Content Management Systems

18

In this chapter you will learn about . . .

- The challenges of managing a website
- Content management systems principles and practices

One of the most significant changes in the web development world has been the widespread adoption of content management systems (CMSs) as a mechanism for creating and managing websites. CMSs provide easy-to-use tools to publish
and edit content, while managing the structure, layout, and administration of the site through simple but powerful administrative interfaces.

18.1 Managing Websites

Throughout this textbook you have seen the core technologies that support a rich and interactive web. You can create attractive web pages with HTML and CSS, make them interactive with client-side scripts, and process dynamic requests with PHP and databases. The most significant drawback to the sites you have created so far in this book is that these sites require a software developer to edit the code in order to make changes in the future.

For a small company, this can be a significant problem, since they may want to update the website weekly or daily and cannot afford a full-time programmer on staff. In such an environment, the person managing the website likely performs other, nondevelopment duties. Depending on the size of a company the person could be anyone from a receptionist all the way up to the CEO.

These companies want a system that is

- Easy for a nontechnical person to make changes to
- Consistent and professional looking across the site
- Cost effective

Content management systems, once installed, can indeed be easy, consistent, professional, and cost effective. However, they still have technical underpinnings that need to be understood by the people installing and supporting them.

18.1.1 Components of a Managed Website

Beyond the requirements for the business owner, a typical website will eventually need to implement the following categories of functionality:

- **Management** provides a mechanism for uploading and managing images, documents, videos, and other assets.
- **Menu control** manages the menus on a site and links menu items to particular pages.
- **Search functionality** can be built into systems so that users can search the entire website.
- **Template management** allows the structure of the site to be edited and then applied to all pages.
- **User management** permits multiple authors to work simultaneously and attribute changes to the appropriate individual. It can also restrict permissions.
- **Version control** tracks the changes in the site over time.
- **Workflow** defines the process of approval for publishing content.
- **WYSIWYG editor** allows nontechnical users to create and edit HTML content and CSS styles without manipulating code.

FIGURE 18.1 The challenge of managing a WWW site without hosting considerations

Even for a sophisticated web developer, the challenge of implementing all this functionality can be daunting as illustrated in Figure 18.1. Systems that can manage all of the pieces reduce the complexity for the site manager and simplify the management of a site, replacing the web of independent pieces with a single web-based CMS as illustrated in Figure 18.2.

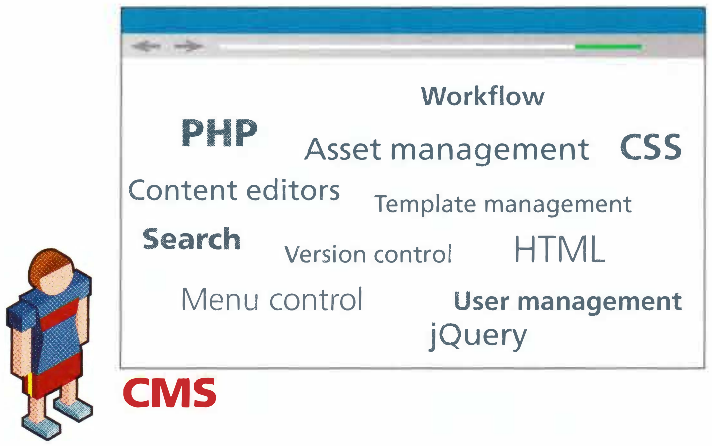


FIGURE 18.2 The benefit of a Web Content Management System

You might consider using a CMS yourself even though you could address all the issues, since a CMS manages many of these pieces for you in the majority of situations, leaving you more time for other things.

18.2 Content Management Systems

HANDS-ON EXERCISES

LAB 18 EXERCISE
Set Up WordPress

Content management system (CMS) is the name given to the category of software that easily manages websites with support for multiple users. In this book we focus on web-based content management systems (WCMS), which go beyond user and document management to implement core website management principles. We will relax the formal definitions so that when we say CMS we are referring to a web-based CMS.

PRO TIP

Document management systems (DMSs) are a class of software designed to replace paper documents in an office setting and date back to the 1970s. These systems typically implement many features users care about for documents including: file storage, multiuser workflows, versioning, searching, user management, publication, and others.

The principles from these systems are also the same in the web content management systems. Benefiting from a well-defined and mature class of software like DMS in the web context means you can avoid mistakes already made, and benefit from their solutions.

It also means that many companies already have a document management solution deployed enterprise wide. These enterprise software systems often have a web component that can be purchased to leverage the investment already made in the system. Tools like SharePoint are popular when companies have already adopted Microsoft services like Active Directory and Windows-based IIS web servers in their organization. Similarly, a company running SAP may opt to use their web application server rather than another commercial or open-source system.

With a CMS end users can focus on publishing content and know that the system will put that content in the right place using the right technologies. Once properly configured and installed, a CMS requires only minimal maintenance to stay operational, can reduce costs, and often doesn't need a full-time web developer to make changes.

18.2.1 Types of CMS

A simple search for the term "CMS" in a search engine will demonstrate that there are a lot of content management systems available. Indeed, a Wikipedia page listing available web CMSs has, at the time of writing, 109 open-source systems and 39 proprietary systems.[1] These systems are implemented using a wide range of development technologies including PHP, ASP.NET, Java, Ruby, Python, and others. Some of these systems are free, while others can cost hundreds of thousands of dollars.

This chapter uses WordPress as its sample CMS. Originally a blogging engine, more and more CMS functionality has been added to it, and now, due in part to its popularity as a way to manage blogs, WordPress is by far the most popular CMS, as shown in Figure 18.3. As a result, the ability to customize and adapt WordPress has become an important skill for many web developers. As you will see throughout this chapter, it implements all the key pieces of a complete web management system, and goes beyond that, allowing you to leverage the work of thousands of developers and designers in the form of *plugins* and *themes* (written in PHP).

Before moving on to the specifics of WordPress, you will notice from Figure 18.3 that other content systems enjoy substantial support in industry. As well, remember that the technology used in intranet sites (i.e., sites within a company) is typically hidden from analytic sites like builtwith.com. Private corporate intranet portals are one of the most common uses of CMSs so the market share of systems like SharePoint and IBM's suite may in reality be substantially larger than shown in Figure 18.3.

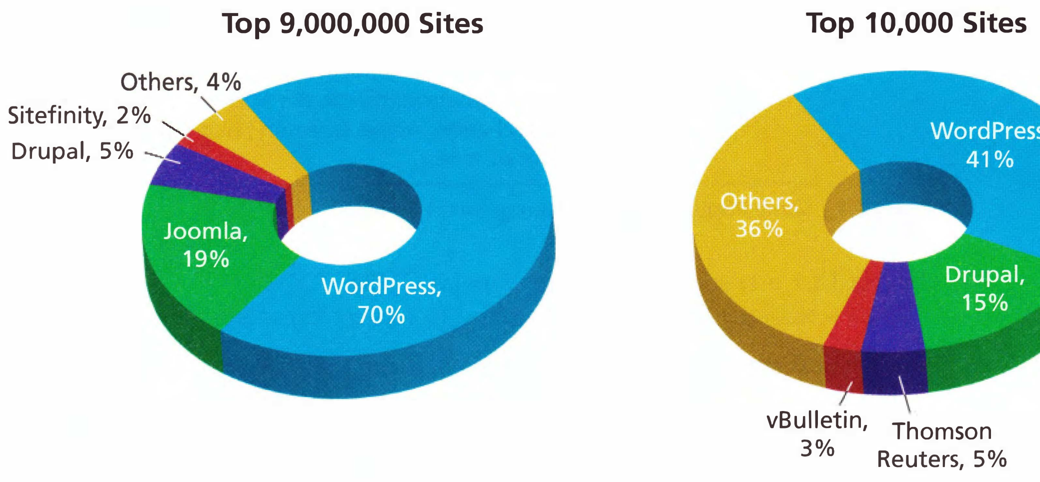


FIGURE 18.3 Market share of content management systems (data courtesy of BuiltWith.com)

DotNetNuke	Written in C#, this CMS has both open-source and commercial versions. Its use of the popular .NET Framework from Microsoft makes it a popular open-source alternative to PHP-based CMS.
Drupal	Written in PHP, Drupal is a popular CMS with enterprise-level workflow functionality. It is a popular CMS used in many large organizations including **whitehouse.gov** and **data.gov.uk**.
ExpressionsEngine	A proprietary CMS written in PHP with a "core" version available for free for nonprofit and personal use. ExpressionsEngine uses its own template syntax to make customization easier for nondevelopers.
IBM Enterprise Content Management (ECM)	This proprietary system (written in Java) requires the use of several additional proprietary components. It is popular in companies that have already licensed software from IBM and require mature enterprise CMS with advanced auditing, and workflow capabilities that integrate with other enterprise systems from IBM.
Joomla!	Written in PHP, Joomla! Is one of the older free and open-source CMS (started in 2005). With many plugins and extensions available, it continues to be a popular CMS.
Moodle	Written in PHP, Moodle is an open-source learning management system with over 7.5 million courses using it.[2] The functionality is focused on assignment submissions, discussion forums, and grade/enrollment management although it implements most core CMS principles as well.
SharePoint	SharePoint is an enterprise-focused, proprietary CMS from Microsoft that is especially popular in corporate intranet sites. It is tightly integrated with the Microsoft suite of tools (like Office, Exchange, Active Directory) and has a mature and broad set of tools.

TABLE 18.1 Some Popular Content Management Systems

Table 18.1 lists some of the more popular CMSs.

When selecting a CMS there are several factors to consider including:

- **Technical requirements:** Each CMS has particular requirements in terms of the functionality it offers as well as the server software needed and the database it is compatible with. Your client may have additional requirements to consider.
- **System support:** Some systems have larger and more supportive communities/companies than others. Since you are going to rely on the CMS to patch

bugs and add new features, it's important that the CMS community be active in supporting these types of updates or you will be at risk of attack.

- **Ease of use:** Probably the most important consideration is that the system itself must be easy to use by nontechnical staff.

NOTE

WordPress is designed to be easy to use. If you have a running server, you should really stop reading this section and install WordPress right now! Reading this section while you play around in your own installation's dashboard will help reinforce how WordPress implements the key aspects of a CMS in an experiential way. Later, when we go into the customization of WordPress, we assume you have completed the lab exercises and have some experience.

18.3 CMS Components

As mentioned at the beginning of the chapter, a managed website typically requires a range of components such as asset management, templating, user management, and so on. A CMS provides implementations of these components within a single piece of software.

It should be reiterated that these web content management systems are themselves web applications. As such, they provide a series of web pages that you can use to add/edit content, manage users, upload media, etc. Most content systems use some type of dashboard as an easy-to-use front end to all the major functionality of the system.

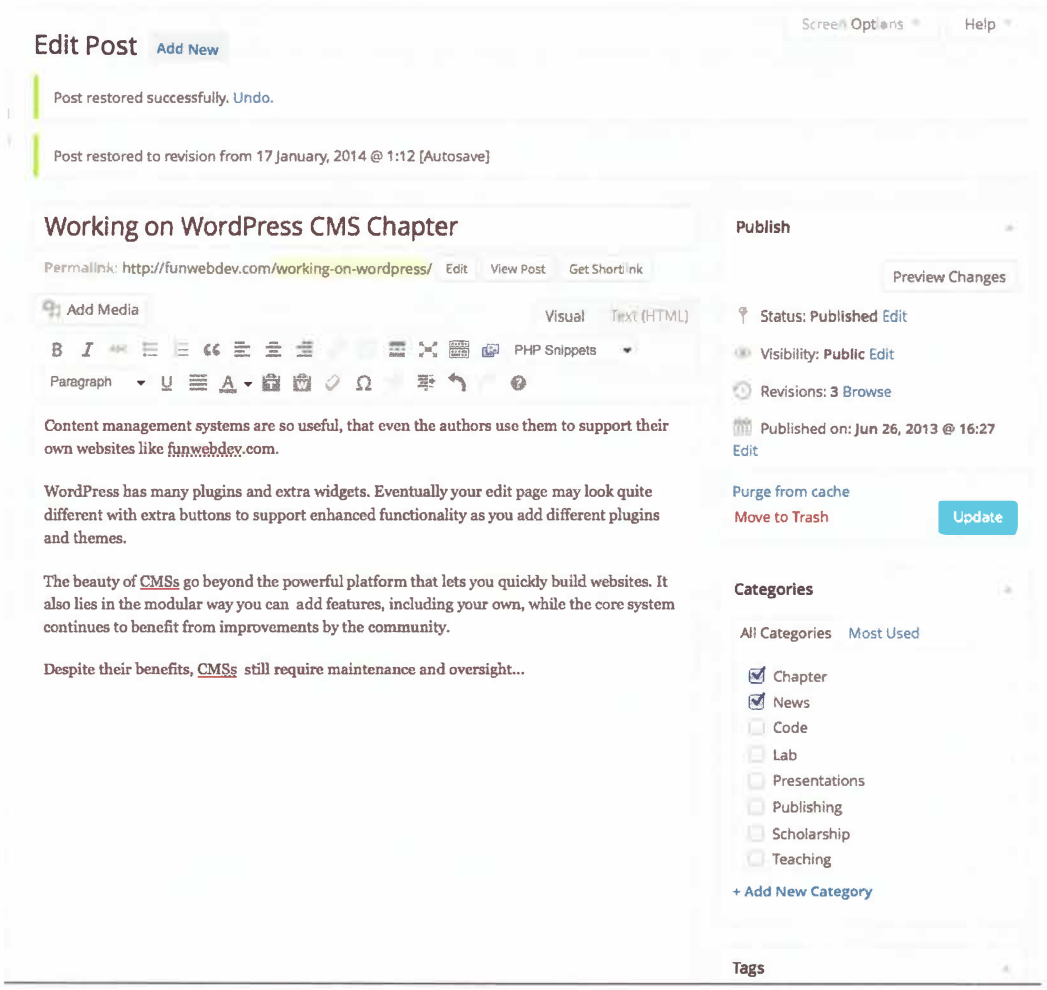
In WordPress the dashboard is accessible by going to **/wp-admin/** off the root of your installation in a web browser. You will have to log in with a username and password, as specified during the installation process (more on that later). Most users find that the dashboard can be navigated without reading too much documentation, since the links are well named and the interface is intuitive.

18.3.1 Post and Page Management

Blogging environments such as WordPress use posts as one important way of adding content to the site. Posts are usually displayed in reverse chronological order (i.e., most recent first) and are typically assigned to categories or tagged with keywords as a way of organizing them. Many sites allow users to comment on posts as well. Figure 18.4 illustrates the post-editing page in WordPress. Notice the easy-to-use category and tag interfaces on the right side of the editor.

HANDS-ON EXERCISES

LAB 18 EXERCISE
Create Pages



Post restored successfully. Undo.

Post restored to revision from 17 January, 2014 @ 1:12 [Autosave]

Working on WordPress CMS Chapter

Permalink: http://funwebdev.com/working-on-wordpress/ Edit View Post Get Shortlink

Content management systems are so useful, that even the authors use them to support their own websites like funwebdev.com.

WordPress has many plugins and extra widgets. Eventually your edit page may look quite different with extra buttons to support enhanced functionality as you add different plugins and themes.

The beauty of CMSs go beyond the powerful platform that lets you quickly build websites. It also lies in the modular way you can add features, including your own, while the core system continues to benefit from improvements by the community.

Despite their benefits, CMSs still require maintenance and oversight...

Status: Published Edit
Visibility: Public Edit
Revisions: 3 Browse
Published on: Jun 26, 2013 @ 16:27 Edit

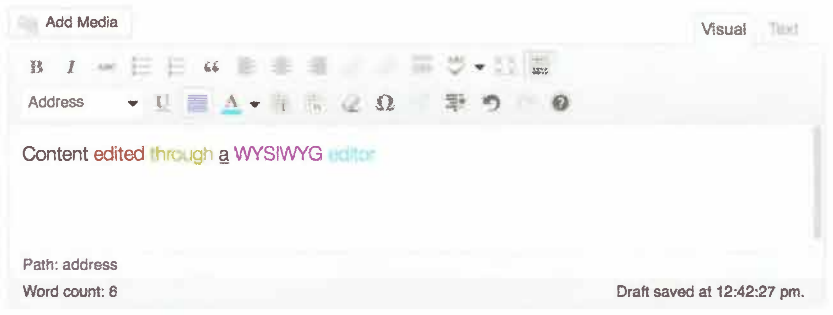
[✓] Chapter
[✓] News
[ ] Code
[ ] Lab
[ ] Presentations
[ ] Publishing
[ ] Scholarship
[ ] Teaching

FIGURE 18.4 Screenshot of the post editor in WordPress

CMSs typically use pages as the main organizational unit. **Pages** contains content and typically do not display the date, categories, and tags that posts use. The main menu hierarchy of a CMS site will typically be constructed from pages.

WordPress supports both posts and pages; you typically use pages for substantial content that needs to be readily available, while posts are used for smaller chunks of content that are associated with a timestamp, categories, and tags.

Most CMSs impose some type of restrictions on page and post management. Some users may only be able to edit existing pages; others may be allowed to create posts but not pages. More complex CMSs impose a workflow where edits from users need to be approved by other users before they are published. Larger



Content edited through a WYSIWYG editor

FIGURE 18.5 Screenshot of the TinyMCE WYSIWYG editor included with WordPress

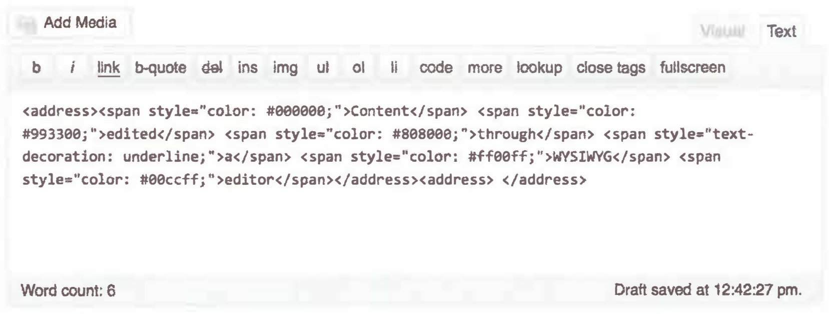
organizations often require this type of workflow management to ensure consistency of content or to provide editorial or legal control over content.

18.3.2 WYSIWYG Editors

What You See Is What You Get (WYSIWYG) design is a user interface design pattern where you present the users with an exact (or close) view of what the final product will look like, rather than a coded representation. These tools generate HTML and CSS automatically through intuitive user interfaces such as the one shown in Figure 18.5.

The advantage of these tools is that nontechnical users are not required to know HTML and CSS, thus permitting them to edit and create pages with a focus on the content, rather than the medium it will be encoded into (HTML). These tools normally also allow the user to edit the underlying HTML as shown in Figure 18.6.

WYSIWYG editors often contain useful tools like validators, spell checkers, and link builders. A good CMS will also allow a super-user like you to define CSS styles, which are then available through the editor in a dropdown list as illustrated in Figure 18.7. This control allows content creators to choose from predefined styles, rather than define them every time. It maintains consistency from page to page, and yet still allows them to create new styles if need be.



```
<address><span style="color: #000000;">Content</span> <span style="color:
#993300;">edited</span> <span style="color: #808000;">through</span> <span style="text-
decoration: underline;">a</span> <span style="color: #ff00ff;">WYSIWYG</span> <span
style="color: #00ccff;">editor</span></address><address> </address>
```

FIGURE 18.6 The HTML view of a WYSIWYG editor

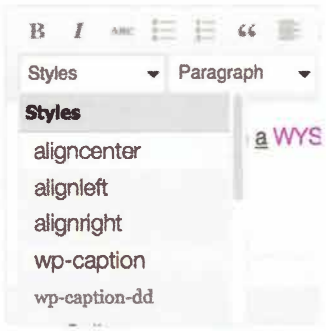


FIGURE 18.7 TinyMCE with a style dropdown box using the styles from a predefined CSS stylesheet

18.3.3 Template Management

Template management refers to the systems that manage the structure of a website, independently of the content of each particular page, and is one of the most important parts of any CMS. The concept of a template is an old one and is used in disciplines outside web development. Newspapers, magazines, and even cake decorators have adopted the design principle of having a handful of layouts, and then inserting content into the templates as needed.

When you sketch a wireframe design (i.e., a rough preliminary design) of a website, you might think of the wires as the template, with everything else being the content. Several pages can use the same wireframe, but with distinct content as shown in Figure 18.8. While the content is often managed by mapping URLs to pages in a database, conceptually the content can come from anywhere.

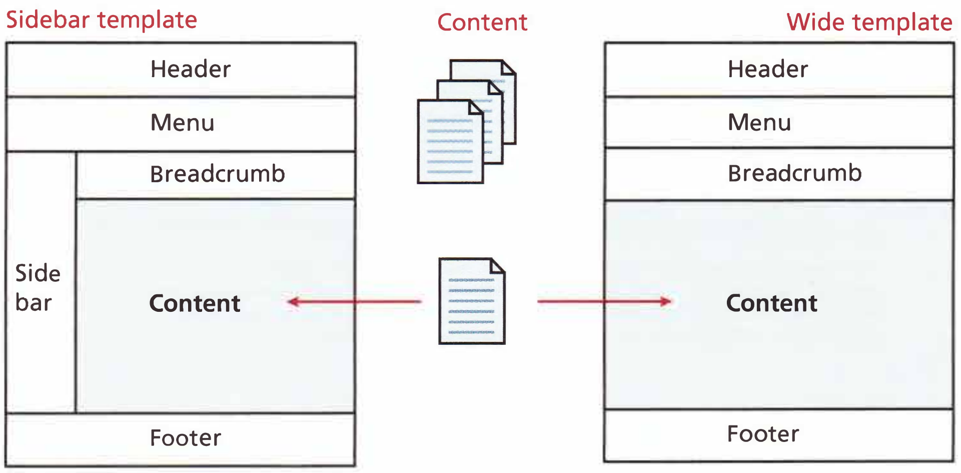


FIGURE 18.8 Multiple templates and their relationship to content

One of the trickiest aspects of creating a dynamic website is implementing the menu and sidebars, since not only are they very dynamic, but they need to be consistent as well. Templates allow you to manage multiple wireframes all using the same content and then change them on a per-page, or site-wide basis as needed. You could, for example, maintain multiple holiday templates and apply the appropriate one for each holiday season. Or, you might want one template with a sidebar full of extra links, and another template for big content as in Figure 18.8.

18.3.4 Menu Control

The term menu refers to the hierarchical structure of the content of a site as well as the user interface reflection of that hierarchy (typically a prominent list of links). The user interacts with the menu frequently, and they can range in style and feel from pop-up menus to static lists. A menu is often managed alongside templates since the template must integrate the menu for display purposes.

HANDS-ON EXERCISES

LAB 18 EXERCISE
Navigation from Pages

Some key pieces of functionality that should be supported in the menu control capability of a CMS include:

- Rearrange menu items and their hierarchy.
- Change the destination page or URL for any menu item.
- Add, edit, or remove menu items.
- Change the style and look/feel of the menu in one place.
- Manage short URLs associated with each menu item.

In WordPress menus are typically managed by creating pages, which are associated with menu items in a traditional hierarchy. By controlling the structure and ordering of pages, you can define your desired hierarchies. In addition there is a menu management interface in the WordPress dashboard that allows more granular management of multiple menu lists.

18.3.5 User Management and Roles

User management refers to a system's ability to have many users all working together on the same website simultaneously. While some corporate content management systems tie into existing user management products like Active Directory or LDAP, a stand-alone CMS must include the facility to manage users as well.

A CMS that includes user management must provide easy-to-use interfaces for a nontechnical person to manage users. These functions include:

- Adding a new user
- Resetting a user password
- Allowing users to recover their own passwords

- Allowing users to manage their own profiles, including name, avatars, and email addresses
- Tracking logins

In a modern CMS the ability to assign roles to users is also essential since you may not want all your users to be able to perform the above functions. Typically, user management is delegated to one of the senior roles like site manager or super administrator.

18.3.6 User Roles

Users in a CMS are given a user role, which specifies which rights and privileges that user has. Roles in WordPress are analogous to roles in the publishing industry where the jobs of a journalist, editor, and photographer are distinct.

A typical CMS allows users to be assigned one of the four roles as illustrated in Figure 18.9: content creator, content publisher, site manager, and super administrator. Although more finely grained controls are normally used in practice, the essential theory behind roles can be illustrated using just these four.

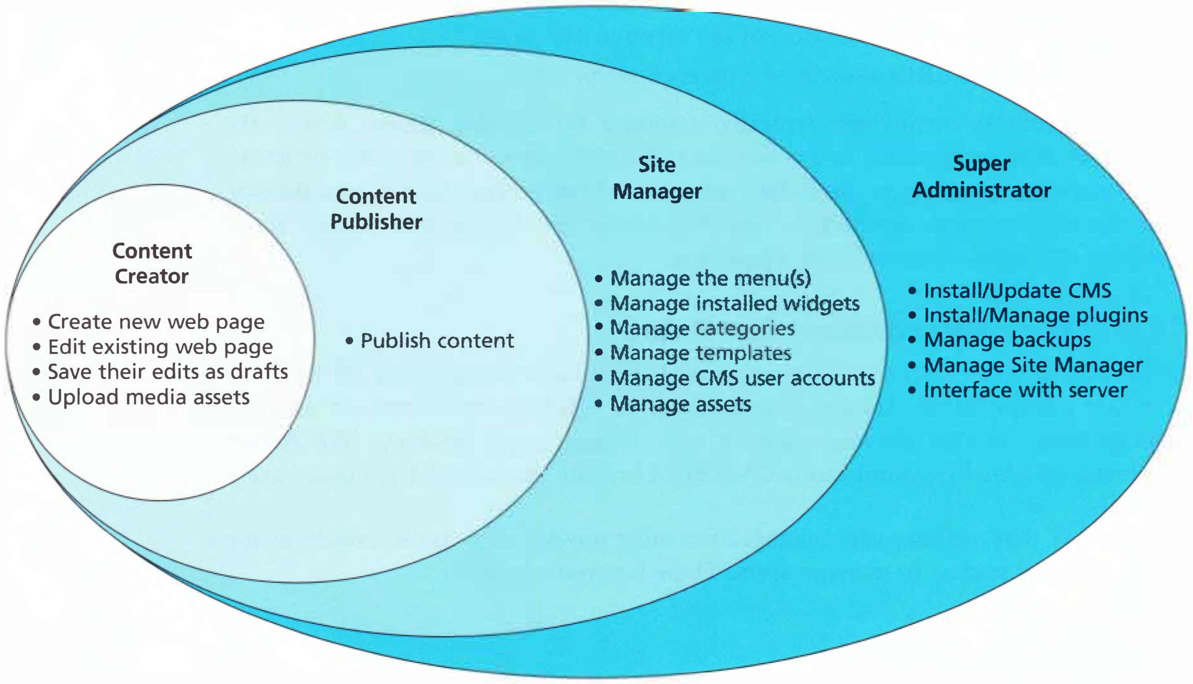


FIGURE 18.9 Typical roles and responsibilities in a web CMS

Content Creator

Content creators do exactly what their title implies: they create new pieces of content for the website. This role is often the one that requires sub-roles because there are many types of content that they can contribute. These users are able to:

- Create new web pages
- Edit existing web pages
- Save their edits in a draft form
- Upload media assets such as images and videos

None of this role's activities result in any change whatsoever to the live website. Instead the *draft* submissions of new or edited pages are subject to oversight by the next role, the publisher.

Content Publisher

Content publishers are gatekeepers who determine if a submitted piece of content should be published. This category exists because entities like corporations or universities need to vet their public messages before they go live. The major piece of functionality for these users is the ability to publish pages to the live website. Since they can also perform all the duties of a content creator, they can also make edits and create new pages themselves, but unlike a creator, they can publish immediately.

The relationship between the publisher and creator is a complex one, but the whole concept of workflow (covered in the next section) relies on the existence of these roles.

Site Manager

The site manager is the role for users who can not only perform all the creation and publishing tasks of the roles beneath them, but can also control more complicated aspects of the site including:

- Menu management
- Management of installed plugins and widgets
- Category and template management
- CMS user account management
- Asset management

Although this user does not have unlimited access to the CMS installation, they are able to manage most of the day-to-day activity in the site. These types of users are typically more comfortable with computational thinking, although they can still be nonprogrammers. Since they can control the menu and templates, these users can also significantly impact the site, including possibly breaking some functionality.

Super Administrator

The super administrator role is normally reserved for a technical person, often the web developer who originally configured and installed the CMS. These users are able to access all of the functionality within the CMS and normally have access to the underlying server it is hosted on as well. In addition to all of the functionality of the other types of user, the super administrator is often charged with:

- Managing the backup strategy for the site
- Creating/deleting CMS site manager accounts
- Keeping the CMS up to date
- Managing plugin and template installation

Ideally, the super administrator will rarely be involved in the normal day-to-day operation of the CMS. Although in theory you can make every user a super administrator, doing so is extremely unwise since this would significantly increase the chance that a user will make a destructive change to the site (this is an application of the *principle of least privilege* from Chapter 16, Section 16.2.5).

WordPress Roles

In WordPress the default roles are Administrator, Author, Editor, Contributor, and Subscriber, which are very similar to our generic roles with the Administrator being our super administrator and the Subscriber being a new type of role that is read-only. One manifestation of roles is how they change the dashboard for each class of user as illustrated in Figure 18.10. The diagram does not show some of the additional details, like the ability to publish versus save as draft, but it gives an overall sense of the capabilities.

18.3.7 Workflow and Version Control

HANDS-ON EXERCISES

LAB 18 EXERCISE
Create WordPress Users

Workflow refers to the process of approval for publishing content. It is best understood by considering the way that journalists and editors work together at a newspaper. Using roles as described above, you can see that the content created by content creators must eventually be approved or published by a higher-ranking user. While many journalists can be submitting stories, it is the editor who decides what gets published and where. In this structure another class of contributor, photographers, may be able to upload pictures, but editors (or journalists) choose where they will be published.

CMSs integrate the notion of workflow by generalizing the concept and allowing for every user in the system to have roles. Each role is then granted permission to do various things including publishing a post, saving a draft, uploading an image, and changing the home page.

Figure 18.11 illustrates a sample workflow to get a single news story published in a newspaper or magazine office. The first draft of the story is edited, creating new

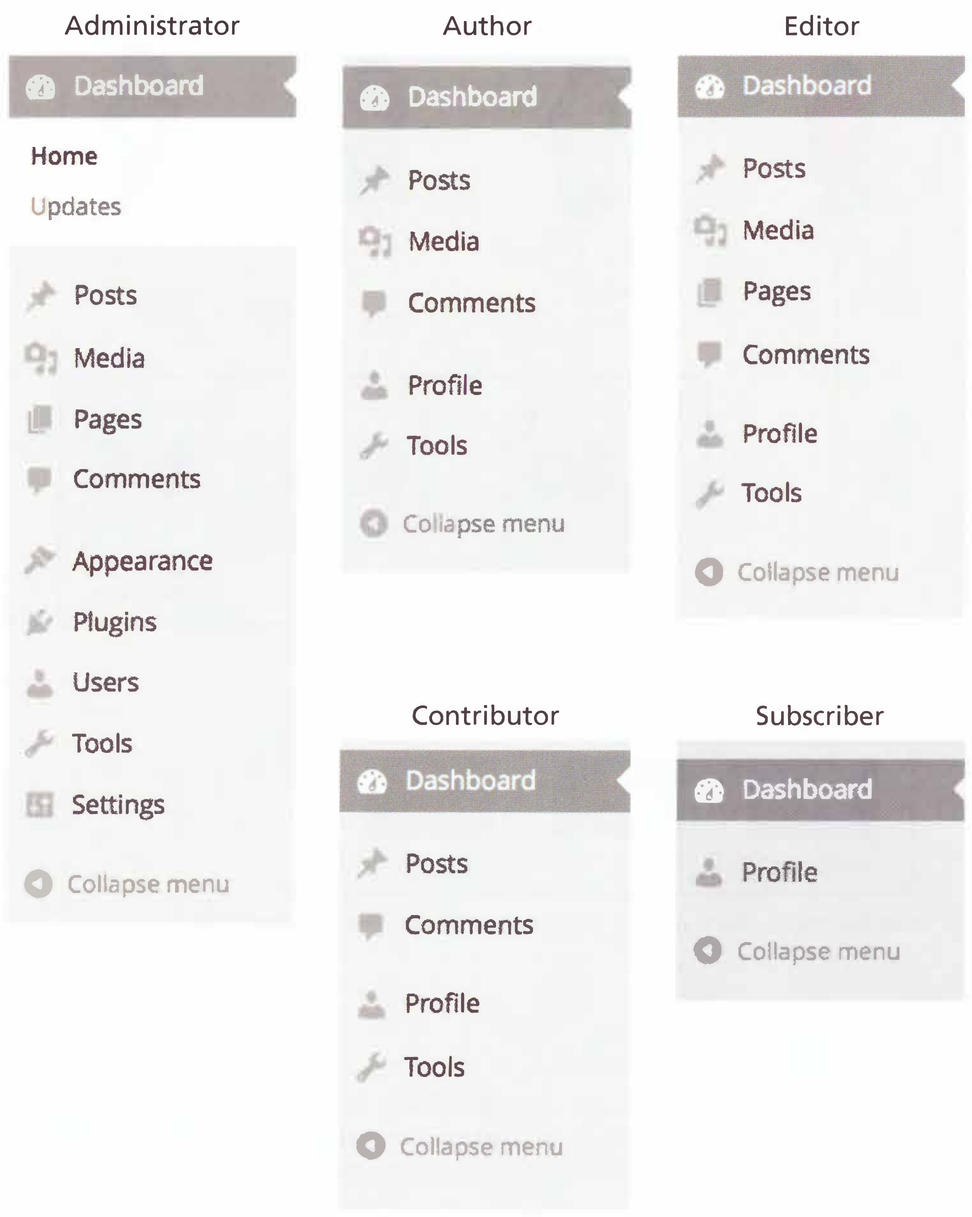


FIGURE 18.10 Multiple dashboard menus for the five default roles in WordPress

versions, until finally the publisher approves the story for print. *Notice that the super administrator plays no role in this workflow; while that user is all-powerful, he or she is seldom needed in the regular course of business.*

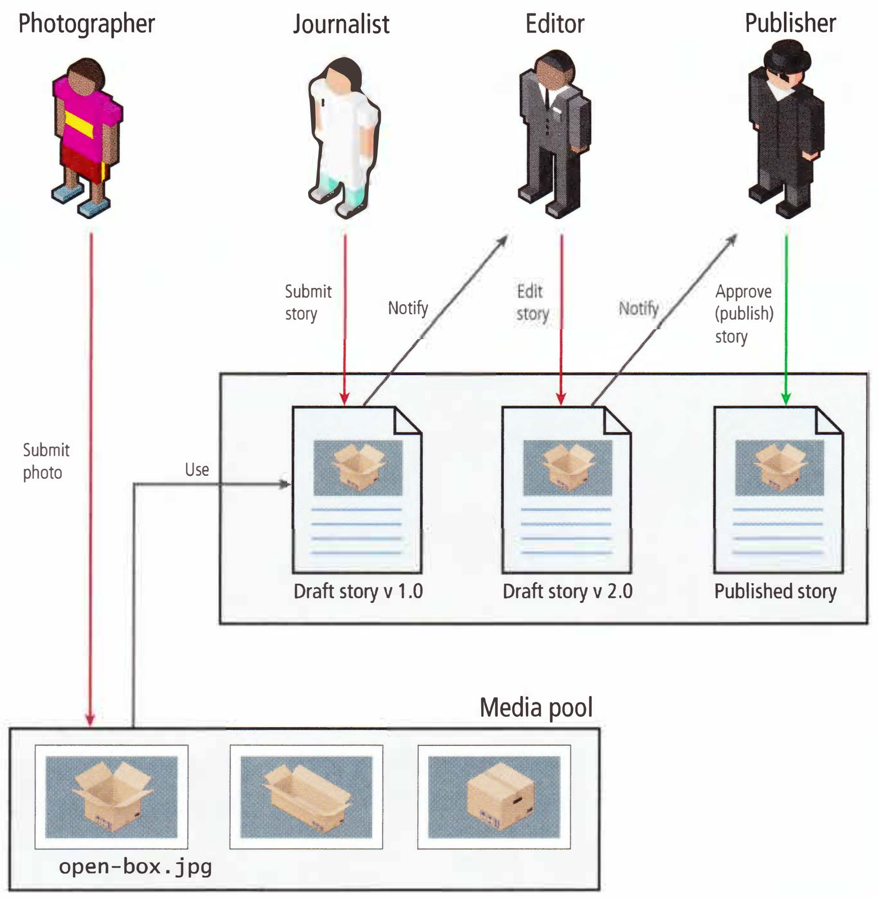
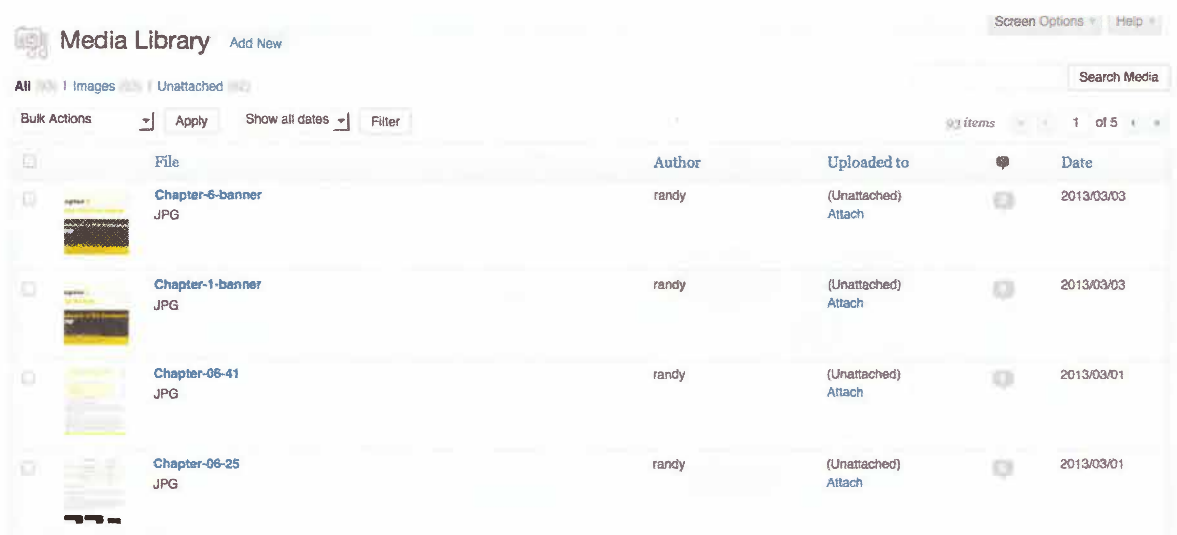


FIGURE 18.11 Illustration of multiple people working in a workflow

18.3.8 Asset Management

Websites can include a wide array of media. There are HTML documents, but also images, videos, and sound files, as well other document types or plugins. The basic functionality of digital asset management software enables the user to:

- Import new assets
- Edit the metadata associated with assets
- Delete assets
- Browse assets for inclusion in content
- Perform searches or apply filters to find assets


Media Library Add New

Screen Options Help

All | Images | Unattached

Search Media

Bulk Actions Apply Show all dates Filter

93 items 1 of 5

File	Author	Uploaded to	Date
Chapter-6-banner JPG	randy	(Unattached) Attach	2013/03/03
Chapter-1-banner JPG	randy	(Unattached) Attach	2013/03/03
Chapter-06-41 JPG	randy	(Unattached) Attach	2013/03/01
Chapter-06-25 JPG	randy	(Unattached) Attach	2013/03/01

FIGURE 18.12 Media management portal in WordPress

In a web context there are two categories of asset. The first are the pages of a website, which are integrated into the navigation and structure of the site. The second are the non-HTML assets of a site, which can be linked to from pages, or embedded as images or plugins. Although some asset management systems manage both in the same way, the management of non-HTML assets requires different capabilities than pages.

In WordPress, media management is done through a media management portal and through the media widgets built into the page's WYSIWYG editor. This allows you to manage the media in one location as shown in Figure 18.12 but also lets content creators search for media right from the place they edit their web pages as shown in Figure 18.13.

The media management portal allows the manager of the site to categorize and tag assets for easier search and retrieval. It also allows the management of where the files are uploaded and how they are stored.

18.3.9 Search

Searching has become a core way that users navigate the web, not only through search engines, but also through the built-in search boxes on websites.

Unfortunately, creating a fast and correct search of all your content is not straightforward. Ironically, as the size of your site increases, so too does the need for search functionality, and the complexity of such functionality. There are three strategies to do website search: SQL queries using LIKE, third-party search engines, and search indexes.

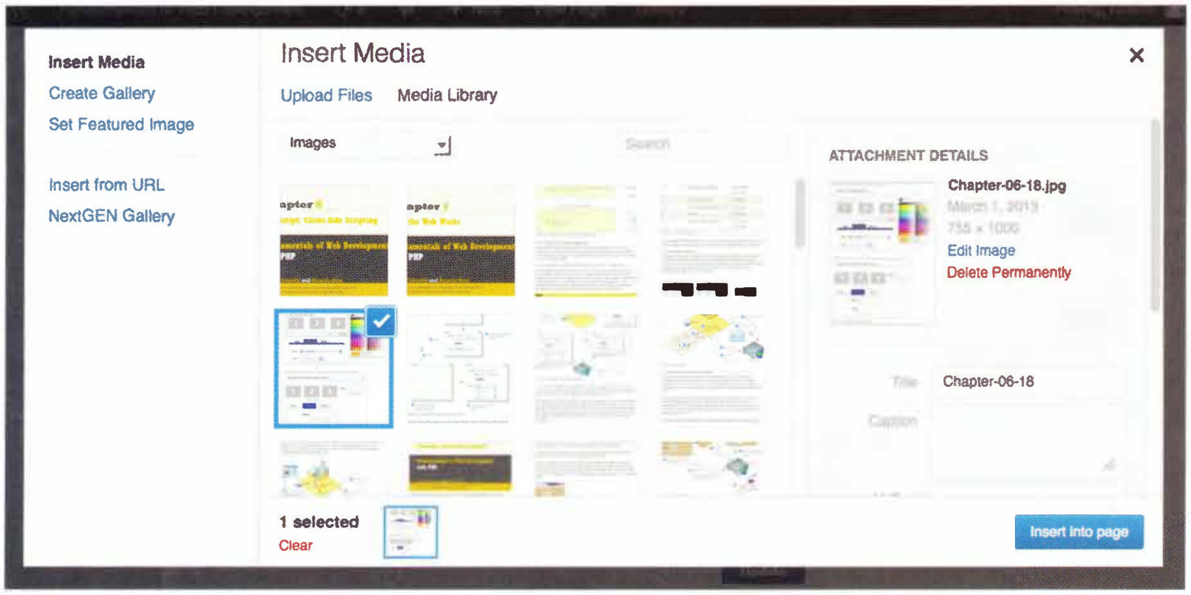


FIGURE 18.13 Screenshot of a media insertion dialog in the page editor

Although you could search for a word in every page of content using the MySQL LIKE with `%` wildcards, that technique cannot make use of database indexes, and thus suffers from poor performance. A poorly performing search is computationally expensive, and results in poor user satisfaction. Included by default with WordPress, it's worth seeking a replacement.

HANDS-ON EXERCISES

LAB 18 EXERCISE
Install a Plugin

To address this poor performance, many websites offload search to a third-party search engine. Using Google, for example, one can search our site easily by typing `site:funwebdev.com SearchTerm` into the search field.

The problem with using a third party is that you are subject to their usage policies and restrictions. Also, you are encouraging users to leave your site to search, which is never good, since there is a chance they won't return. Also, you are relying on the third party having updated their cache with your newest posts, something you cannot be sure of at all times.

Doing things properly requires that the system build and manage its own index of search terms based on the content, so that the words on each page are indexed and cross referenced, and thus quickly searchable. This is a trade off where the preprocessing (which is intensive) happens at a scheduled time once, and then on-the-fly search results can use the produced index, resulting in faster search speeds.

While you could build a search index yourself, plugins normally exist such as WPSearch, which already implements this strategy so that you can easily build an index to get faster user searches.[3]

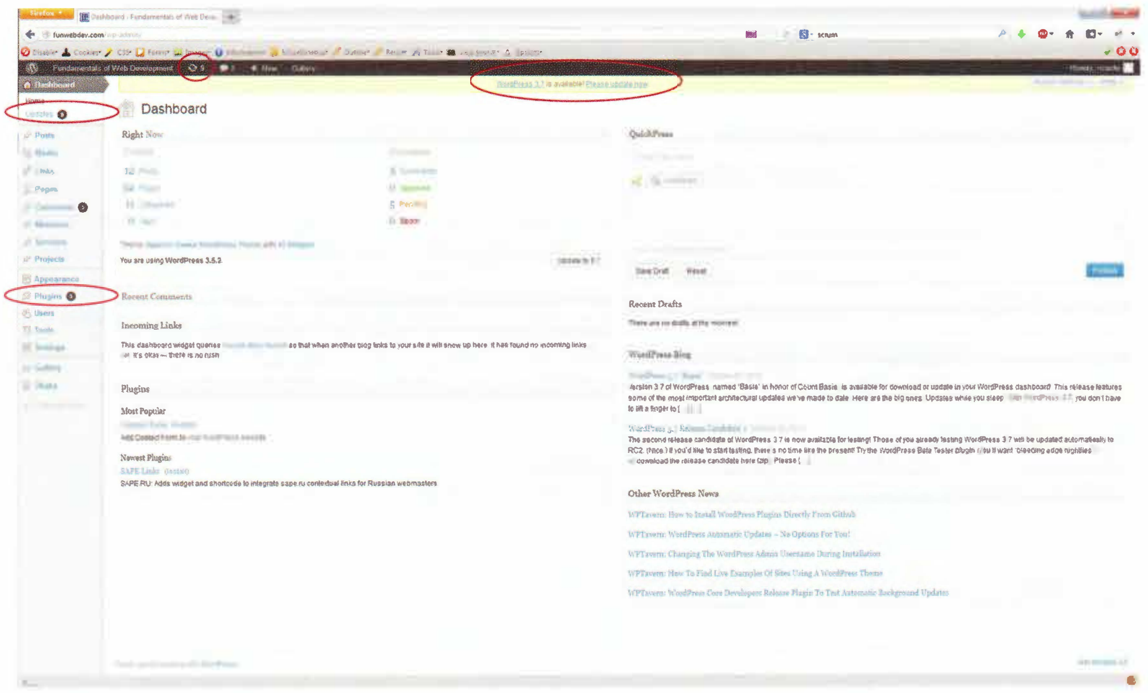
18.3.10 Upgrades and Updates

Running a public site using an older version of a CMS is a real security risk. Newer versions of a CMS typically not only add improvements and fixes bugs, but they also close vulnerabilities that might let a hacker gain control of your site. As we described in depth in Chapter 16, the security of your site is only as good as the weakest link, and an outdated version of WordPress (or any other CMS) may have publicly disclosed vulnerabilities that can be easily exploited.

NOTE

One benefit of open-source software like WordPress is the ability of the developer community to collectively identify and patch vulnerabilities in a short time frame. However, the openness of the identification and patching process provides hackers with a detailed guide on how to exploit vulnerabilities in old versions.

When logged in as an editor in WordPress, the administrative dashboard prominently displays indicators for out-of-date plugins and yellow warning messages about pending updates (as shown in Figure 18.14).



WordPress 3.7 is available! Please update now.

You are using WordPress 3.5.2.

This dashboard widget queries [illegible] so that when another blog links to your site it will show up here. It has found no incoming links … It's okay — there is no rush.

SAPE.RU: Adds widget and shortcode to integrate sape.ru contextual links for Russian webmasters.

There are no drafts at the moment.

Version 3.7 of WordPress, named 'Basie' in honor of Count Basie, is available for download or update in your WordPress dashboard. This release features some of the most important architectural updates we've made to date. Here are the big ones. Updates while you sleep [illegible] you don't have to lift a finger to […]

The second release candidate of WordPress 3.7 is now available for testing! Those of you already testing WordPress 3.7 will be updated automatically to RC2. (Nice.) If you'd like to start testing, there's no time like the present! Try the WordPress Beta Tester plugin (you'll want "bleeding edge nightlies") or download the release candidate here (zip). Please […]

WPTavern: How to Install WordPress Plugins Directly From Github

WPTavern: WordPress Automatic Updates – No Options For You!

WPTavern: Changing The WordPress Admin Username During Installation

WPTavern: How To Find Live Examples Of Sites Using A WordPress Theme

WPTavern: WordPress Core Developers Release Plugin To Test Automatic Background Updates

FIGURE 18.14 Screen of the dashboard with update notifications circled in red

What actually happens during an update is that the WordPress source PHP files are replaced with new versions, as needed. If you made any changes to WordPress, these changes might be at risk. Your wp-config and other content files are safe, but a backup should always be performed before proceeding, just in case something goes wrong. There is also a very real danger that your plugins are not compatible with the updated version. Be prepared to check your site for errors after updating it.

The other complication with upgrading is that the user doing the upgrade needs to know the FTP or SSH password to the server running W ordPress. If you do allow a nontechnical person to do updates, you should make sure the SSH user and password they are provided has as few privileges as possible. Since upgrades can break plugins and cause downtime to your site if unsuccessful, this task should be left to someone who is qualified enough to troubleshoot if a problem arises.