

Flexible Website-Verwaltung mit PostNuke, Teil 1

Ein Modul für alle Fälle

von Axel Guckelsberger und Steffen Voß

Mit der aktuellen PostNuke-Version liegt ein solides wie flexibles Application Framework vor, mit dem man von einer kleinen Clan-Seite bis hin zum professionellen Shop vielfältige Seiten aufbauen kann. Mit dieser ersten Folge unseres dreiteiligen Workshops möchten wir Ihnen neben der Basisinstallation die verschiedenen Elemente des Systems anhand praktischer Beispiele vorstellen.

Durch seine Herkunft als Fork des Portalsystems PHPNuke wird PostNuke im Allgemeinen oft noch als „Nuke-System“ abgetan. Jedoch hat sich PostNuke weit von diesen Wurzeln entfernt. PostNuke setzt auf die durchgängige Abstraktion und den Einsatz offener Standards: Die Datenbankzugriffe laufen per ADODB, das Layout per Smarty Templating und die Interaktion der einzelnen Teile über das strukturierte eigene API. Mit PostNuke lassen sich XHTML-konforme sowie mehrsprachige Seiten erstellen, und das Entwicklerteam legt großen Wert auf Barrierearmut und Systemsicherheit. Ein nicht zu unterschätzender Faktor für Lernende ist die aktive Community. Sie hat sich in Deutschland in einem eingetragenen Verein und in den USA in einer Foundation organisiert und bietet qualitativen Support und permanente Hilfestellung für Einsteiger.

Die Installation

Schritt für Schritt werden wir in diesen Artikeln eine einfache Community-Site mit Forum und Benutzerprofilen, News usw. realisieren. Wir möchten Ihnen vor allem die Werkzeuge an die Hand geben, mit denen Sie selbst tiefer ins System einsteigen können.

PostNuke benötigt einen Webserver mit mindestens PHP 4.1 und MySQL 4.3.x, eine Ausstattung also, die so ziemlich jeder Webespace erfüllen sollte. PostNuke läuft aber auch auf den neueren Versionen von Apache, PHP und MySQL.

Die aktuelle Version (0.762) von PostNuke gibt es auf der Homepage des PostNuke e.V. Dort laden Sie auch das dazugehörige deutsche Sprachpaket herunter, entpacken beide Archive und laden daraus jeweils die Inhalte des Ordners *html* auf den Webespace. Diese Dateien und Verzeichnisse müssen jetzt Schreibrechte bekommen:

- *config.php*
- *config-old.php*
- *pnTemp/pnRender_cache*
- *pnTemp/pnRender_compiled*
- *pnTemp/Xanthia_cache*
- *pnTemp/Xanthia_compiled*

Anschließend rufen Sie die Datei *install.php* im Browser auf und starten somit den Installationsvorgang. Nun geht es mithilfe des Assistenten Schritt für Schritt weiter und wir geben einige grundlegende Daten ein, wie beispielsweise die Zugangsdaten für die Datenbank und den Administrator.

Ausführlichere Informationen bzgl. der Installation finden Sie im Online-Handbuch [3]. Am Ende der Installation werden Sie darum gebeten, einige Dateien des Installers aus Sicherheitsgründen zu löschen. Danach können Sie auch die Schreibrechte bei den beiden *config*-Dateien wieder entfernen. Jetzt ist das Grundsystem von PostNuke installiert und einsatzbereit.

Ein erster Überblick

Auf den ersten Blick ist das System karg: Das Layout ist eher praktisch als schön, und wenn wir uns als Administrator auf unserer

neu installierten Seite angemeldet haben, enthält das Menü auf der linken Seite nur einen weiteren Link zur Administration. Durch diesen Eindruck sollte man sich aber nicht täuschen lassen: Wie im wirklichen Leben liegen die wahren Werte auch hier tiefer verborgen.

Im Administrationsbereich finden Sie die Links zu den Admin-Funktionen der einzelnen *Module* – so nennt man die Bestandteile des Systems und deren Erweiterungen. Außerdem ist der Admin-Bereich kategorisiert, z.B. in *System-Module* oder *Werkzeuge*. Die Kategorisierung lässt sich frei anpassen und der Zugriff über Rechte reglementieren. Wir schauen uns zunächst die wichtigsten Systemmodule an, da diese die Grundfunktionen bereitstellen:

- Das Modul *Einstellungen (Settings)* ist der zentrale Sammelpunkt für alle Einstellungen und Eigenschaften, die die Seite allgemein betreffen. Hierzu zählen etwa der Name der Seite, die E-Mail-Adresse des Administrators oder Optionen bzgl. der Mehrsprachigkeit.

PostNuke-Reihe

PHP Magazin 5.2006:
Flexible Website-Verwaltung mit PostNuke

PHP Magazin 6.2006:
News-Anzeigen an das Layout anpassen

PHP Magazin 1.2007:
Content mit Pagesetter verwalten

- Im Modul *Module* werden die installierten Module verwaltet. Dies beinhaltet neben der Installation neuer Module auch deren Updates oder das Verändern des dargestellten Namens eines Moduls.
- Das *Mailer*-Modul erlaubt es, systemweite Vorgaben für den Versand von E-Mails einzustellen, z.B. kann eine E-Mail entweder mithilfe eines SMTP-Servers oder aber auch mit sendmail versendet werden. Die meisten Shared-Hosting-Umgebungen sind so vorkonfiguriert, dass hier keine Einstellungen nötig sind.

Für die Verwaltung der Benutzer sowie deren Zugriffsrechte sind weiterhin folgende Systemmodule relevant:

- In der Benutzerverwaltung lassen sich manuell neue Benutzer anlegen sowie bestehende bearbeiten. Außerdem lassen sich eigene Felder für das Profil der Benutzer mithilfe der dynamischen Benutzerdaten definieren.
- In der Gruppenverwaltung sind die Standard-Gruppen *User* und *Admin* voreingestellt. Daneben lassen sich beliebig viele

weitere Gruppen hinzufügen und verwalten. Über das Zusatzmodul *pnGroups* [4] kann man unter anderem den Benutzern ermöglichen, sich für Gruppen anzumelden. Mithilfe der Gruppen werden bei PostNuke übrigens auch die Zugriffsrechte auf einzelne Bestandteile des Systems festgelegt.

- Zu den mächtigen, aber auf den ersten Blick vielleicht etwas kompliziert anmutenden Elementen von PostNuke zählen dessen Zugriffsrechte. Sobald man es jedoch erfasst hat, lässt es sich umso vielseitiger einsetzen (für mehr Informationen siehe Extrakasten „Sie sind nicht berechtigt, diese Funktion zu benutzen.“).

Erweiterungen

Die mitgelieferten Module decken schon eine Reihe von Grundanforderungen: Ein einfaches News-Modul, FAQ usw. Da die PostNuke-Entwickler in den letzten Jahren vor allem Wert auf die Entwicklung des APIs, des Templating usw. gelegt haben, sind die Content-Module leider weitestgehend auf dem Stand der Funktionalität von PHPNuke stehengeblieben. Die wirk-

lich interessanten Module muss man sich zusätzlich herunterladen.

Um sich in dem Dschungel zurechtzufinden, hat der PostNuke e.V. eine Liste mit Modultipps [6] zusammengestellt. Hier sind wichtige Module zu finden, mit denen man die Arbeit beginnen kann. Die Module funktionieren alle in der jeweils aktuellen PostNuke-Version, haben eine gewisse Zukunftsperspektive, werden von den Mitgliedern des PostNuke e.V. selbst eingesetzt und können deswegen im Forum am besten unterstützt werden. Wer darüber hinaus Module sucht, wird sicherlich im PostNuke-NOC [7] fündig. Hier bietet die PostNuke Foundation Modulentwicklern einen ähnlichen Service wie zum Beispiel SourceForge.

Eine besondere Bedeutung innerhalb der Module kommt den Hooks zu: Diese kleinen Module können sich in andere Module einklinken. Dadurch werden Funktionen, die von vielen Modulen benötigt werden, zentral verwaltet. Dazu zählen etwa Kommentare, Smilies, BB-Codes, Uploads, Akronyme oder Benachrichtigungen. Die Hooks sind ein Beispiel dafür, dass Erweiterungen des PostNuke-

Sie sind nicht berechtigt, diese Funktion zu benutzen

Das Berechtigungssystem in PostNuke erlaubt es, die erteilten Berechtigungen spezifischen Anforderungen anzupassen. Für jede Gruppe, jedes Modul, bis hin zum einzelnen Artikel können Rechte vergeben werden. Die einzelnen Berechtigungen werden in einer Liste verwaltet, die von oben nach unten abgearbeitet wird, bis eine Regel auf den aktuellen Benutzer (bzw. dessen Gruppe) greift. Diese Regel wird dann angewendet, alle nachfolgenden Zeilen werden ignoriert; ergo ist die Reihenfolge der Berechtigungen wichtig für die gewünschte Funktionsweise.

Aufbau einer Regel

Eine Regel besteht aus mehreren Elementen, die in Kombination eine Berechtigung erteilen oder verweigern:

- Gruppe – die Gruppe, der die Regel zugeordnet werden soll
- Komponente – Modul oder Block, auf das oder den sich die Regel bezieht (z.B. „Menublock:“)
- Instanz – Instanz der spezifizierten Komponente

- Stufe – Eine von neun Berechtigungsstufen (z.B. *kommentieren, moderieren, administrieren*)

Der Name einer Komponente ist von dem jeweiligen Modul, für das man die Regel definieren möchte, vorgegeben. Die Bedeutung der aus zwei durch Doppelpunkte getrennten Teilen bestehenden Instanz ergibt sich dabei aus der Komponente. Da sich aus der Instanz ein regulärer Ausdruck ergibt, lassen sich an dieser Stelle flexible Definitionen erstellen:

```
alle Gruppen|Menublock::|Hauptmenü:(Administration|
Link1|Link2):|keines
```

Dieses Beispiel versteckt drei bestimmte Links im Menublock namens „Hauptmenü“ vor allen Gruppen. Damit man die Links als Administrator noch sehen kann, muss – wir erinnern uns an die Wichtigkeit der Reihenfolge – über dieser Beispielregel eine entsprechende Zeile stehen, die der Admin-Gruppe alle Rechte zuteilt.

Bzgl. der Stufen ist anzumerken, dass sich diese kumulativ verhalten. Das heißt, dass eine

Berechtigungsstufe alle niedrigeren Stufen beinhaltet. So kann ein Benutzer mit dem Recht „administrieren“ auch lesen, kommentieren und editieren. Wichtig zu wissen ist, dass der jeweilige Modulentwickler entscheidet, was genau eine bestimmte Berechtigungsstufe für das jeweilige Modul bewirkt. Dies bedeutet, dass z.B. das Recht „moderieren“ in zwei verschiedenen Modulen unterschiedliche Auswirkungen haben kann. Das macht das System auf der einen Seite nicht leicht zu verstehen – aus der Sicht des fortgeschrittenen Anwenders oder des Entwicklers ist es jedoch sehr flexibel.

Für ausführlichere Anmerkungen und weitere Beispiele sei auch hier der entsprechende Abschnitt des Online-Handbuchs [5] ans Herz gelegt. Und zu der genauen Rechtevergabe in Zusatzmodulen sollte die jeweilige Dokumentation befragt werden. Normalerweise verhält sich ein Modul in der Standardkonfiguration so, wie man es erwartet. Man kann also durchaus eine Site betreiben, ohne mit dem Berechtigungssystem in Berührung zu kommen.



Abb. 1: Die PostNuke-Supportseite



Abb. 2: Jugendferienbörse Schleswig-Holstein

Systems nicht wie Bauklötzchen über- und nebeneinander gestapelt werden. Es gibt verschiedene Schnittstellen, über die Module interagieren können.

Die Modulverwaltung läuft ebenfalls über die Administration. Jedes Modul, für das es etwas einzustellen gibt, hat ein eigenes Icon in der Administration. Wie die Module aus der Basisinstallation im Einzelnen funktionieren, kann man leicht selbst herausfinden oder im Handbuch [2] nachlesen. Wir wollen uns in diesem Artikel auf ein paar Module beschränken, die wir für unser Community-Beispiel benötigen.

Themes, Templates

Das Aussehen einer Site wird in PostNuke über Templates geregelt. Templates sind einzelne Dateien, die einfaches HTML mit Aufrufen für die Inhalte beinhalten. Templates bieten den Vorteil, dass jeder Webmaster ohne Hacken der Moduldateien

nicht nur das Aussehen, sondern auch den Funktionsumfang der Module beeinflussen kann. Wer zum Beispiel nicht die Größe eines Downloads anzeigen lassen will, kann den entsprechenden Aufruf einfach aus dem Template löschen. Und Zusätze wie Druckansichten und RSS-Feeds sind dann auch ohne Programmierkenntnisse schnell umgesetzt.

Es gibt Templates, die die Grundstruktur der Seite festlegen (*Theme*), aber auch Modul- und Block-Templates. Die Verwaltung der Themes und Templates geschieht mithilfe der Templating-Engine *Smarty* [8], die in Form von zwei Unterklassen in PostNuke integriert wurde: *pnRender* für Module, *Xanthia* für Themes. Damit wurde PostNuke um Features wie das Caching von Inhalten bereichert, vor allem aber wurde eine Möglichkeit für die konsequente Trennung zwischen Funktion und Form erreicht. In PostNuke .8 wird die künstliche

Trennung zwischen Basis-Layout und Modultemplates aufgehoben und alle Templates werden vom kompatiblen Theme-Modul verwaltet. Fertige Themes kann man sich zum Beispiel unter [6] herunterladen. Dank der einfachen Smarty-Syntax lassen sich aber auch beliebige HTML-Vorlagen in relativ kurzer Zeit in Themes verwandeln. Bis Sie im zweiten Teil des Workshops erfahren, wie Sie aus jeglicher Vorlage ein eigenes Theme erstellen, können Sie in der Administration/*Xanthia* eines der anderen mitgelieferten Themes aktivieren und in den Einstellungen als Haupt-Theme setzen. Schon hier können Sie erahnen, wie flexibel Sie Ihre eigene Seite gestalten können. Die beiden Screenshots illustrieren, wie unterschiedlich PostNuke-Seiten mit *Xanthia* aussehen.

Themes werden im Unterverzeichnis */themes/* gespeichert, die Templates der einzelnen Module – so sie denn *pnRender*

unterstützen – liegen unter `/modules/ModulName/pnTemplates`. Deren Optik lässt sich auch ohne PHP-Kenntnisse einfach an die eigenen Wünsche anpassen, indem die Templates mit einem Editor bearbeitet werden. Dafür reichen HTML-Kenntnisse völlig aus.

Um eigene Änderungen beim nächsten Update nicht zu verlieren, lassen sich Modul-Templates auch ins Theme-Verzeichnis kopieren (`/themes/MeinTheme/templates/modules/Modulname`) und dort anpassen. Auch sämtliche anderen Anpassungen können im Theme hinterlegt werden, sodass die Dateien der Grundinstallation nicht gestört oder manipuliert werden müssen. Auch kleine Funktionen wie etwa eine Zufallsauswahl für die Header-Grafik können Sie mithilfe von Plugins leicht im Theme erledigen. Weiteres zu den Möglichkeiten, die durch das Templating erreicht werden, werden wir in den folgenden Teilen dieses Workshops eingehender ansprechen.

Baut auf! Baut auf!

Nachdem wir uns nun einen Überblick über die Systemkomponenten verschafft haben, wollen wir endlich unsere eigene Community aufbauen. Vor allem möchten wir die Prinzipien präsentieren, nach denen Sie sich beliebige Module installieren und so Ihre eigene Seite zusammenstellen können. Für unsere Community und als Basis für die weiteren Teile des Workshops benötigen wir: ein Forum, ordentliche Benutzerprofile, verschiedene Benutzergruppen und News mit der Möglichkeit, Bilder zu integrieren.

Als Forum schlagen wir *pnForum* [11] vor. Es gibt zwar auch Brücken zu phpBB2 oder Simple Machines Forum, jedoch ist pnForum als einziger Kandidat vollständig in PostNuke integriert, nutzt die Benutzerverwaltung, die Zugriffsrechte, das Templating und ist mit den erwähnten Hooks vielfältig und individuell erweiterbar. Es ist damit ideal für Webmaster, die ein schlankes Forum benötigen und sich nur die benötigten Funktionen zusätzlich installieren wollen.

Die eingebauten Benutzerprofile sind leider noch nicht auf Templates umgestellt, sodass wir an dieser Stelle *advProfile* [4] empfehlen möchten. Hier lassen

sich die Profile an das eigene Layout anpassen und per Plug-ins um Funktionen erweitern, wie beispielsweise eine Anzeige der letzten zehn Beiträge des Benutzers im Forum.

Mit dem schon erwähnten *pnGroups* [4] kann man nicht nur den Benutzern erlauben, sich für bestimmte Gruppen anzumelden oder zu bewerben, man kann auch in Listenansichten darstellen, welche Mitglieder eine Gruppe hat.

Auch bei den Galerien stehen verschiedene Alternativen zur Auswahl. Es gibt Brücken zur Menalto Galerie und zur Coppermine-Galerie. Als interne Verwaltung von Artikel-Bildern eignet sich am besten das *safe_mode*-kompatible *MediaShare* [12], da es neben Bildern auch andere Medieninhalte (z.B. Videos) verwalten und in verschiedene Content-Module integriert werden kann.

Da die normalen News aus dem Basispaket nicht viel zu bieten haben, möchten wir schon im ersten Teil des Workshops das Universalmodul *Pagesetter* [12] installieren und die vorkonfigurierte News-Funktion nutzen. Mehr zu den vielfältigen Fähigkeiten von Pagesetter erfahren Sie im dritten Teil unseres Workshops.

Nachdem Sie sich die Module von den verschiedenen Webseiten heruntergeladen und entpackt haben, werden Sie feststellen, dass es leider keinen Standard für die Auslieferung von Modulen gibt. In den Zip-Archiven sind verschiedene Verzeichnisse gespeichert. Schnell werden Sie aber lernen zu erkennen, welches das eigentliche Modul-Verzeichnis ist: Es heißt wie das Modul und enthält Dateien, die *pninit.php*, *pntables.php*, *pnuser.php* usw. heißen. Dazu gibt es zum Beispiel Verzeichnisse für die Sprachdateien (*/pnlng*) und die Templates (*/pntemplates*) – sollte es ein Verzeichnis */docs* oder */pndocs* geben, finden Sie hier übrigens die Modulanleitung.

Dieses Modulverzeichnis laden Sie mit dem gesamten Inhalt in Ihr *modules*-Verzeichnis auf dem Webserver. Als Nächstes gehen Sie in die Administration Ihrer Seite und rufen dort die Modulverwaltung (MODULE) auf. Mit NEU ERZEUGEN sucht die Verwaltung nach den neu installierten Modulen. Wählen Sie aus dem Dropdown rechts NICHT INITIALISIERT aus, um auf die Module mit diesem Status zu fil-

Beste Bücher für besten Code!



Christian Wenz

Reguläre Ausdrücke

103 Seiten, Softcover, 7,90 €
ISBN 3-935042-90-6
Juni 2006



Markus Nix

Ruby und Rails

ca. 80 Seiten, Softcover, 7,90 €
ISBN 3-939084-08-5
Juli 2006



Sissi Closs

Single Source Publishing

ca. 300 Seiten, Softcover, 34,90 €
ISBN 3-935042-98-1
September 2006

tern und initialisieren Sie die Module, die Sie gerade hochgeladen haben. In diesem Schritt werden die nötigen Datenbanktabellen angelegt. Neuere Module nutzen die Möglichkeit einer interaktiven Installation und fragen schon nach den ersten Einstellungen. Danach stellen Sie den Filter auf Module mit dem Status *inaktiv* und aktivieren die neuen Module. Nun stehen Ihnen die Funktionen derselben zur Verfügung und Sie werden die entsprechenden Icons in der Administration finden.

Verlinken

Um im Navigationsmenü Links auf die Module setzen zu können, sollten Sie den Aufbau von Links in PostNuke kennen:

- *index.php?module=ModuleName* – für die Userseite
- *index.php?module=ModuleName&type=admin* – für die Adminseite

Sie können das einmal für pnForum ausprobieren, in der Administration die Verwaltung der Blöcke aufrufen und dort das Hauptmenü editieren: als Erstes schalten Sie den Punkt ANZEIGE | MODULE aus und tragen dann in der Liste mit den Menülinks die URL zum Forum ein.

index.php?module=pnForum – als Titel können Sie zum Beispiel FORUM eingeben. AKTUALISIEREN klicken, fertig: Der Link aufs Forum erscheint im Menü. Sie können nun anfangen und in der Verwaltung von pnForum die gewünschten Kategorien und Foren anlegen.

Um in den Genuss der besseren Benutzerprofile zu gelangen, müssen wir doch noch eine Datei hacken – in zukünftigen PostNuke-Versionen wird das nicht mehr notwendig sein, da advProfile in PostNuke .8 das alte Profil-Modul ersetzt. In der Datei *modules/Users/user.php* wird die komplette erste Funktion ersetzt:

```
function user_user_userinfo()
{
...
}
```

Und zwar durch diese Funktion:

```
function user_user_userinfo()
{
```

```
$uname = pnVarCleanFromInput('uname');
pnRedirect(pnModUrl('advProfile', 'user', 'main',
array('uname' => $uname)));
}
```

Wenn Sie nun testweise einen Beitrag im Forum schreiben und dann dort auf den Link zum Benutzerprofil klicken, werden Sie sehen, dass Sie automatisch auf das erweiterte Profil des Benutzers von advProfile weitergeleitet werden.

Erste News mit Pagesetter zu veröffentlichen ist nicht schwer und lässt sich im Pagesetter-Handbuch nachlesen. Im Teil 2 unseres Workshops erfahren Sie, wie Sie die Anzeige der News optisch an Ihr Layout anpassen, und im dritten Teil erfahren Sie, wie Sie eigenen Content mit Pagesetter verwalten. Am Ende werden Sie nicht mehr auf dieses vielseitige Modul verzichten wollen.

Ausblick

Wir haben Ihnen bisher gezeigt, aus welchen Teilen PostNuke besteht und wie Sie zusätzliche Module installieren. Wir ha-

ben damit auch die Basis für die weiteren Teile des Workshops gelegt.

Im zweiten Teil werden wir eingegebenes Layout in ein Xanthia-Theme umsetzen und außerdem verschiedene Modul-Templates an unsere Anforderungen anpassen. Der Schwerpunkt wird also darin bestehen, praktisch auf die Vorteile durch pnRender/Xanthia zu sprechen zu kommen.

Der dritte Teil wird schließlich ein wenig mehr unter die Haube von PostNuke schauen und aufzeigen, wie einfach sich eigene Erweiterungen und Funktionalitäten realisieren lassen. Die flexible Handhabung der pnAPI wird hier im Mittelpunkt stehen. Außerdem beschäftigen wir uns näher mit dem Universal-Contentmodul Pagesetter, mit dem man mit ein paar Mausklicks eine Rezeptdatenbank oder eine Kontaktbörse zaubern kann.



Axel Guckelsberger und Steffen Voß sind im deutschen PostNuke-Team tätig. Außerdem sind sie Mitglied im deutschen PostNuke e.V. und engagieren sich als Entwickler und Organisatoren für die Community.

Links & Literatur

- [1] Beispiel: Der Gott bin ich Blöd-Clan: www.gbib-clan.de
- [2] Beispiel: T-Com IT-Shop: it-shop.t-com.de
- [3] Installation von PostNuke: support.pn-cms.de/modules/dokuwiki/doku.php?id=handbuch:systeminstallation
- [4] pnGroups, advProfile: dev.pnconcept.com
- [5] Zugriffsrechte in PostNuke: support.pn-cms.de/modules/dokuwiki/doku.php?id=handbuch:zugriffsrechte
- [6] PostNuke Themes: themes.pn-cms.de
- [7] PostNuke Developers Center: noc.postnuke.com
- [8] Smarty – Template Engine: smarty.php.net
- [9] Die PostNuke-Supportseite: support.pn-cms.de
- [10] Beispiel: Jugendferienbörse Schleswig-Holstein: www.ferienboerse-sh.de
- [11] pnForum: noc.postnuke.com/projects/pnforum/
- [12] Mediashare/Pagesetter: www.elfisk.dk
- [13] offizielle deutsche Webseite von PostNuke und dem PostNuke e.V.: www.pn-cms.de
- [14] Modultipps: support.pn-cms.de/pagesetter-tid-2.htm
- [15] Modulempfehlungen: support.pn-cms.de/pagesetter-tid-2.htm
- [16] PostNuke-Dokumentation: support.pn-cms.de/modules/dokuwiki/doku.php?id=handbuch:index
- [17] Markus Gossmer, Michael Schumacher, Andreas Schauerl: „Das PostNuke Kompendium“, Springer 2005, ISBN: 3540219420
- [18] Kevin Hatch: „PostNuke Content Management“, Sams 2005, ISBN: 0672326868