

# Embarking on Web Content Management Projects

Ever since the advent of the World Wide Web in the early 1990s, there has been a focus on publishing information. Indeed, the very first Web sites were set up by scientists at CERN, the European Organization for Nuclear Research, so physicists around the world could publish information in a consistently accessible way. Since then, the Web has moved to more than publishing; this started with transactional Web sites, and led to collaboration, social networking, and aggregation-focused sites, to name a few, and all of these are addressed by Microsoft Office SharePoint Server (MOSS) 2007, if not by this book.

Even as the technology has evolved, the need for Web publishing remains pervasive. For example, transactional Web sites publish catalogs and terms of sale; collaboration and social networking sites publish usage guides and ground rules. Therefore, Web publishing remains a core function of any public, extranet or intranet Web site, even if it is more than just “brochureware.”

Take a moment to consider this book, which is the product of a modern and technically advanced publishing company. In addition to the authors, there are many other contributors to this book. Someone selected the topic as part of the publisher’s catalog and developed the title and “brand” for the book; other people designed the cover and page layout; editors checked for quality and consistency, and still other people typeset and printed the book.

Publishing a Web site is no different: People in specialized roles each want to control particular aspects of the final product. However, unlike a book, the Web site is being constantly updated, and people associated with the site want the freedom necessary to change their aspects of the site without affecting one another. For example, an author may want to add a new page, an editor may want to reorganize several pages, and a branding manager may want to change the colors and logo of all pages, all at the same time. The final “product” — a connected set of Web pages — needs to reflect the input of each of these contributors at any given point in time.

# Chapter 1: Embarking on Web Content Management Projects

---

This is the problem solved by Web Content Management (WCM). A WCM system organizes the content and design from all of the site's contributors, allows for versioning, editing, and moderation, and stitches it all together for the end user.

Consider a typical Web page. The banner, color scheme, and general look and feel are part of the branding of the site. Some sort of navigation is probably visible, revealing the organization of the site. In addition to the site navigation, there may be listings of content such as a "front page" list of articles or other topics. These are another form of navigation, one which cuts across the formal structure of the site to highlight contextually relevant content. Authored content — that is, content written by an author and possibly run through an editorial process — may appear in one or more sections of the page, along with images that may require acquisition and approval. Syndicated content, such as news feeds and advertisements, might also appear. Down at the bottom, in the fine print, there may be a legal notice or other disclaimer. Within a typical organization that has a Web site, different people will want to manage each of these aspects of the same Web page, all while the site is up and serving customers.

In the bad old days, the approach to managing a Web site was to edit Web pages and associated files on the file system of each Web server. This approach is simple enough at first, but makes it very hard to modify things such as branding, navigation, or legal disclaimers that appear repeatedly on many Web pages. Moreover, if the authored content is stored in the same files as branding, navigation, and other page features, in the course of editing a paragraph an author could accidentally modify the wrong thing and break the page entirely. This led to the role of Webmaster, a person to whom all Web site content and other changes are fed, and who knows the intricacies of HTML and CSS and any other page programming. The Webmaster's job quickly became a tedious one — copying, pasting, and reformatting content submitted via e-mail and in documents. As the single gatekeeper for all aspects of a Web site, Webmasters often were seen as bottlenecks by contributors whose changes had to wait at the end of the queue.

As a WCM system, MOSS 2007 provides flexibility and independent control over all these aspects of a Web page. The Webmaster bottleneck is largely eliminated by giving control over the many aspects of a site directly to business users, information architects, developers, and designers. Instead of endlessly copying and pasting content, Webmasters can focus on system administration, site design, and development, enabling them to have a much greater impact than ever before.

## The Web Content Management Experience

To better understand the use cases for WCM, this section will follow a couple of typical users through their interaction with the system. At first the focus is on authors and editors who produce the authored content on the site. Then the focus shifts to the role of designers and developers, who have a very different kind of interaction with the site.

### ***Authors and Editors***

The scenario opens with a product marketing manager who is about to launch a new product and wants to add some information about it to her company's Web site. Rather than having to ask someone to do this for her, she can simply edit the Web site directly, as shown in Figure 1-1. She navigates to the Web site (possibly using a special, internal URL that allows the necessary authentication) and adds a few pages. She enters the product information based on her knowledge as product manager, and is able to

# Chapter 1: Embarking on Web Content Management Projects

format the text as she likes, as long as she stays within company style guidelines. Next, she posts some product images to the site's image library and uses them in the pages as well. She sets each page's start date to the product launch date, which is a couple of weeks in the future, and submits the pages for approval.



Figure 1-1

Next, an editor gets an e-mail notifying him that there are new Web pages awaiting approval. He clicks a URL in the e-mail and is led to the pages the product manager just created. He makes a few minor changes and approves the pages. The author is then notified that the pages have been approved, and she checks them over one more time to ensure that they look right. However, only she, the editor, and a few other privileged users can see the pages; the general public will get to see them when the product is launched.

On the product launch day, the marketing manager is busy at a big press event, but knows that her new Web pages went live at 8:00 A.M. that morning, and that she has carefully reviewed them in her own Web browser. Furthermore, the pages will automatically appear in the site navigation and on summary pages at the moment the pages go live.

### **Designers and Developers**

The new product was so successful that the company has decided to expand, and has just merged with one of their best partners. The merger will result in a new company, with a new brand. A complete overhaul of the Web site is in order. All the existing Web site content is still relevant — it just needs a new look and branding.

The marketing manager hires a Web design firm to help them create their new site. An information architect draws up a series of wire-frame diagrams showing the new page layouts, while a graphics designer starts working on a look for the new site in the form of mock-ups, which are really just image files showing what the site will look like. Ultimately, the marketing manager agrees to an approach, and chooses one of the mock-ups and a set of wire frames for the new site.

Because the designer's mock-up is just an image file, the next step is to translate it into HTML with Cascading Style Sheets (CSS) and a bunch of smaller image files that will make up each kind of page on the site, resulting in an HTML mock-up. So far, the steps for the new MOSS site branding resemble those that would be used in any Web site.

Next, a SharePoint developer takes the HTML mock-up and merges it with a SharePoint blank *master page* she downloaded from the Microsoft Web site. This provides the basis for all the newly branded Web pages, with placeholders for all the SharePoint functionality to show through. She packages the master page, CSS, and supporting image files into a SharePoint *solution package* and checks it into the source control system.

Meanwhile, the information architect wants to change the page layouts to match the new wire-frame diagrams. To do this, she creates (or modifies existing) MOSS *page layouts*, which define where *fields* should appear on each type of page. For example, she might specify that all product pages should have a title at the top, followed by the model number and list price, and that the main body of the page will have a large floating image on the right. A developer translates this into a page layout, which contains an HTML fragment with the structure of the page. The HTML fragment includes *field controls*, which alternately render or allow editing of the content. The properties of each field control define what text formatting and CSS styles are permitted in each field.

The master page and page layouts could be developed using SharePoint Designer 2007 directly on the production servers (so they would be treated as part of the site content), or they could be developed using Visual Studio and included in a solution package. The fundamental difference between these approaches is that a Visual Studio solution package is installed on the file system of each of the SharePoint Web servers in a farm, whereas SharePoint Designer modifies the SharePoint content database. SharePoint Designer can *customize* SharePoint pages, which means that at runtime, the customized page from the database is substituted for any original page that may have been on the Web servers (or the customized page may only exist in the database, with no file-system-based counterpart at all). Because content is not subject to the same release cycle as code, this often means that SharePoint Designer customizations — which are really content changes, as they are in the content database — are put directly into production, whereas SharePoint solution packages can go through a regular software development life cycle, including controlled releases through a formal testing process. The latter is the recommended approach. Keep in mind that SharePoint Designer is still useful as a developer tool, enabling developers to start with customized pages on a development server, and then migrate the markup into Visual Studio; it is also useful to allow business users to customize their sites directly.

# Chapter 1: Embarking on Web Content Management Projects

In the scenario, it is determined that the new branding on the site is core functionality and that it should be subject to a controlled release cycle. The SharePoint solution package is part of a Visual Studio solution stored in a source control system, and installed on servers in development, quality assurance (QA)/staging, and ultimately production. In this case, the package deploys the master page, page layouts, and dependent files to every Web server in the farm. Once the new master page is enabled, the new branding appears on all the site's Web pages. The new page layouts also take effect, even on existing pages if they used the same layout name and set of fields (or *content type*).

After thorough testing, development releases the new solution package to a system administrator who installs it on the production SharePoint farm and switches the site to use the new master page. Immediately, the new look and feel takes effect, even on existing pages. In other words, a new site was not really needed after all — the infrastructure and existing content were all reused. Only the branding and layout were changed, without affecting other aspects of the Web site.

Developers can go much further in customizing and extending the WCM system, using solution packages that are listed in the following table:

Extension	Description	Example of Use	Chapter
Custom site definitions	A set of XML-based instructions for creating a new site with specific settings and initial content. Note that in MOSS, a <i>site</i> refers to a container in a larger <i>site collection</i> , not what users consider to be a whole "Web site."	A starting point for a specific class of site, such as a product information site, a job posting site, a business partner site, etc.	5
Custom site columns, content types, and lists	A set of XML-based instructions for creating new site columns (to store specific kinds of information), content types (groups of columns to use in a list or library), including when a site is created or a Feature is activated	For a product information site, a site column could be created for each product attribute to be displayed, and a new "product page" content type would be created with these and some of the built-in columns.	6
Master pages and page layouts	As described above, master pages provide the branding and overall page structure, and page layouts define the position of field controls on the page.	For a product information site, there might be a master page with the overall branding, and a page layout defining how the product page's columns should be displayed.	7

Table continued on following page

## Chapter 1: Embarking on Web Content Management Projects

---

Extension	Description	Example of Use	Chapter
Custom navigation	A new and different way to render site navigation, or custom links added to the navigation across many pages or sites	A new MOSS WCM site is being set up alongside an existing site that is not going to be changed; navigation for the old site needs to be stitched into the MOSS site to make a consistent user experience.	8
Custom field types and field controls	A custom field type is a new data type, which may inherit one of the existing types, and can be used in site columns. A field control is an ASP.NET Web control that renders and allows editing of a particular field type.	Fields requiring custom storage and/or rendering might lead to a custom field type, such as a page rating or a compound part number. Field controls are used for custom and out-of-the-box field types whenever any special rendering or editing is required. For example, a complex field might be stored as XML, edited using a grid control, and rendered using an XSLT style sheet.	10
Custom Web Parts	An ASP.NET Web control that renders external or internal site content	Display external information such as a weather report or stock feed; display lists of items or search-driven results.	11
Custom workflow and forms	A workflow definition for managing content and content-related work, with forms to capture information from end users	A more complex approval workflow than is built into MOSS, with special business rules or exceptions.	12
Custom Authoring Console	Additional menu options or formatting capabilities when editing Web pages	A simple option is added to perform common formatting, content insertion, or other tasks on the page.	14

# Designing and Planning a Successful WCM Solution

Anyone who has tried to find their way around a new city knows that the experience varies considerably between a planned city, such as most of New York or Washington, DC, versus an unplanned city such as London or Boston. A planned city has some kind of logical organization or grid layout that makes navigation much easier, whereas an unplanned city may seem like a twisting maze of confusing passages, presenting quite a challenge for newcomers. A Web site is no different, and a little planning and forethought will go a long way in terms of site usability and control.

*Information architecture* is the emerging field of designing the structure of shared information environments, such as Web sites, to improve usability and facilitate the finding of information. Professional information architects often have a background in library science or cognitive psychology; and depending on the scope of a project, it may be desirable to enlist such a specialist to lead the effort. On smaller projects, or projects for which the basic structure is well understood, the information architecture may be created by the analysts and design team, with input from developers, users, and other stakeholders.

It is important to remember that *every WCM solution has an information architecture*, whether it is designed to be compelling and intuitive or left to evolve randomly.

MOSS comes loaded with a very simple information architecture and visual design that is set up when a Publishing site is created. The out-of-the-box Publishing site definition offers a choice of about half a dozen master pages, basic layouts for welcome and article pages, and a simple navigation structure with a home page and a child site for press releases. This is great for getting a quick start, and indeed some of the default settings may prove useful. However, it is worth at least considering each aspect of the WCM solution and deciding which of the defaults to keep and which to augment or replace.

The sections that follow describe the major areas to consider, along with general best practices and ideas about how to get started.

## Use Case Scenarios

As with any system, the first step is to identify the site's target users and determine the major use cases for the site — for example, “a reseller looks up product specifications,” “a patient looks up side-effects of a prescription drug,” or “a recruiter posts a new job opening.” Answering this question should produce an idea of who the users are, their degree of expertise, and what it is that they want to do. Don't forget to include internal as well as external users, such as content contributors and editors. Prioritize the uses cases and make them the basis for the decisions that follow.

## Site Structure and Navigation

Navigation begins with the overall site structure or a site map, and this is a good place to start the information architecture. SharePoint sites are always part of a *site collection*, which has a single parent or *top-level site*. The top-level site can have any number of child sites, grandchildren, and so on, and each site will have one or more *pages* in a *page library*. The site structure is therefore based on pages (the leaf

# Chapter 1: Embarking on Web Content Management Projects

nodes, which are viewable by end users) and sites (the containers, each of which has at least one default page to display). This is illustrated in Figure 1-2.

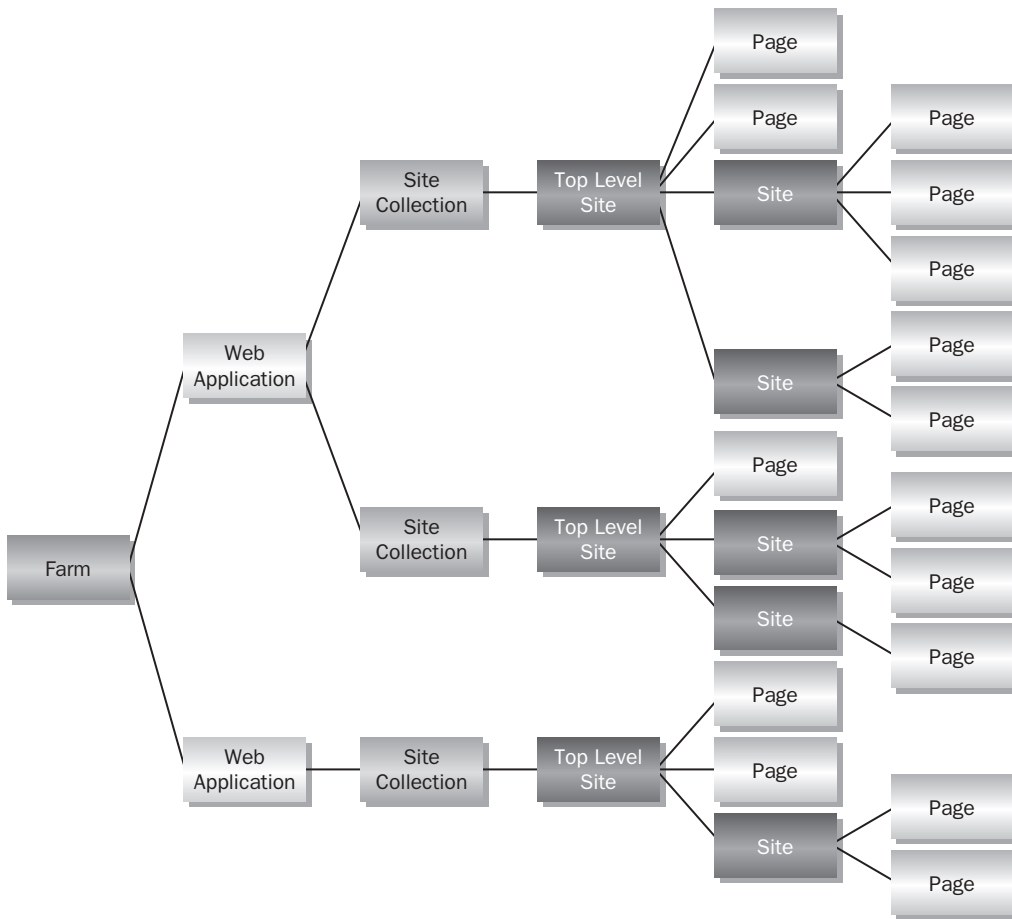


Figure 1-2

Begin with the logical site structure or site map. The site map is generally determined based on weighing a number of considerations, including the following:

- ❑ **What is the most logical browsing structure for end users?** This is certainly the most important consideration, but not an end-all, as other forms of navigation can short-circuit the site's structure, such as the Content Query Web Part (these are discussed in general later in this section and in greater detail in Chapter 11, "Web Parts").
- ❑ **What permissions will be set on the sites?** By default, SharePoint sites inherit their permissions from their parent site, so grouping sites with the same permissions (who can author, edit, etc.) under a common parent simplifies managing the permissions.



- ❑ **What is the look and feel of the sites?** By default, SharePoint sites inherit their master page settings and Cascading Style Sheets from their parent site, so grouping sites with the same look and feel under a common parent simplifies configuration and provides a more consistent browsing experience.
- ❑ **What information might need to be queried within the sites?** This refers not to search queries, which can span site collections and even external content, but to the Content Query Web Part and similar Web Parts developers create using the `SPQuery` object in the Windows SharePoint Services (WSS) object model. These queries are more like database queries that retrieve items based on their location, type, and property values, as long as they are within the same site collection. For example, a content query might locate Web pages on a particular topic, events in a particular location, or tasks assigned to a particular user. Content queries can be set to scan a site and all its descendants, so if the content to be queried is grouped within the same site hierarchy, the queries are easier to set up and maintain.

## Site Collection Boundaries

In many cases, the entire site structure can be contained in a single site collection, and indeed this is convenient because the built-in navigation is based on the sites and pages within the collection. In addition, content queries, content types, storage quotas, and numerous other SharePoint capabilities are scoped at a site collection, so there is a tendency to design large site collections to make them work over a large set of content. However, sometimes it makes sense to break the solution down into multiple site collections. The primary reasons are as follows:

- ❑ A site collection is always stored in a single SharePoint content database, although a content database can contain many site collections. If the site collection becomes too large — this includes all the content, such as documents, images, and videos, along with the Web pages — then the database can become unwieldy and hard to back up and restore in a timely manner. Therefore, many SharePoint administrators limit site collections to 50–200GB, and place large site collections in their own, dedicated content database.
- ❑ SharePoint’s built-in groups (for permissions) are scoped at the site-collection level, so if separate sets of groups are desired for administrative control, separate site collections will be necessary.
- ❑ Some SharePoint Features are scoped at the site collection level, and if these Features are desired in some areas but not others, then the areas need to be in different site collections. For example, the WCM capabilities are controlled by a site collection Feature called “Office SharePoint Server Publishing Infrastructure,” so team collaboration sites that don’t need WCM Features could be kept in a separate site collection.
- ❑ Anonymous access is scoped at the site collection level, so if part of a Web site is to be open to anonymous users, whereas another part forces a login, these sections should be in separate site collections.

SharePoint makes it very easy to reorganize sites within a site collection, but not so easy to move them to a new site collection, so it is worth thinking through the site collection boundaries up front. A common pitfall is to build a solution with one giant site collection and then find out months or years later that the database has become too large to restore from backup within the service-level agreement, or that very expensive backup solutions are needed to handle it. If a site will contain large items such as videos, then consider putting them in separate site collection(s) and linking to them to divide the storage.

## Navigation and Page Listings

The built-in navigation is based on the site hierarchy and the pages within each site in a site collection. Two views of the navigation are shown: *global navigation* (by default at the top of the page, the global navigation starts at the top-level site, or wherever the inheritance is broken) and *current navigation* (by default on the left of the page, the current navigation starts at the current site). It is easy to add arbitrary links to the navigation, or to hide sites and pages that should not be displayed. The navigation system can be fully customized; for details see Chapter 8, “Navigation.”

Two other types of “navigation,” which are really just ways of listing links to relevant pages, are also provided out of the box (OOTB):

- ❑ **Summary Link fields** — These manage a list of hyperlinks as content, and provide navigation to related pages, both within and beyond the WCM solution. These hyperlinks become part of the “authored content” and are thus subject to the normal page approval workflow.
- ❑ **Content Query Web Parts** — These dynamically query a site collection for content matching key criteria, such as “press releases issued this year” or “events in Oklahoma.” They provide links to cross-sections of site content based on the query settings, and have the advantage that they are updated automatically as new content is added.

Combined with the built-in navigation, these features enable placement of relevant links. In general, try to logically group links, and keep in mind that people tend to stop reading after the first four or five links in a list. In addition, they will only click a few levels deep, so this can limit the practical size of the site map. If the site map becomes unwieldy, consider breaking the solution down into multiple Web sites or *Web applications*, each with its own URL, and enable users to begin by selecting an appropriate starting point (e.g., sales information versus support). The same SharePoint farm can host these applications and provide a common search infrastructure that spans them all, so choose a structure that is logical from a usability and maintenance point of view.

## Page Layouts and Content Types

Just as navigation planning begins with the site map, the page layout planning begins with page wireframes. A *wireframe* is simply a sketch indicating how information will be laid out on each type of page. This includes *welcome* pages (as they are called in the MOSS-provided templates), which are intended to provide summary information on a site or section of a site, and *detail* pages (“article” detail pages are in the MOSS-provided templates).

In general, for each type of page identified in the wireframes, there is a single page layout. The page layout consists of an HTML fragment that defines the layout itself (often this is an HTML table). Places for content are inserted as needed within the HTML, and can include any combination of the following:

- ❑ **Field controls** — Each field control displays a piece of content that is stored in a field of each page that uses the page layout. The content in these fields is approved and versioned along with the rest of the page. Field controls are provided for HTML content (and text), images, and summary links; and as shown in Chapter 10, “Field Types and Field Controls,” it is easy to create your own for other types of information. Field controls are the most common way to present content in a MOSS WCM site.

- ❑ **Web Part zones** — Web Part zones enable authors and editors to place Web Parts on the page. This is very useful for adding functionality, such as Content Queries, RSS feeds, or KPI lists. However, note that Web Parts are part of the ASP.NET 2.0 infrastructure, and therefore are not aware of the MOSS publishing system.

A common confusion arises because Web Part zones store the Web Part placements and metadata in a separate Web Part store, not in fields of the page. As a result, changes to Web Parts within a zone are subject to approval along with the page, *but not to versioning*. Therefore, the expected approval behavior will work fine, but if a page is rolled back to a previous version, the Web Part zones won't be affected. For this reason, Web Parts are best used to add functionality, rather than pure content to a page. For example, the Content Editor Web Part might seem to be the same as a Rich HTML field control but it is not, because the Web Part's content is not versioned along with the page.
- ❑ **Web Parts and ASP.NET controls** — Web Parts and other ASP.NET controls can be embedded directly in the page layout. As such, they cannot be edited on the individual page instances; this can be quite useful to ensure that certain information is always displayed in a certain way.

Page layouts are closely related to the use of content types. A *content type* is simply a set of columns and policy settings that define some kind of content, be it a catalog page, a contract document, or a calendar event. Content types can share columns — for example, an article page, a press release page, and a product description page might all contain an Author column, and by mapping them all to the same Author site column, MOSS realizes that they are the same information when performing queries, constructing views of content, and so on.

The fields available for use in a page layout depend on its underlying content type. It is entirely possible to have multiple page layouts for a single content type; for example, the built-in MOSS page layouts include three layouts (`ArticleLeft.aspx`, `ArticleRight.aspx` and `ArticleLinks.aspx`), which are all layouts of the same content type ("article page"). This enables the page layout to be changed independently of the content — for example, an article page could be changed from `ArticleLeft.aspx` to `ArticleRight.aspx` in order to move its image to the right of the main text on the page.

It is useful to group the page layouts from your wireframe into content types, and to identify the site columns that will make up the fields of each page layout. If multiple page layouts can share a content type, this will provide more flexibility later. Keep in mind that some page layouts may not expose all the fields in the content type. Moreover, it's possible to put field controls inside an Edit Mode Panel control to make them appear only when the page is edited. This can be useful to capture metadata that will be shown in summaries (via the Content Query Web Part) without showing it on the detail page.

For planning purposes, understanding these distinctions and sketching out the wireframes should be sufficient; for details about how to implement page layouts, see Chapter 7, "Master Pages and Page Layouts."

## Supporting Content: Images, Attachments and Reusable Content

MOSS field controls will place much of the page content directly in fields of the page, but there will inevitably be other content that needs to be managed outside of the page. For example, the image field control does not store an image but a link to an image that is stored elsewhere. Likewise, a Web author

# Chapter 1: Embarking on Web Content Management Projects

---

may want to include attachment links to downloadable documents, but where do the documents themselves reside?

The natural answer is to store this supporting content right in MOSS, where it will be easily accessible, subject to the same business rules and permissions, and backed up in the same content database. The image field control looks for image libraries named Images in the current and top-level site of the site collection, and these libraries are convenient places to store images. Policy and permissions can be set on these libraries to allow only selected people to add images and to require an approval before they are available. This can be very useful in enforcing site policy and avoiding the problem of authors who might upload unlicensed images for use on the site.

Part of the information architecture includes providing places to store images and other supporting content, such as documents, on the site. In general, it is a good idea to store assets at the root of the hierarchy in which they will be used. For example, general company images might be stored on the top-level site for use throughout all the child sites, and images relating to a particular product line might be stored on that product's site, which has all the related sites underneath it.

Reusable content is a MOSS feature for managing snippets of content, such as legal disclaimers and trademark declarations, which must appear on many or all pages. These snippets are kept in a special list, aptly named Reusable Content, in the top-level site. Authors can insert the reusable content on their pages, and rather than copy the content into the page, a reference to the content is stored and replaced with the snippet at runtime. This enables the reusable content to be changed in one place for every page in which it is used. Planning this up front is a good idea, before the site contains thousands of copy-and-paste snippets (each of which would have to be updated individually).

## Site Definitions

With a site navigation structure, wireframes for the page layouts, and a plan to store supporting content, turn to the containers that will hold the pages and other content. These are MOSS sites, and if the Publishing site definitions that come with MOSS provide everything you need, then no additional work is required here.

However, it may be desirable to allow users to create a site that has certain lists, content, or features already set up. This is most common in collaboration scenarios, but it sometimes is useful in publishing scenarios as well. For example, a product site might contain a specific set of pre-defined pages ready to fill with content, an image library for product pictures, and a special workflow associated with the page library.

To automate this, it's possible to create a new site definition, which tells MOSS how to create the new product sites. For planning purposes, determine what site definitions will be needed and what they need to do.

*For details on implementation, see Chapter 5, "Minimal Publishing Site Definition."*

Note that WSS has a similar concept called *site templates*, which can be created by simply clicking Save Site as Template on the Site Settings page. This link is not available for Publishing sites, and it would not be a good idea to use it if it were because site templates, like customized pages, live in the content database and cannot easily be staged and tested in a software release cycle. Instead, it is better to create a new site definition in Visual Studio and use your normal release cycle to test and deploy it.

## Roles and Permissions

Another planning activity is designing the security settings for your WCM solution. MOSS provides a number of authentication options; for details see Chapter 15, “Authentication and Authorization.” It is also possible to manage the content on one physical infrastructure and deploy it to a separate hosting environment; this is detailed in Chapter 17, “Content Deployment.” However, it is authorization that is of the greatest concern to the information architecture, as it is directly related to content.

MOSS provides just two levels of groups, *SharePoint groups* and *permission levels*, as shown in Figure 1-3.

*Just to be clear, in previous versions of SharePoint, groups were called “cross-site groups” and permission levels were called “site groups.”*

In addition, an ASP.NET role provider can add additional levels of grouping stored outside of SharePoint; out of the box, MOSS supports Active Directory (AD) groups in this manner, so with no code or extra effort, developers can use AD groups freely in MOSS. Here, the term “role provider group” is used to highlight the fact that it can be any external directory or other data source that has an ASP.NET role provider.

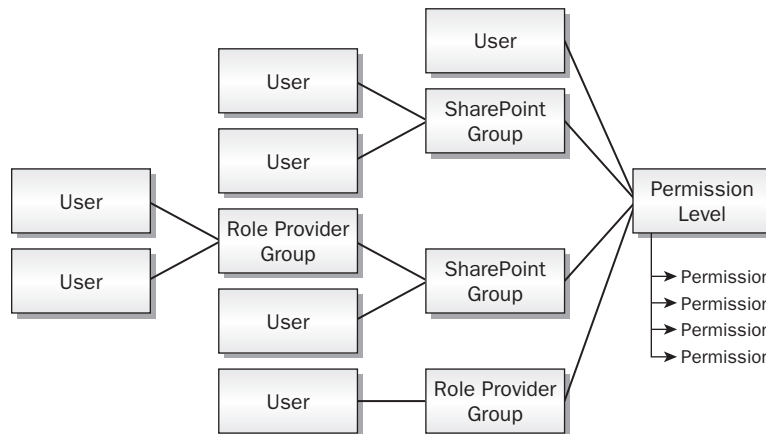


Figure 1-3

Permission levels are stored at the site level, though by default they are inherited from the parent site. Out of the box, a Publishing site has eight permission levels: Full Control, Design, Manage Hierarchy, Approve, Contribute, Read, Restricted Read, and Limited Access. Each of these has a set of fine-grained permissions associated with it, such as permission to view, add, edit, or delete items; the ability to add pages and child sites, and so on.

Permission levels grant their granular permissions on a site to a specific set of users, SharePoint groups and role provider groups. Within a site, the permission levels for these users and groups will apply to all the lists, libraries, folders, and items in the site, unless explicit permission levels are assigned to an object. For example, the user Joe may have Restricted Read access to the site, but Full Control over a particular folder of documents, and no access at all to a specific security Web page.

# Chapter 1: Embarking on Web Content Management Projects

---

It is possible to create custom permission levels, or to edit the granular permissions of the OOTB ones, except for Full Control and Limited Access. Full Control always allows complete access to the site, and Limited Access allows no direct access to site content at all, but is intended to allow users to traverse the site in order to access items within it that they have explicit permission to see. For example, a user might have access only to one page of a site, but still need access to style sheets and other supporting site infrastructure in order to view it; in that case, the user would need Limited Access permission on the site and Restricted Read access to the page.

The other groups within MOSS are SharePoint groups, which are more traditional groupings. SharePoint groups cannot be nested, but can contain external role provider groups. Note that SharePoint groups are stored at the site-collection level; this makes the same groups available in all sites in the site collection. This can be good or bad; it is convenient to have groups that apply across many sites, but sometimes it would be nice to have a second level of grouping (other than permission levels) within a site.

More details about permissions and groups can be found in the WSS documentation at [www.andrewconnell.com/go/200](http://www.andrewconnell.com/go/200).

The task then is to design the setup of permission levels, SharePoint groups, and role provider groups and how they will apply to the content in the site map. Think of the permission levels as roles — going back to the use cases, who are the actors, and what roles will they take on with respect to the various sites and content in the system? Those roles will end up reflecting the permission levels. The OOTB levels may be fine, but walk through them and see whether they make sense.

In general, it is better to grant permissions to SharePoint or role provider groups, rather than to individual users, as it is a lot easier to edit a group membership when you have organizational changes than it is to remember all the permissions that need to be set. The built-in SharePoint groups are also role oriented, but these are roles that extend throughout the site collection. This is a good approach, but a large site will probably need to have more explicit groups, such as Marketing Page Approvers or Product Site Authors, to easily manage different permissions in various parts of the site collection.

If you are using AD (or another role provider) and have the authority to create groups there, then groups can be nested as much as desired. This can be used, for example, to manage organizational groups — for example, if all the design engineers are already in a Design Engineers AD group, and these are the same people who can approve specification changes, then the existing Design Engineers AD group can be placed into a Specification Approvers SharePoint group.

A common pitfall is developing a solution in an environment where everybody has full control, only to later discover in system testing that it does not work in production, where more restrictive permissions are in place. Planning the permissions up front and making them part of unit testing is much less painful in the long run!

## ***User Profiles and Targeting***

If users are authenticated, then MOSS can do more than just set permissions for them. MOSS also contains a user profile system to store information about users, and a targeting system to automatically select relevant content based on the user's profile.

User profiles can be configured in the shared service provider (SSP) to have whatever properties are of interest. The usual approach is to populate the profile database by periodically importing data from the

directory service and to allow users to modify their profiles in their *My Sites*, although profiles can also be created and updated programmatically.

SharePoint *audiences* select users based on rules about their user profiles — for example, users whose country is equal to Australia, whose list of interests includes the word “boating” or who are members of the Top Partners group in AD. Content, including Web pages, can be targeted to specific audiences by placing a list of audiences in the Target Audiences field. The Content Query Web Part can be set to respect these audience settings, so users are presented only with content that matches the query *and* the user’s audiences. Note that audience targeting is different from security; users can still get to content via navigation, search, or an explicit URL, even if they are not in the target audience. Rather than security, targeting is intended to highlight content of interest to users.

## Search Strategy

Site users expect to be able to both search for content and navigate to it, and fortunately MOSS has a great built-in search engine to accommodate their needs. MOSS search is a huge topic that is beyond the scope of this book, but it is worth pointing out some of the possibilities.

In addition to indexing MOSS content, the MOSS search engine can index external file shares, Microsoft Exchange public folders, and Lotus Notes databases. It can also crawl any Web site, which can be useful when there are related sites that are not in MOSS.

Here are some strategies for optimizing MOSS search in a WCM solution:

- ❑ **Branding** — Naturally, the search pages inherit the master page and Cascading Style Sheets (CSS) along with the rest of the site. Further customizations are easy, but the search user interface is comprised of Web Parts that can be reconfigured and rearranged to meet your needs.
- ❑ **Search scopes** — A *search scope* is basically a set of partial queries called *scope rules* that narrow down a user’s search. Users normally see scopes in a drop-down list next to the search box, or as tabs on the results page. For example, a News search might show only news stories matching the user’s query, and the scope rule behind it could select the news based on Web address, content source, or other properties.

When planning a search scope that is not based on the location of the content, the scope needs to be based on a property of some kind. One of the built-in properties is `contentclass`, which contains the name of the content type; therefore, if there is a content type for press releases, then making a Press Releases search scope would be easy.

In other cases, it may be necessary to include a property in the page content type, visible only at editing time, that puts pages into the scope. For example, if a random cross-section of pages contains medical information, and Medical search scope is planned, placing the word Medical in a hidden field would enable these pages to be selected in a search scope.

Search scopes can be set up in the SSP administration site.

- ❑ **Authoritative pages** — Search relevancy can be adjusted by specifying authoritative pages in the search catalog. Pages can be designated as most, second, or third most authoritative, and their relevancy is adjusted upwards; pages can also be designated as non-authoritative and their relevancy is demoted. This can be useful when areas of a site contain information that needs to

## Chapter 1: Embarking on Web Content Management Projects

---

be highlighted (authored information, for example) or pushed to the bottom (such as old or less popular content). Note that an authoritative page also affects the pages to which it links: For example, a product summary page that links to all products could be marked as authoritative to raise the relevancy of all the product pages, as well as the summary. Authoritative pages can be set up in the SSP administration site.

- ❑ **Keywords/best bets** — Pages can be designated as *best bets* and associated with keywords. When the keyword is included in a search query, any related best bets will show up in their own section of the search results page. Keywords can be set up as best bets in the top-level site settings of each site collection.

A best practice is to monitor the Search Usage reports (in the SSP administration sites) and look at the Search Results report called Queries with Zero Results. Guess what? There is a frustrated user behind every one of those queries! These are candidates for keywords and best bets.

Traditional Web sites place metadata in META tags, but this isn't necessary for the MOSS search engine to pick up the metadata when crawling MOSS content; it can query the metadata directly. However, it may be desirable to include META tags in order to allow external search engines to crawl a WCM solution. This can be accomplished by incorporating a custom control or Web Part in the master page, which emits the desired META tags based on the page content (some of which may be hidden from site visitors by placing it within an Edit Mode Panel so they only appear when the page is being edited). An example of such a control, called *MetaTagsGenerator*, is available on CodePlex at [www.andrewconnell.com/go/201](http://www.andrewconnell.com/go/201).

## Summary

WCM removes bottlenecks in the process of Web publishing by enabling business users to directly author and edit content independently of one another and of site designers. Similarly, site designers can work independently of the content creators, and can update existing content with new branding and other visual changes.

WCM can also improve the quality and consistency of a Web site by providing a structure, but only if the solution has the structure implemented as part of the design. Here is a checklist of planning activities:

- ❑ Use case scenarios defining both internal and external actors and how they will use the site
- ❑ Overall site map showing the site structure, including the placement of site collections if there is more than one
- ❑ Image, and eventually HTML mock-ups, of the visual design
- ❑ Wireframe representations of desired page layouts, mapped to documented content types that will define the fields used to store the data
- ❑ A list of required custom features for development based on the wireframe representations and visual design
- ❑ Defined locations for supporting content such as images and attached documents
- ❑ A structure for SharePoint groups and permission levels, perhaps color coded to sites on the site map



## Chapter 1: Embarking on Web Content Management Projects

---

Remember that your Web site will have information architecture, just as every city has a street map. Whether yours will be easy or confusing depends on the planning and thought that goes into the solution. Keep in mind that the real point of Web publishing is not the cool technology, but the content itself. Organizing the content in the most helpful way possible will do more for a site's success than any other factor, and a well designed WCM site will make that easy.

The following two resources are useful for designing usable Web sites:

- ❑ Improving Web Site Usability and Appeal: [www.andrewconnell.com/go/202](http://www.andrewconnell.com/go/202)
- ❑ Step-by-Step Usability Guide: [www.andrewconnell.com/go/203](http://www.andrewconnell.com/go/203)

