Wünsche offen.

Am Rande dieses Bereichs – leichtgewichtige Open-Source-Lösungen für Web Publishing – findet man die sogenannten "Wikis". Wikis – man könnte sie als kollaborative, leicht zu bedienende Hypertext-Dokumente umschreiben – konkurrieren nun nicht gerade mit Web Content Management Systemen (WCMS), sondern gehören einem anderen, vor ihrer Entwicklung im Prinzip gar nicht vorhandenen, Anwendungsbereich an. Dennoch sind sie aus der Perspektive des Content Management interessant, da sie einige Dinge richtig machen, die sonst überall viel zu kompliziert sind. Diesen Dingen soll in dieser Arbeit nachgespürt werden.

Zuerst soll die Frage beantwortet werden, was genau Wikis sind, wodurch sie sich auszeichnen, und in welchen Bereichen sie Verwendung finden. Anschließend wird das Wiki-Konzept nach den Kriterien des Content Management untersucht, um Gemeinsamkeiten zu finden, aber auch um die wesentlichen Unterschiede herauszuarbeiten. In drei weiteren Abschnitten sollen diese Unterschiede genauer untersucht werden, um aufzuzeigen, welche Möglichkeiten sich durch eine breitere Verwendung von Wiki-Prinzipien eröffnen könnten.

## Was ist ein Wiki?

Wiki ist die Kurzform für *WikiWikiWeb*. Dies ist eine Form von Hypertext-Dokumenten, die 1995 von Ward Cunningham eingeführt wurde.<sup>1</sup> "Wiki wiki" ist der hawaiische Ausdruck für "schnell". Der Begriff "Wiki" wird verwendet, um ein konkretes solches Dokument ("Wikis"), das gemeinsame Konzept dieser Dokumente ("Wiki"), oder eins der zu ihrem Betrieb verwendeten Programme ("Wiki-Server") zu bezeichnen.<sup>2</sup>

Verschiedene Wikis unterscheiden sich erheblich in Verwendung und Funktionalität. Kanonisches Beispiel soll der erste, von Cunningham entwickelte Wiki-Server sein, sowie die mit ihm betriebene Website (das "originale" Wiki oder Portland Pattern Repository).<sup>3</sup> Auf die wichtigsten Varianten des Konzepts wird später eingegangen.

Der Besucher surft ein Wiki mit dem Webbrowser an, ebenso wie eine normale Website. Das Wiki besteht aus einer Menge einzelner Seiten, die meist stärker untereinander verlinkt sind als traditionelle Webseiten. Anders als bei diesen gibt es am Ende jeder Seite eine Schaltfläche oder einen Link, der beispielsweise "*EditText*" oder "Edit this Page" beschriftet ist. Ein Klick führt zu einem Formular, das den Text der Seite in einem großen Textbearbeitungsfeld anzeigt. Jeder Besucher kann hier Änderungen am Inhalt vornehmen. Nach dem Speichern ist die Seite sofort in der veränderten Form für alle Besucher sichtbar.

Die Formatierung des Textes erfolgt meist mit wenigen einfachen Regeln, beispielsweise werden aufeinanderfolgende Zeilen, die mit einem Sternchen oder einem Strich beginnen, als Listen formatiert. Links zu anderen Seiten innerhalb des Wikis erstellt man, indem man mehrere aufeinanderfolgende Worte mit großen Anfangsbuchstaben zu einem Wort zusammenfasst. Beispiele wären *HausarbeitÜberWiki*, *WasIstEinWiki*, *StartSeite*.

Diese sogenannten *WikiWords* werden automatisch im Browser als Links zu den Wiki-Seiten gleichen Titels dargestellt. Eine neue Seite erzeugt man, indem man ein *WikiWord* ohne entsprechende Seite anklickt. Der Link führt dann zu einer neuen, leeren Seite mit dem entsprechenden Titel.

Zusammenfassend enthält das Wiki-Konzept zwei zentrale Ideen:

- Jeder Besucher kann jede Seite verändern.
- Das Verändern und Erzeugen von Seiten wird so weit wie möglich erleichtert.

Im Folgenden soll kurz auf die wichtigsten Anwendung eingegangen werden.

### Wiki als Diskussionsforum

Ein typisches Wiki ähnelt einem öffentlichen Internet-Messageboard oder -forum, in dem jeder Besucher Beiträge posten kann. Allerdings verschwinden die Postings nicht nach wenigen Tagen in den Tiefen eines Archivs, sondern bilden eine permanente Website.

### Wiki als Web Content Management System

Nicht jedes Wiki gestattet jedem Besucher die Änderung des Textes. Am anderen Ende eines breiten Spektrum möglicher Abstufungen gibt es Websites, die aus der Sicht des Besuchers nicht von einer konventionellen Webpräsenz zu unterscheiden sind, aber "unter der der Haube" Wiki-Funktionalitäten aufweisen. Die Autoren solcher Sites profitieren immer noch von der einfachen Bedienung, die ein Wiki bietet.

### Wiki als Groupware

Eine weitere naheliegende Verwendung für ein Wiki ist die als Kommunikationstool in einem geschlossenen Projekt. Ein Wiki kann dabei als Ergänzung zu den üblichen Tools – Mailinglisten und

---

<sup>1</sup> leitner

<sup>2</sup> wikipedia

<sup>3</sup> leuf/cunningham, wikiwikiweb

Groupware-Lösungen – dienen.<sup>4</sup> Dies spielt auch in den Bereich des Knowledge Management hinein. Dass 60 % aller Intranetprojekte scheitern<sup>5</sup>, sollte ein Grund sein, leichtgewichtige Lösungen in Erwägung zu ziehen.

### Wiki als Personal Information Manager

Eine sicherlich exotische, aber interessante Verwendung ist die als PIM. Hier ist Wiki eine gute Möglichkeit, Textinformationen strukturierter zu speichern als in einfachen Textdateien oder der Email-Inbox – und mit Hyperlinks.

Anders als typische Lösungen aus den genannten Bereichen (CMS, KMS, Groupware) ist Wiki ein extrem leichtgewichtiges Tool. Ein Wiki-Server ist schnell installiert, und die meisten sind Open Source. Bei einer Reihe von kommerziellen<sup>6</sup> oder kostenlosen<sup>7</sup> Anbietern kann man ein Wiki anmieten. Besonders, wenn Art und Umfang der Verwendung zu Beginn nicht völlig klar sind, kann dies ein Argument für die äußerst flexiblen Wikis sein.

### Wiki und Content Management

Einen Wiki-Server mit einem Content Management System zu vergleichen scheint eine abwegige Idee zu sein. Der originale Wiki-Server von Cunningham besteht aus gerade etwa 400 Zeilen Quellcode<sup>8</sup> – überraschend genug, dass ein so kurzes Programm überhaupt etwas Nützliches tut. Zum Vergleich: phpCMS, ein Open-Source-CMS mit kurzer Featureliste besteht aus 14.000 Zeilen Quellcode<sup>9</sup>. Und über kommerzielle Systeme soll erst gar nicht spekuliert werden. Dennoch ist der Vergleich sinnvoll, denn er zeigt die Stärken, Schwächen und konzeptionellen Unterschiede des Wiki-Konzepts gegenüber dem traditionellen Web Content Management.

Büchner et al stellen eine Reihe von Vorteilen des WCMS-Einsatzes gegenüber dem herkömmlichen Webpublishing heraus<sup>10</sup>. Diese decken sich zum großen Teil mit den Vorteilen, die ein Wiki bringt:

Webpublishing mit WCMS	Von Wiki erfüllt?
Inhalt und Layout in Assets und Vorlagen getrennt	ja
Direktes Publizieren auch technisch nicht versierter Mitarbeiter	ja
Dezentrales Arbeiten an einem verteilten System	ja
Kürzere Time-to-Web durch Dezentralisierung	ja
Proportionales Wachstum von Aufwand und Contentmenge	fraglich
Automatisierter Workflow über gesamten Content Life Cycle	nein
Automatisierte Pflege durch WCMS, z.B. Link-Checking	teilweise
Mitarbeiter entsprechend ihrer Kompetenz eingebunden	ja
Einfaches Redesign über Änderungen der Vorlagen	ja

Im Folgenden soll ein genauerer Blick auf einzelne Funktionen eines WCMS geworfen werden. Werden diese von einem Wiki erfüllt? In welchen Punkten zeigt sich ein konzeptioneller Unterschied zwischen CMS und Wiki, und welche sind nur eine Frage der konkret betrachteten Wiki-Server<sup>11</sup> bzw. des allgemeinen Entwicklungsstandes dieser Programme? Welche Funktionen eines Wikis werden von einem CMS nicht abgedeckt? Die Zusammenstellung der Funktionen eines CMS orientiert sich an Rothfuß und Ried<sup>12</sup>.

<sup>4</sup> leuf/cunningham, S. 370

<sup>5</sup> nach einer in jung2001 zitierten Studie, S. 15

<sup>6</sup> wikiservice

<sup>7</sup> seedwiki

<sup>8</sup> daynotes

<sup>9</sup> phpcms

<sup>10</sup> büchner, S. 94

<sup>11</sup> wikiwikiweb, phpwiki, twiki, usemod

<sup>12</sup> rothfuß/ried, S. 73ff

## **Zugangskontrolle**

Das originale Wiki ist absolut offen angelegt, was es für viele potentielle Anwendungen ausschließt. Verschiedene Stufen von Benutzer- und Contentkontrolle sind daher meist die ersten Punkte, die in der Featureliste erweiterter Wiki-Server auftauchen. Für statische Websites mit Wiki-Backend oder Extranet-Wikis sind diese Features unabdingbar. TWiki<sup>13</sup> ermöglicht die Registrierung individueller Benutzer, sowie das Anlegen von Benutzergruppen und die feingranulare Vergabe von Schreib- und Leserechten (pro Seite), und lässt damit in diesem Punkt kaum Wünsche offen.

## **Protokollfunktionen**

Die "RecentChanges"-Seite ist ein Grundbestandteil des Wiki-Konzepts und Anlaufpunkt für regelmäßige Besucher. Im originalen Wiki ist sie sehr einfach über den Timestamp der geänderten Seiten implementiert. Mächtigere Funktionen zur Protokollierung stehen ebenfalls weit oben in der Featureliste besserer Wiki-Server. Sie speichern in der Regel eine detaillierte Historie jeder einzelnen Seite. Eine meist "diff" genannte Funktion ermöglicht die genaue Verfolgung aller gemachten Änderungen.

## **Datensicherung / Rollback**

In einer offenen Umgebung wie einem Wiki muss der Schutz vor böswilligen Eingriffen bedacht werden. Das originale Wiki verlässt sich hierbei lediglich auf eine externe Backup-Lösung und hat überraschend gute Erfahrungen damit gemacht<sup>14</sup>. Wie schon erwähnt, speichern viele andere Server eine Historie aller Seiten und ermöglichen dadurch auch das Rollback jeder einzelnen Seite zu einem beliebigen vergangenen Zeitpunkt.

## **Mehrplatzfähigkeit**

Diese ist bei einem Wiki per Definition gegeben. Verschiedene Seiten können natürlich unabhängig voneinander bearbeitet werden. Konflikte bei der gleichzeitigen Bearbeitung der selben Seite durch mehrere User werden im originalen Wiki nicht abgefangen. Die anderen untersuchten Server warnen den User beim Speichern der Änderungen, so dass er selbst seine Änderungen mit denen des anderen Users abgleichen kann. TWiki warnt schon am Anfang des Editiervorgangs, dass die Seite zur Zeit von einem anderen User bearbeitet wird.

## **Check-In und Check-Out**

Ein solches Konzept ist in den betrachteten Wiki-Servern nicht bekannt. Die zuletzt beschriebene Funktion von TWiki kann als eine Art von Checkout-Support betrachtet werden. Heutige Wikis haben wenig Bedarf nach dieser Funktion, Wiki-Content wird nicht für mehrere Tage ausgecheckt. Gründe dafür sind die informelle Natur der meisten Wikis und die Art des Contents (einfacher, "flacher" Text). Bedarf für diese Funktion ist aber durchaus vorstellbar, z.B. wenn ein Buch kollaborativ mittels einem Wiki geschrieben wird.

## **Metainformationen und Verwaltungsfunktionalität**

Hier tritt ein deutlicher Unterschied zwischen Wikis und CMS zutage. Bei letzteren ist die Fähigkeit zur Anreicherung der verwalteten Objekte mit Metadaten ein wesentliches Merkmal. Wikis hingegen kümmern sich kaum um Metainformationen. Der originale Wiki-Server begnügt sich mit den vom Dateisystem gebotenen Attributen und speichert keinerlei zusätzliche Daten. Warum Wikis trotzdem funktionieren, ist eine interessante Frage, der später ein eigener Abschnitt gewidmet werden soll.

Zum Thema Verwaltungsfunktionalität sei nur eines am Rande angemerkt: Eine charmante Eigenschaft der Wikis ist, dass es per Definition keine toten Links gibt (zumindest Site-intern).

---

<sup>13</sup> twiki

<sup>14</sup> leuf/cunningham, S. 326

## Anfragefunktionen

Der Mangel an Metainformationen in Wikis beschränkt auch die Möglichkeiten zur Formulierung von Anfragen. Typischerweise unterstützte Anfragen sind die Suche nach Seitentitel, die Volltextsuche, *RecentChanges* (Liste der letzten Änderungen) und Backlinks (Liste der Seiten, die auf Seite X linken), sowie die Suche nach verwaisten Seiten, die keinen "eintreffenden" Link mehr haben. Die phantasievolle Verwendung von *WikiWords* kann weitere Anfragemöglichkeiten eröffnen. Auch hierzu später mehr.

## Massenoperationen

Massenoperationen sind in traditionellen CMS häufig nur über Makrosprachen zugänglich<sup>15</sup>. Ähnlich sieht es in den Wikis aus, wo man oft mit Perl-Skripten oder ähnlichen Werkzeugen hantieren muss. Das Open-Source-Modell der meisten Wikis und ihre einfache Datenstruktur erleichtern solche Vorgänge.

## Bearbeitung und Verifikation

Bearbeitung und Verifikation sind bei Wikis naturgemäß relativ unkompliziert. Der Content von Wiki ist plain text, und als Editor wird das Formulareingabefeld des Webbrowsers verwendet. WYSIWYG-Editoren für Wiki (als Java-Applet oder JavaScript) werden immer wieder diskutiert und auch umgesetzt<sup>16</sup>, haben aber nicht Fuß fassen können. Die Gründe dafür sind unklar. Möglicherweise bringen WYSIWYG-Editoren in diesem Bereich nicht so viel, wie man instinktiv annehmen könnte (vgl. Email-Clients, deren seit Jahren verfügbare WYSIWYG-Editoren wenig genutzt werden).

## Aggregation und Beziehungen

Links sind ein Mehrwert, sonst gäbe es das WWW nicht. Nirgendwo ist die Schaffung dieses Mehrwerts leichter als in einem Wiki. Wiki-Content besteht aus Seiten. Jede Seite hat einen Namen. Jede Verwendung des Namens stellt automatisch den Link zu der Seite her. Erfahrene Wiki Autoren formen aus Gewohnheit Interessante Begriffe zu *WikiWörtern* um, ohne zu wissen, ob sie damit eine existierende Seite erreichen werden. So entsteht ein weitaus dichteres Netz von Beziehungen zwischen den einzelnen Seiten, als es möglich wäre, wenn das Erstellen von Links mit zusätzlichem Aufwand (und seien es nur ein paar Mausklicks) verbunden ist. Wenn der Mehraufwand für das Setzen eines Links gegen Null geht, entstehen interessante neue Verwendungen für Links, auf die später eingegangen wird.

## Mehrsprachfähigkeit

Für die parallele Verwaltung des selben Inhalts in verschiedenen Sprachen sind Wikis sicherlich nicht geeignet. Wikis sind dynamische Gebilde. Die synchrone Verwaltung verschiedensprachiger Versionen des gleichen Inhalts ist ein aufwändiger Vorgang und passt dabei nicht ins Konzept. Die Grundvoraussetzung für den Einsatz in vielen Sprachen bringen Wikis jedoch mit, da der Unicode-Standard von allen gängigen Browsern unterstützt wird, und ein Wiki sich für Ein- und Ausgabe auf den Browser stützt.

## Workflow

Hier ist wieder ein deutlicher konzeptioneller Unterschied zwischen Wikis und CMS zu sehen. Wikis kennen das Konzept eines Content Lifecycle nicht. Die schnelle und unmittelbare Verfügbarkeit aller Änderungen ist ein wichtiges Merkmal der Wiki-Idee und lässt sich mit einem komplexen Workflow nicht vereinbaren. Für Umgebungen, in denen die Schaffung neuen Inhalts ein mehrstufiger Prozess ist, kommt ein Wiki daher nicht in Frage.

---

<sup>15</sup> rothfuß/ried, Abschnitt 4.4.2.9

<sup>16</sup> seedwiki (WYSIWYG-Editor nur im Microsoft Internet Explorer lauffähig)

## **Gestaltung**

Die Trennung von Content und Layout ist in einem Wiki notwendiger Weise realisiert, da die Inhalte in einer nicht online publizierbaren Form geschaffen werden. Die Generierung von HTML bzw. XHTML für die Anzeige im Browser erfolgt mittels einfacher Templates. Da für die Formatierung nur ein Satz einfacher Regeln zur Verfügung steht, sind Autoren in ihren Möglichkeiten zur Gestaltung des Inhalts eingeschränkt. Dies ist jedoch in traditionellen CMS ähnlich, es sei denn, die Autoren sollen den Komplexitäten von HTML ausgesetzt werden.

## **Fremdformatwandlung**

Die wichtigste Im- und Exportschnittstelle eines Wikis ist wahrscheinlich die Copy & Paste-Funktion des Webbrowsers. Das Hauptproblem bei der Wandlung verschiedener Formate, nämlich die oftmals drastisch unterschiedliche Expressivität, reduziert sich bei der Konvertierung in oder aus einfachem Text erheblich. Dementsprechend lassen sich für fast alle Im- und Exportanforderungen eines Wikis relativ einfach Ad-Hoc-Lösungen finden. Heutige Wikis verzichten daher auf explizite Im- und Export-Schnittstellen zu bestimmten Formaten. Oftmals mitgeliefert sind Tools zum Import von Daten aus anderen Wiki-Servern.

## **Link-Stabilität**

Ein konzeptionelles Problem der Wikis ist, dass der Content einer bestimmten Wiki-Seite per Definition dynamisch ist. Daher kann man nicht davon ausgehen, dass die gleiche Information zu einem späteren Zeitpunkt noch unter der gleichen URL zu finden ist, was prinzipiell vermieden werden sollte.<sup>17</sup> Damit wird es riskanter, auf Wiki-Seiten von außerhalb zu linken oder Zitate aus einem Wiki mit einer URL als Referenz zu belegen.

## **Multimedia**

Durch das originale Wiki kann man stundenlang surfen, ohne eine einzige Grafik zu sehen. Die Einbindung multimedialer Assets ist in Wiki zwar geringfügig leichter als für den HTML-Autor mit Texteditor und FTP-Programm (Hilfen sind z.B. Fileupload-Features und spezielle Textformatierungsregeln für die Einbindung von Grafiken), aber gegenüber dem herrlich einfachen Umgang mit Text müssen Grafiken et al als wahrer Krampf erscheinen. Eine Randnotiz ist das TWikiDraw-Plugin wert, welches über ein Java-Applet die Erstellung einfacher Grafiken ermöglicht. Das Beispiel deutet darauf hin, dass die Textlastigkeit heutiger Wikis nicht prinzipbedingt ist. Die Unterstützung von reinem Text ist jedoch viel einfacher zu realisieren als andere Medienformate. Beim heutigen Entwicklungsstand der Wiki-Server sind sie jedoch damit für viele Anwendungen disqualifiziert, da Seiten aus reinem Text im heutigen Web eigentlich nicht zeitgemäß sind.

## **Kollaboration**

Dies ist kein Feature eines CMS, aber ein Feature von Wiki. Die Idee des kollaborativen Schreibens ist vermutlich nicht neu, aber konnte wohl noch nie so effektiv realisiert werden wie in einem Wiki. Potentiell kann dort jeder vorbeisurfende Besucher seine Beiträge hinterlassen und den Wert der Seite damit – hoffentlich – erhöhen.

Damit sind folgende konzeptionelle Unterschiede zwischen Wikis und CMS ausgemacht:

- Wikis verwischen die Grenze zwischen Besuchern und Autoren, indem sie beiden weitgehend gleiche Rechte einräumen.
- Wikis verzichten weitgehend auf eine feste Inhaltsstruktur und Metainformationen, und emulieren diese durch Links.
- Wikis folgen nicht dem Modell des Content Life Cycle, sondern betrachten die Gesamtheit des Contents als evolutionäres Gebilde.

Auf diese Eigenheiten soll nun jeweils in einem eigenen Abschnitt näher eingegangen werden.

---

<sup>17</sup> nielsen, berners-lee98

## Jeder ist ein Autor

Das WWW entstand aus der Arbeit von Tim Berners-Lee am CERN. Ziel war es, die großen Informationsmengen, die an einer solchen Forschungseinrichtung anfallen, besser nutzbar zu machen. Berners-Lee hoffte zum Beispiel, dass man damit den ständigen Informationsverlust durch die hohe Mitarbeiterfluktuation verringern könnte.<sup>18</sup> Man kann also spekulieren, dass seine ursprüngliche Vision eine stärkere aktive Beteiligung aller Benutzer vorsah, als wir sie heute erleben. So war der von Berners-Lee entwickelte erste Webbrowser gleichzeitig ein Web-Editor. Neue Links konnten über einen eigenen Menüpunkt erstellt werden. Eine andere Funktion ermöglichte es, etwas Text zu markieren, und daraus einen Link zu einer neuen, leeren Seite zu machen.<sup>19</sup>

Durchgesetzt hat sich dieser Ansatz nicht. Der heutige Webbrowser ist weitgehend eine Einbahnstraße vom Content Provider zum Konsumenten. In dieser Hinsicht gleicht das Web den älteren Medien Fernsehen, Hörfunk und Zeitungen. Bidirektionale Kommunikation findet eher in anderen, älteren Teilen des Internets statt (Email, Newsgroups).

Warum? Hier kann nur eine spekulative Erklärung angeboten werden. Den frühen Usern des Webs (mit überwiegend technischem Hintergrund) genügte es, einen komfortablen Browser zu haben, um das Web zu einem wertvollen Medium zu machen. Das Fehlen eines benutzerfreundlichen Tools zum Erzeugen von Content störte sie wenig. Als sich das Web größeren Kreisen erschloss, wurde diese Schwierigkeit durch stetig steigende Ansprüche an die Präsentation des Contents weiter vergrößert. Das Web war schon als unidirektionales Medium so nützlich, dass an einem Ausbau der bidirektionalen Möglichkeiten wenig Interesse bestand.

Was ursprünglich Bestandteil des Webbrowsers sein sollte, aber nicht geworden ist, muss jetzt über Umwege realisiert werden. Content Management Systeme gehen den langen Umweg, indem sie alles zur Verfügung stellen, was man zur automatisierten Erstellung einer jener Websites benötigt, die ursprünglich von Hand gemacht wurden, und darauf aufbauend neue Möglichkeiten eröffnen. Wikis gehen den kurzen Umweg, indem sie das zur Verfügung stellen, was unbedingt nötig und dabei technisch am einfachsten machbar ist – Webseite, Link, Suchfunktion. Schattenseite dieser Beschränkung ist, dass Wikis sich in ihrem Charakter von der gewohnten Standard-Website unterscheiden. Vorteil ist jedoch, dass die Bedienung eines Wikis um Größenordnung einfacher ist.

In dieser Hinsicht werden Wiki-ähnliche Umgebungen schwer zu übertreffen sein (es sei denn, die Editorfunktionen kehren in die Webbrowser zurück). Selbst simple und äußerst populäre interaktive Web-Anwendungen wie Message Boards (oder selbst einige Gästebücher) sind schwieriger zu bedienen. Wikis wahre Stärke liegt damit in der Verwendung als Kommunikationstool. Es verwischt die dem Web inhärente Grenze zwischen Konsument und Autor, indem es potentiell jeden Konsumenten zum Autor macht.

## Der Content ist die Struktur

Das klassische Strukturmodell einer Website ist der hierarchisch verzweigte Baum mit der Homepage als Wurzel. Jedes Content-Objekt nimmt einen bestimmten Platz in dieser Hierarchie ein. Zusätzliche Querverweise erleichtern die Navigation zwischen verwandten Elementen in unterschiedlichen Ästen. Dieses Modell ist vielfach bewährt, aber nicht besonders flexibel. Wikis verzichten meist völlig auf eine vom System vorgegebene hierarchische Ordnung. Das klingt nach einem Rezept für Chaos, und genau das ist es.

Der Mangel an Struktur scheint das größte Problem beim praktischen Einsatz eines Wikis zu sein.<sup>20</sup> Je größer die Zahl der Seiten, desto schwerer wird es, den Überblick zu behalten und bestimmte Informationen zu finden. Inhalt wird doppelt eingegeben, die Links zwischen den Seiten sind planlos und zufällig, Links zeigen auf Seiten, die inzwischen ganz andere Informationen enthalten.

---

<sup>18</sup> berners-lee89

<sup>19</sup> berners-lee

<sup>20</sup> leuf/cunningham, S. 370f



Natürlich erlauben es die Wiki-Bordmittel, eine hierarchische Struktur durch menüartige Überblicksseiten nachzubilden, und dies ist auch die empfohlene Vorgehensweise.<sup>21</sup> Doch das Anlegen neuer und das Verändern bestehender Seiten führen zu beständiger Erosion der einmal angelegten Sitestruktur. Diesem Prozess muss durch sogenannte "Refactorings" entgegengewirkt werden. Darunter versteht man die Neustrukturierung bestehenden Inhalts, im Gegensatz zur Schaffung neuen Inhalts.<sup>22</sup> Dies umfasst Tätigkeiten wie das Erstellen und Aktualisieren von menüartigen Überblicksseiten, das Umbenennen von Seiten, die Schaffung neuer Links, die Zusammenfassung doppelter Informationen, die Aufspaltung zu großer und Löschung überflüssiger Seiten. Dies ist ein nötiger Mehraufwand, um die vorhandene Informationsmenge in nützlicher und strukturierter Form zu halten.

Während in traditionellen Web-Publishing-Verfahren bereits in der Design-Phase eine Struktur festgelegt wird, in die sich der Content einpassen muss, entsteht in einem Wiki diese Struktur erst durch den Content und kann sich demnach leichter an veränderte Bedürfnisse anpassen. Das hat Vorteile, wenn Art des Contents und Verwendung der Website vorher nicht genau absehbar sind. Oft entstehen aus dem Einsatz eines Wikis neue Verwendungsmöglichkeiten, die von den Initiatoren nicht vorgesehen waren.<sup>23</sup>

Es ist interessant, wie die Einfachheit der Erzeugung von Hyperlinks deren Gebrauch in Wiki verändert. Eine Reihe von Aufgaben, die in anderen Umgebungen die aufwändige Einführung von Metainformationen erfordert hätten, werden in Wiki durch Hyperlinks erledigt. Einfaches Beispiel ist die gängige Praxis des Signierens von Beiträgen mit dem eigenen Namen als *WikiWord*, beispielsweise *WardCunningham*. Ein Klick auf den Name führt zu einer Seite, auf welcher der Autor weitere Informationen über sich hinterlegt. Eine andere interessante Verwendung sind die sogenannten "WikiBadeges" oder "WikiTags",<sup>24</sup> wie *DeleteMe*, *AnswerMe*, *PleaseComment*, *BrokenLink*. Sie zeigen beispielsweise den Bedarf von Refactoring an, und dienen als administratives Werkzeug, da man über die Backlink-Funktion leicht an eine Liste aller Stellen gelangt, die mit einem *WikiTag* gekennzeichnet sind.

Damit stellen die Links in einem Wiki nicht nur einen Ersatz für Metainformationen, sondern gleichzeitig ein generisches Werkzeug zur Meisterung verschiedener administrativer Probleme dar. Ein einziger einfacher Mechanismus erfüllt Funktionen, die in einer anderen Umgebung jeweils einzeln implementiert worden wären, mit eigenen Regeln und eigener Lernkurve für den Anwender. Dazu braucht es einige Konventionen über die Verwendung dieses Hilfsmittels. Diese Vereinbarungen können sich durchaus im Laufe der Zeit entwickeln und verändern. Damit wird Verantwortung aus den Händen des Programms bzw. dessen Entwicklers zurück in die Hände der Anwender und Autoren gelegt.

Ein Blick auf die große Welt des WWW zeigt zwei Beispiele, die darauf hindeuten, dass solche Strukturinformationen im Content nützlicher und aussagekräftiger als Metainformationen sein können. Das erste Beispiel ist die WWW-Suchmaschine Google. Während sich ältere Suchmaschinen auf explizite Metainformationen der untersuchten Seiten stützten, führte Google sehr erfolgreich<sup>25</sup> ein Verfahren ein, dass die Links zwischen den Seiten analysiert<sup>26</sup> und damit deutlich bessere Ergebnisse erzielt.

Das zweite Beispiel ist die Entwicklung der Hypertext Markup Language HTML. Seit den frühesten Entwürfen von HTML<sup>27</sup> war neben dem allseits bekannten Hyperlink (dem A-Element von HTML) ein weiterer Mechanismus zur Herstellung von Beziehungen zwischen Dokumenten vorgesehen: das LINK-Element. Dieses ist optionaler Bestandteil des HEAD-Abschnitts (Abschnitt für Metainformationen) eines HTML-Dokuments und ermöglicht eine genaue Klassifizierung der Art der Beziehung. Autoren können damit ausdrücken, dass ein Dokument beispielsweise Nachfolger in einer Reihe von Dokumenten, übergeordnet in einer Hierarchie, Inhaltsverzeichnis, Index oder Übersetzung eines anderen Dokuments ist.<sup>28</sup> Webbrowser können diese Beziehungen beispielsweise als eine Toolbar

---

<sup>21</sup> leuf/cunningham, S. 390

<sup>22</sup> wikiwikiweb-refactoring

<sup>23</sup> leuf/cunningham, S. 357ff

<sup>24</sup> wikiwikiweb-badges

<sup>25</sup> google2001, searchenginewatch

<sup>26</sup> google2002

<sup>27</sup> berners-lee/connolly

<sup>28</sup> w3c

mit Buttons, die zu den entsprechenden Seiten führen, darstellen.<sup>29</sup> Trotz der umfangreichen Möglichkeiten des LINK-Elements hat es sich bisher nicht durchsetzen können und ist von Webbrowser-Entwicklern weitgehend ignoriert worden.

Eine breitere Unterstützung des LINK-Elements würde die Navigation im WWW zweifellos vereinfachen.<sup>30</sup> Dennoch zeigt das Beispiel, dass im Kontext des WWW der im Content integrierte "klassische" Hyperlink dem typisierten, strukturfördernden Metainformations-Link klar überlegen ist. Ohne den ersten wäre das WWW nicht vorstellbar. Ohne den zweiten kommen wir sehr gut zurecht, auch wenn er bei der Navigation durch stark strukturierten Content eine Hilfe sein könnte.

Um den Abschnitt zusammenzufassen: Betrachtet wurden zwei verschiedene Möglichkeiten, dem Content einen Mehrwert zu verleihen. Die eine Möglichkeit ist es, den Content mit Metainformationen zu beschreiben. Die andere ist es, den Content durch verschiedene Links eng mit relevanten anderen Objekten zu verknüpfen, also Kontext herzustellen. Die zweite Methode wird im Bereich des Dokumenten- und Content-Management eher weniger eingesetzt. Beide Möglichkeiten überschneiden sich teilweise, ergänzen sich, und haben ihre Vorteile:

- Metainformationen: Aussagekraft, Genauigkeit
- Kontext durch Links: konzeptionelle Einfachheit, Flexibilität

Traditionelle CMS verlassen sich mehr auf Metainformationen, Wikis verlassen sich mehr auf die Herstellung von Kontext durch Links. Wikis können dadurch mit wesentlich einfacheren Mitteln gute Resultate erzielen und sind deutlich flexibler, leiden aber an einem Mangel an klarer Struktur.

## Lernende Texte

Im Gegensatz zu den meisten Texten, die zu lesen wir gewohnt sind, hat ein Wiki keine fertige, endgültige Form. Es ist nur einen Schnappschuss aller Beiträge zu einem gegebenen Zeitpunkt und ist ständig in Entwicklung. In der Welt von Büchern und Zeitungen kann ein einmal publiziertes Wort nicht mehr geändert werden. Bis vor Kurzem gab es daher keine andere Möglichkeit als den strikten Content Lifecycle aus Erstellung, Prüfung, Veröffentlichung, Archivierung. Die vorherrschende Textform im WWW ist immer noch der Artikel – also der kurze, in sich geschlossene Sachtext.<sup>31</sup> Im Prinzip wurden im Wesentlichen Arbeitsabläufe und Konzepte aus der Zeitungswelt auf ein neues Medium verlagert. Den WCMS kommt dabei die Aufgabe zu, den übernommenen Arbeitsablauf weitestgehend zu automatisieren. Dabei haben die neuen technischen Möglichkeiten durchaus das Potential, den Arbeitsablauf des Publizierens von Grund auf zu verändern. Wiki zeigt eine solche Möglichkeit.

Betrachtet man die Bereiche des Internets, in denen tagtäglich der meiste Content produziert wird – News-Portale, Usenet-Newsgruppen, Diskussionsforen, Mailinglisten – so ist ihnen gemein, dass sie aus einem mehr oder weniger konstanten Contentstrom bestehen. Es interessiert nur der aktuelle Content – also jene Informationsmenge, die innerhalb der letzten n Stunden oder Tage durch das Medium geflossen ist. Älterer Content liegt zwar archiviert vor, ist aber um Größenordnungen weniger relevant. Durch den Archiveingang der genannten Medien fließt ein Schatz an Informationen, die leider kaum verwertbar sind, da keinerlei Filterung und Verdichtung erfolgt, und ein vernünftiges Retrieval der Informationen später kaum möglich ist.

Die nötige Filterung und Verdichtung erfolgt durchaus, aber außerhalb des unmittelbaren Mediums. Besucher kopieren oder merken sich interessantes Material, erinnern sich später daran, und können es an anderer Stelle einfließen lassen. Eine Menge kreativer Tätigkeit im Bereich des geschriebenen Wortes findet in dieser Weise statt.

Wikis oder ähnliche Technologien könnten diesen Prozess unmittelbarer gestalten. Autoren lernen dazu - warum sollen nicht auch Texte dazulernen? Ein Wiki-Autor kann aus einer langen Diskussion mit vielen Beiträgen verschiedener Autoren die Essenz herausdestillieren und in kurzer, übersichtlicher

---

<sup>29</sup> gutfeldt

<sup>30</sup> tekelenburg

<sup>31</sup> websters

Form konzentrieren. Wiederholte Anwendung dieses Wiki-Refactorings kann wertvolle Informationen schaffen, die sonst nicht nutzbar geworden wären.

Besonders naheliegende Anwendungen sind etwa Tech-Support-Datenbanken oder die kollaborative Zusammenstellung von Usenet-FAQs. Gerade diese FAQs<sup>32</sup> sind vielleicht das beste Beispiel dieses "filtern und verdichten"-Prozesses außerhalb einer Wiki-Umgebung, auch wenn sie meist von einer einzelnen Person und mit konventionellen Mitteln erstellt werden.

## Fazit

Ein Wiki-Server ist natürlich kein WCMS. Beide haben weitgehend unterschiedliche Anwendungsbereiche mit einer kleinen Schnittmenge. WCMS benutzt man, um Websites zu betreiben. Wiki-Server benutzt man, um, nun ja, Wikis zu betreiben. Dennoch drängt sich die Frage auf, warum Wiki in einem dem CMS verwandten Bereich mit dramatisch weniger Komplexität auskommt.

Content Management Systeme beschäftigen sich mehr mit Management als mit Content. Man kann endlos über ihre Features und technischen Details reden. Wenn man über Wikis redet, stellt man fest, dass sie zu einfach sind, als dass man sich lange mit technischen Details aufzuhalten könnte. Man kommt schnell darauf zu sprechen, wie die Leute ein Wiki verwenden, was für interessante Anwendungen und kreative Möglichkeiten es eröffnet. Warum kann Content Management nicht so einfach sein?

Die Antwort ist klar. Ein Web Content Management System versucht, die Erfahrungen, Arbeitsabläufe und Methodologien der vergangener Jahrzehnte auf die heutige Technik abzubilden. Ein Wiki hingegen verschreibt sich ganz den Anforderungen der Technik. Es stellt den Versuch dar, die kleinste Menge von Funktionen zu finden, die nötig sind, um auf der Basis des WWW ein brauchbares kollaboratives Tool zu bilden. Auf die Gewohnheiten des Anwenders nimmt es wenig Rücksicht und diktiert ihm damit teilweise völlig neuartige Arbeitsabläufe und Denkweisen.

Sicher ist beides nicht ganz richtig, weder das von technischen Anforderungen dominierte Wiki-Konzept, noch der traditionelle Publishing-Ansatz der CMS. Aber je mehr sich unsere Erwartungen und Gewohnheiten der Technik des Webs anpassen, desto mehr werden auch einige Wiki-Besonderheiten zur Normalität werden. Wiki zeigt also, wohin die technischen Anforderungen des Webs das Content Management System drängen.

---

<sup>32</sup> hersch

## Quellen

- büchner *Heino Büchner et al.: Web Content Management. Websites professionell betreiben, Bonn 2001*
- berners-lee *Tim Berners-Lee: The WorldWideWeb Browser,*  
<http://www.w3.org/People/Berners-Lee/WorldWideWeb.html> (30.4.2002)
- berners-lee89 *Tim Berners-Lee: Infomation Management. A Proposal, 1989,*  
<http://www.w3.org/History/1989/proposal.html> (11.5.2002)
- berners-lee98 *Tim Berners-Lee: Cool URIs don't change, 1998,*  
<http://www.w3.org/Provider/Style/URI> (11.5.2002)
- berners-lee/  
connolly *Tim Berners-Lee, Daniel Connolly: Hypertext Markup Language (HTML), 1993,*  
<http://www.w3.org/MarkUp/draft-ietf-iiir-html-01.txt> (19.5.2002)
- daynotes *Ward Cunningham, Bo Leuf: Wiki Way Sources. In: Daynotes Forum,*  
<http://leuf.net/cgi/wikidn?WikiWaySources> (5.5.2002)
- google2001 *Google, Inc.: The Google Timeline, 2001,*  
<http://www.google.com/corporate/timeline.html> (11.5.2002)
- google2002 *Google, Inc.: Google Technology, 2002,*  
<http://www.google.com/technology/> (11.5.2002)
- gutfeldt *Matthias Gutfeldt: Browser mit LINK Unterstützung, 2000-2001,*  
<http://matthias.gutfeldt.ch/translation/LINK/addendum.html>  
(19.5.2002)
- hersch *Russ Hersch: FAQs about FAQs, 1998,*  
<http://www.faqs.org/faqs/faqs/about-faqs/> (19.5.2002)
- jung2001 *Alexander Jung et al.: Studie zum Knowledge Management und zur Informationsbeschaffung von Ingenieuren, Kölln 2001,*  
<http://www.jung24.de> (12.5.2002)
- leitner *Helmut Leitner: Die Geschichte des Wiki Web. In: WikiWebAt, 2001,*  
<http://www.wikiservice.at/wikiweb/wiki.cgi?DieGeschichteDesWikiWeb>  
(11.5.2002)
- leuf/cunningham *Bo Leuf, Ward Cunningham: The Wiki Way. Collaboration and Sharing on the Internet, Reading 2001*
- nielsen *Jakob Nielsen: Fighting Linkrot, 1998,*  
<http://www.useit.com/alertbox/980614.html> (19.5.2002)
- phpcms *phpCMS,*  
<http://www.phpcms.de/homepage/phpcms/> (19.5.2002)
- phpwiki *PhpWiki,*  
<http://phpwiki.sourceforge.net/> (30.4.2002)
- rothfuß/ried *Gunther Rothfuß, Christian Ried (Hg): Content Management mit XML. Grundlagen und Anwendungen, Berlin, Heidelberg, New York 2001*
- searchenginewatch *searchenginewatch.com: Searches Per Day, 2000-2002,*  
<http://www.searchenginewatch.com/reports/perday.html> (11.5.2002)
- seedwiki *SeedWiki,*  
<http://seedwiki.com> (30.4.2002)
- tekelenburg *Sander Tekelenburg: Navigating the WWW, 2000-2001,*  
<http://www.euronet.nl/~tekelenb/WWW/LINK/> (19.5.2002)

---

twiki	<i>Peter Thoeny</i> : TWiki, <a href="http://www.twiki.org">http://www.twiki.org</a> (12.5.2002)
usemod	<i>Clifford Adams</i> : Use Mod Wiki, <a href="http://www.usemod.com/cgi-bin/wiki.pl?UseModWiki">http://www.usemod.com/cgi-bin/wiki.pl?UseModWiki</a> (11.5.2002)
w3c	<i>W3 Consortium</i> : Link Types. In: HTML 4.01 Specification. 1997-1999, <a href="http://www.w3.org/TR/html401/types.html#type-links">http://www.w3.org/TR/html401/types.html#type-links</a> (11.5.2002)
websters	<i>Richard Weiner (Hg)</i> : Article. In: Webster's New World Dictionary of Media and Communication, 1996
wikipedia	WikiWiki. In: Wikipedia. The Free Encyclopedia, 2002, <a href="http://www.wikipedia.com/wiki/Wiki">http://www.wikipedia.com/wiki/Wiki</a> (19.5.2002)
wikiservice	Wikiservice.at. Der Spezialist für deutschsprachige WikiWeb-Systeme. <a href="http://www.wikiservice.at">http://www.wikiservice.at</a> (19.5.2002)
wikiwikiweb	<i>Ward Cunningham (Hg)</i> : Portland Pattern Repository and WikiWikiWeb, 1995-2002, <a href="http://www.c2.com/cgi/wiki">http://www.c2.com/cgi/wiki</a> (30.4.2002)
wikiwikiweb- badges	Gently Reduce Wiki Badges. In: WikiWikiWeb, <a href="http://www.c2.com/cgi/wiki?GentlyReduceWikiBadges">http://www.c2.com/cgi/wiki?GentlyReduceWikiBadges</a> (5.5.2002)
wikiwikiweb- refactoring	Refactoring Wiki Pages. In: WikiWikiWeb, <a href="http://www.c2.com/cgi/wiki?RefactoringWikiPages">http://www.c2.com/cgi/wiki?RefactoringWikiPages</a> (11.5.2002)