Chapter 8: BUILD THE WEB SITE: OPTIMIZE IT FOR PRACTICAL USE

It's finally time to get your Web site actually up and running on the Internet so that it can be viewed and used by the world. Some people on the team may have been asking why you didn't reach this step sooner, but all the design work and prototyping you've done up to this point will help ensure that the site you actually publish will be useful and usable for your target audience.

To publish the site your team will need to create the actual computer files which will be installed on the Web server where your site will reside. Computer prototype pages will have to be revised so that the use of HTML is correct and consistent everywhere. Graphic images will be produced at this point, and possibly some programming will be done in order to generate forms or other special functions at your site. You know what the site will contain and how it will be organized. Now you have to pay attention to the myriad technical details that actually make it work.

What should already be done?

Needs analysis
You began the design of your Web site by first conducting a needs analysis to determine who the users and the stakeholders are for this site. You identified their needs and desires for the site.

Paper prototyping
Next you created a rapid holistic prototype of your Web site on paper. The content of the prototype should have emerged from your needs analysis. This paper prototype encompassed the breadth of your site so that all major parts were reflected. It also included several strands to go into the depth of your site -- to get down to the specific documents or tasks that users will want to read or to do.

Usability testing
You conducted two or more rounds of usability tests of the paper prototype with representatives from your target population. Based on what you learned from the usability tests, you attempted to fix major problems in your design, and then conducted further usability tests until it appeared that no further major problems had emerged.

Computer prototyping
You built a computer prototype of your Web site that matched the final version of your paper prototype. While the organization of the content should not have changed from the paper prototype, you did need to deal directly with the interactive nature of your site and you may have had to make some concessions to the limitations of HTML and differences between browsers.

Usability testing
You conducted two or more rounds of usability tests of computer prototypes with people who are representative of your target
population. You tested your computer prototype until you were confident that you had identified and fixed any major design problems it may have had.

**What’s next?**

Now you are ready to build your Web site on the computer. This is the actual production phase. Up to now, you were concerned with creating and improving your Web site design. Now, you need to actually implement the design in the Web environment where it will be “living.”

**What about help with software tools for making Web pages and graphics?**

Almost anything we write here about particular software tools will be obsolete by the time this book goes to press. Software for developing Web pages is getting better and often requires little technical knowledge of things like HTML, XML, CSS, graphics file formats, etc. New tools, including those for developing and managing entire Web sites, are continually being developed and improved by both commercial and shareware software producers. Just about anybody can now make and publish Web pages with a great deal of ease. Indeed, the philosophy behind recent software products such as Microsoft's Office 2000, is that Web publishing should be as easy and convenient as using a word processor or spreadsheet and saving the results on your computer's hard drive.

The issue is not really the tools you use, but what you can do with them. Just because you can type with a word processor doesn't make you a good writer. It is the content of the Web pages and their organization – i.e., the design which really matters. This book is largely about how to make that content and design work for your Web audience's needs.

We do have some general advice about selection of software tools beyond their technical features and ease of use:

**Choose Web tools which support interoperability.**

Remember, the reason the Web has been so successful and is growing at such a high rate is interoperability. ‘Interoperability’ means that it should not matter what kind of computer users have, or which flavor of Web browser, or how fast their Internet connection is. Your Web pages should work acceptably for all these situations for public Web sites that are accessed by a target audience with a wide range of technology capabilities. (This may be less true for private intranets with certain technology expectations.) For Web pages to be interoperable, the tools that create them must strictly follow accepted standards for HTML and XML. We'll say more about these standards later, but if you find yourself putting disclaimers on your Web pages that they only work with Browser X (download it now!), or need proprietary plugins (download now!), then you are violating the principle of interoperability. One tool that we have found useful for these purposes is Bobby(TM), which can be found at URL: [http://www.cast.org/bobby/](http://www.cast.org/bobby/). Bobby is free, so you can test your Web pages to see what problems may occur with certain kinds of popular
Web browsers. You can also test for different versions of HTML to uncover problems that users of Web browsers which render older versions of HTML (e.g. 3.2) might experience.

Bobby can also tell you what problems that disabled persons may have in accessing your Web pages (e.g., with visual impairments). And it will tell you the number of bytes of computer information your Web page requires and how long it will take to load using typical communications devices.

Here’s an example. When you start Bobby, it asks you for the address of the Web page you want to test:

![Bobby Test Interface]

To analyze your web site, type in the URL of the page that you want Bobby to examine and click Submit. Bobby will display a report indicating any accessibility and/or browser compatibility errors found on the page. Once your site receives a Bobby Approved rating, you are entitled to display a Bobby Approved icon on your site.

URL: [http://education.indiana.edu/design.html](http://education.indiana.edu/design.html)  Submit

Terms of Use

Bobby's analysis of accessibility is based on the World Wide Web Consortium's (W3C) Web Content Accessibility Guidelines. For example, to become Bobby approved, a Web site must:

- provide text equivalents for all non-text elements (i.e., images, animations, audio, video)
- provide summaries of graphs and charts
- ensure that all information conveyed with color is also available without color
- clearly identify changes in the natural language of a document's text and any text equivalents (e.g., captions) of non-text content
- organize content logically and clearly
- provide alternative content for features (e.g., applets or Flash-objects)
Next, Bobby gives a report by placing “hats” at places on the Web page where issues are identified:

Then Bobby will given approval if the Web page is acceptable, or not based on the issues identified and their severity:
Choose Web tools that your organization can afford and will support.

If many people in your organization will be creating and maintaining your Web pages, then you will need to purchase enough licensed copies of the software so that they will be able to do so. For example, if the Web editor they need will cost $100 per copy (which is not unreasonable at the time of this writing for a good tool), and you have 300 people who need to use this editor, then you will need to budget $30,000. There are often discounts for multiple user licenses, but you still will need to budget for software acquisition.

For example, in the School of Education at Indiana University, we have a relatively small budget for software acquisition. So we trained our faculty and staff to use Netscape's Page Composer. It comes with the Netscape browser (4.x), which is free to educators. It is fairly easy to learn and to use, since it works like WYSIWYG word processors that they are familiar with. Then we purchase small numbers of additional software products necessary for advanced Web development by just a few individuals who can really benefit from them (e.g., for graphics development, site management).

Also, it is important to point out that our Education Technology Services had installed and continues to support the Netscape Web browser, so everyone in our organization had access to Netscape on their desktops already. While we are not advocating one particular browser over another here, the point is we chose a software product for Web development that our organization supports.

Finally, most modern word processing programs usually have an option to save documents in HTML formats (and perhaps also SGML and XML). So, if your organization already has word processing software that has this capability, this may be an option to consider. Just be sure to check that the resulting Web pages are interoperable and do not require proprietary computer hardware, operating systems or software that your target audience may not have.
Accessing a Web server

You will not be able to make your site available to the world without a Web server on the Internet or on a LAN (Local Area Network) server if you are restricting your site to use within your organization only. Technology in this area is changing rapidly, and someone on your team should either be one of the people who maintain your Web server, or should be designated as liaison to the technology specialists who can tell you what changes may affect your site. If you are renting server space from an Internet Service Provider (ISP), it is particularly important for a member of your team to become knowledgeable about the services you are getting.

Find a stable home for your Web site

Be sure to think about the long-term viability, stability, and dependability of the organization who owns and runs the Web server.
you use. Once you publish your materials on the Web, and other folks find them useful, they will be making links to your site from their Web pages. You won't know who all those people are and if you later decide to move your Web site to another server you could inadvertently "break" all their links to your site. You can program your old pages to take them to your new ones, but that can be annoying for you and for them.

When you move all your Web documents to a different server with a different name and address on the Internet it is as if you moved from your physical residence. Even if you leave a forwarding address you are likely to cause frustration for people who are trying to reach you, and you run the risk that some of them will not be able or willing to make the effort. On the Web there may be people who are providing a link to your site from their own pages, and they will be frustrating their users in turn when those links don't work or when they take the extra time to transfer users to your new site.

Another problem may occur with some search engines on the Web, such as Alta Vista, WebCrawler, InfoSeek, Lycos, etc. We have experienced that some search engines may continue to reference the original locations of documents -- long after the documents have been moved. When users conduct searches, the search engine may continue to provide links on the search results page which point to the old locations of content at your Web site. When users attempt to follow
those links, the Web server where your files once resided would report "file not found" errors.

The same kind of problem will occur with bookmarks. Many of us will create a bookmark in our Web browsers when we have found a useful Web page we may want to return to later. This is a very handy feature of many Web browsers, since most of us often don't remember just how we found something on the Web; we just know we want to come back there later. So we make a bookmark. If you move the Web pages we have book marked, then we may not be able to find them easily. In the future, as the Web evolves and matures, this problem may become less severe as methods of link management improve (e.g., in XML – see BYTE Magazine, Feb. 98 – URL: http://byte.com/art/9803/sec5/sec5.htm ). For now, it is a good practice to keep your Web pages in stable locations to minimize the problems we have described above.

Naming your Web site

If you want folks to use your Web site, they will need access. The key to long-term access is having a Web site name that is:

- unique
- as short as possible
- easy to remember
- obviously connected to your organization.

Every Web site name must be unique, since that name is part of the URL (Universal Resource Locator) that distinguishes each document, or page, on the Web from every other one. The name should be easy to remember so that people can find you even if they haven't kept a record of it. Two ways to make your name memorable are to make sure that the name is associated easily with you and your organization, and to make the name short. Short names are also easier to type, for your users and for the members of your production team! Examples of good domain names:

- amazon.com
- byte.com
- education.indiana.edu
- whitehouse.gov
- yahoo.com

Notice that you don't have to put www in front of the domain name. It was never necessary, and just makes for more typing even though it was a widely adopted practice in the early days of the Web. The last part of the domain name is an Internet convention, where .com refers to a commercial site, .gov to a government site, .edu to an institution of higher education, etc. These suffixes are required at this time.

Examples of URLs that are harder to remember.

- www.ed.uiuc.edu/coe/
- www.stanford.edu/dept/SUSE/
- coe.ohio-state.edu/

These URLs are longer, and contain slashes, dashes, abbreviations or capital letters which if not typed exactly right will result in "not found" errors. Users won't be able to browse your site if they make typo's and can't find it. Note also that the first two URLs contain the
domain name (e.g., www.standord.edu) plus a further path within those sites, which makes the URLs longer.

The reason you want a short, memorable name is that the one sure way folks can find your Web site is to click in the location (address) box of their Web browser and type it. Often this is the fastest way to go to a site, if they don't have it already bookmarked. So make it easy on your users. Remember the adage about the three most important things in business: location, location and location. On the Internet, this would translate to: domain name, domain name, and domain name. After all, this is your location in cyberspace.

Your Internet Service Provider should be able to assist you in acquiring and registering the domain name for your Web site. There may be an extra charge for this, so be sure to ask. If you are setting up your own Web server, you'll have to do a lot more of these things yourself (including getting connected to the Internet, establishing an IP number, registering your domain name, and a whole bunch of other things). If you or your organization does not have the technical expertise, you are probably better off working with a good Internet Service Provider.

Developing off-line

The Web is wonderful because of the easy access via the Internet throughout the world. However, as you develop your Web site, the question you should ask yourself and your organization is: "Do you want people to find your documents while the Web site is under development?" Will you frustrate users by giving them false expectations (because almost everything is under construction and nothing is really useful at this point)? Could somebody steal your thunder, and cost your business lots of money? Will it be embarrassing? How would you like a television camera focused on you as you take a shower and get dressed before making yourself "presentable"? This is roughly the kind of exposure you risk by developing your site on-line.

Those peeping search engines

You might wonder, "How will people find our Web site when we haven't told anybody and there are no other documents on the Web yet which have links to our site?" Unless you disconnect your Web server from the Internet, computer search engines and people who use them still may find it.

You have to register your Web site name and specialized servers (domain name servers) keep track of your site regardless of its actual physical address on the Internet. At the same time automated programs "walk the web" building indexes of pages from every domain they find. Search engines use these indexes to look for information requested by Web users. When one of these search engine finds a document that matches a user request for information, it will build a link to that document for the user. One click on the link, and ta-da! You hope you aren't in the middle of getting dressed!

Even if there are no other pages on the Web that point to your new documents, they still could be found by "web walkers" and users, depending on how the Web server is configured at your site. Some configurations require the Web server to return a list of links to all files
in a folder when no file is specified, or when the default name (e.g., "index.html" or "home.html") is requested and does not exist in that folder. In other words, since you didn't tell it which file you wanted to see, it gives you a list of all the files and lets you choose.

**Link-Happy Webheads**

If being found by a search engine isn't bad enough, the person who finds you this way might decide to add a link at their site leading to your page. Anyone who reaches that person's site simply clicks on the supplied link ... and may catch you getting dressed. Worse yet, someone who finds your site may post the URL on a widely-read bulletin board and suddenly lots of people ... watch you get dressed.

This is great if your site is ready to go. Wide dissemination of your URL is something that Web developers dream about. But, if your site is not ready for public viewing, you may want to think twice about doing your development on the Web server.

**Summary**

Develop your Web site off-line and don't move your files to the Web server until you're ready for the world to view them. Ask
yourself, "How undressed can our organization be and still be presentable?"

If you have a development team which must be able to see the work of everyone else who is also developing parts of the site, then you may not have a choice but to develop on-line. It may also be necessary to do testing of on-line databases or search engines in the environment they will exist. If so, then be sure to put disclaimers or "under construction" messages on your main pages so that users who come across your site will have some warning about the state of your undress.

Optimizing Performance of the Web Site

In this section we will cover three major areas: organization of files, performance of the site, and production paths.

Optimizing organization of files

Where do you put your files on your Web server and how should you name them? We recommend keeping your file structure, or hierarchy of directories, as flat as possible and keeping both directory and file names short, meaningful, and easy-to-remember.

Keep your file structure flat

The best way to keep your file structure flat is to put all the files in the same directory or folder on your Web site. For example, if your domain name is funny.net, and you have files named: jokes, limericks, and comics, then your URL's would be:

- http://funny.net/jokes.html
- http://funny.net/limericks.html
- http://funny.net/comics.html

If you put your files into subdirectories or folders, then the URL's might look like:

- http://funny.net/jokes/favorites.html
- http://funny.net/limericks/dirty.html
- http://funny.net/comics/famous.html

The first set of URL’s is shorter than the second, and requires less typing. Unless you really need subdirectories in order to prevent duplication of Web file names in the same directory, then you can avoid the extra baggage by putting your files into a single directory. Of course, on very large Web sites, this may not be practical, and you will need to put files into separate folders or user accounts.
Advantages of flat file structures

The primary advantage of a flat file structure is that the URL (Universal Resource Locator) for any given file in your site will remain as short and uncomplicated as possible. This helps your maintenance staff and the users who are looking for, typing in, or copying down URLs for your pages. Even experienced Web users, who generally copy and paste URLs from one place to another to save time and avoid typing errors, appreciate short URLs that fit easily onto a single line in an email message or can be recalled quickly when the user is out of reach of his bookmark file.
Disadvantages of flat file structures

The main disadvantage of putting all files in a single directory is that each file name must be unique. If the number of files is fairly large you may trouble finding short, unique names for all of them without resorting to cryptic naming schemes. In the case when you have large numbers of files you will need to create subdirectories. You will probably also want to create subdirectories for different groups within your organization, so they can working independently with their respective sets of Web files without inadvertently destroying files by giving them the same names or causing other problems for each other.

Keep file names short

Give your files short, meaningful names. And keep the names in lowercase letters. Some examples:

- people.html
- orgchart.html
- objective.html
- contact.html

Advantages of short file names

Short file names help keep URL’s from ballooning.

File names become part of the URL when links are created in HTML leading to those files, as illustrated above.

Relative addressing.

By keeping all the files in the same directory, you gain one other important advantage. Since they are in the same directory on the same
Web server, links within them can use what is called 'relative' addressing to point to each other. Not only does this make the HTML cleaner, it may decrease Web server response time when relative links are used within Web pages.

**Portability without fuss.**

When files are kept in the same directory they can be moved as a group later to a different directory and all the links will still work correctly. Otherwise someone (or some program) would have to edit all the links within each HTML document that referred to those files. It is very useful when you are developing and testing your site off-line (i.e., not on the Web server itself), to use relative addressing which works from your hard drive and then works equally well from the Web server with no additional editing.

**Summary**

Keep file and directory names simple and clear. Do everything you can to minimize the lengths of paths to files at your Web site because once that site goes on-line and people start to use it they will depend on being able to find your files easily, and find them at the same place every time.

**Keep files small to optimize the performance of your Web site**

When users of the Web click on a link they have to wait while their Web browser makes contact with the server holding the document they want. Then they wait while that file is transmitted, along with any associated graphics files, through the Internet back to their browser. And they wait some more while the browser finally interprets and displays those files. This traffic of messages and files back and forth on the Web is much like the traffic on a highway; when it gets very crowded cars (or files) can sit "bumper to bumper" waiting for their chance to move forward.

**Waiting on the Web**

If you've been using the Web, you have most likely experienced delays from time to time. The standing joke is that WWW stands for "World Wide Wait". According to the Tenth World Wide Web User Survey done at the Georgia Institute of Technology, "(S)peed remains the most cited problem users experience with the Web." (Kehoe, Pitkow, Sutton, Aggarwal & Rogers, May 1999, http://www.cc.gatech.edu/gvu/user_surveys/survey-1998-10/tenthreport.html#ex. There can be many different causes for such delays: the server which has the document you want might be swamped with other requests at the moment; traffic on the Internet may be heavy at any given moment; your Web browser may be running on a computer with a slow central processor or video display; or you may have a relatively slow communications device connected to the Internet (e.g., 28.8 Kbps modem or slower). The most common cause of delay, however, is the size and number of files being transmitted to your local computer in order to display a Web page. If a Web page has associated graphics, then in addition to the HTML document, one file for each of the graphic images is also being transmitted while you wait.
More bytes, more waiting -- ugh!

We have seen repeatedly during usability tests that users don't like to wait. In fact, most users of the Web -- possibly including you -- will not wait for longer than 15 - 20 seconds for a page to display before they give up and click somewhere else. According to Terry Sullivan (http://www.pantos.org/atw/), "readers are mad as hell, and they aren't going to take it anymore."

"Analysis of traffic patterns at a commercial Web site helps to illustrate just how intolerant readers have become with slow-loading Web pages. The site in question had a large "splash screen" that took nearly 90 seconds to load... It's not surprising to discover that some percentage of readers simply hit their [STOP] buttons and surfed away. But the sheer size of the phenomenon was noteworthy. Nearly 25% of all visitors to the site never made it past the splash screen. They never entered the main site.

The implication is clear, and confirms the conventional wisdom from human factors/usability disciplines: you have a matter of


The more bytes of information that need to be transferred from the Web server to the user's computer, the longer users have to wait -- regardless of any other delays that might be affecting them at the time. When files are transmitted, they are broken into packets of information. Each packet is routed through the Internet separately until it reaches its destination. The computer receiving the packets then reassembles the file in the correct order even if some packets arrive before others. No one packet gets priority over any other. Thus, all other things equal, files consisting of more packets (bigger files) will take longer to transmit than files with fewer packets.
How can I test the size of my Web pages and estimate the wait time? You can test your pages with a Web service called Bobby (http://www.cast.org/bobby/). Not only does Bobby tell you about the interoperability and accessibility of your Web pages—as illustrated earlier in this chapter—it will also estimate the load time of your page and its associated graphics.

This can help you identify the problem of what files may be increasing the load time and making your users wait unnecessarily. The basic strategy is to decrease the size in bytes of the files to be loaded. Some tips for doing this are given by Terry Sullivan at: http://www.pantos.org/atw/35305.html.

Won't technological advances make all these "economy measures" unnecessary in a few years? Won't Internet connections be much faster in the near future? Yes, they certainly will, but reasonable "economy measures" are always going to be important in delivering Web pages and users will always appreciate Web files that load quickly. Users should not have to wait more than 15 seconds as a general rule of thumb for most parts of your Web site. Users with faster communications devices will wait even less, and even when there is a lot of congestion on the Internet, your smaller files will get through the traffic in less time since there will be fewer packets transmitted between the server and the user's computer. Remember, if you make them wait, they may not stay long enough to see any of your Web site.

**Optimizing Web page production and maintenance**

During the design and prototyping phases of your Web project you began thinking about consistent style for your Web pages, and you
began keeping a record of decisions that should apply to all the pages in the site. This record should now begin to serve as a style guide for all the different people working on the final production. It should help establish a consistent look and information standard for all pages, even though it may leave plenty of room for creativity to be exercised by the designers in different parts of your organization. Since you have been keeping stakeholders informed about your work, including those who will be information providers for the whole Web site, you should have some agreement by this time on the minimum conventions to which everyone contributing to the site will adhere.

**Make templates to promote efficient development and to avoid introducing errors**

Develop and thoroughly test templates for your HTML documents. You may have several different kinds of templates, to be used in different parts of the site. Your templates should include some standard top-of-the-page information, and some bottom-of-the-page information. In addition to the parts of the Web page which give your site identity and a consistent "look and feel" as discussed in an earlier chapter (WHICH ONE???), you should also consider the following:

**Crediting work**

In our experience, Web page developers appreciate being acknowledged for their contributions in page credits. In addition to bolstering morale, this practice also makes clear who was or is responsible for the page.

**User feedback channel**

It is also important to provide a way for users to make comments or ask questions when they visit a page on your Web site. Although few users may write to you with in-depth criticism or praise of your site, they will know that you are willing to hear from them since you have created a way for them to contact you. This increases the credibility of your site overall (in fact, inclusion of a "mailto:" link on Web pages is one criteria listed in a number of current check sheets for helping Web users determine the quality of information in a Web site). If your site proves to be very heavily visited, you will probably get a steady stream of messages asking about items that people couldn't find or notifying you of errors and problems with your pages. When you fix the problems and answer the questions right away you increase people's positive impressions of your Web site and of your organization. Your "mailto:" link (or a feedback page) should also increase the chances that someone will notify you of glaring errors or technical problems with your site very quickly so that you can address problems before they inconvenience all your visitors.

**Copyright notice**

Your pages are copyright protected even if you do not include a notification of copyright on them. However, we encourage you in the strongest terms to develop and display a copyright notice on your pages when you develop a site for your organization. Many visitors to the Web erroneously extend the original notion of "open Web culture" to assume that anything and everything they see is free for the taking. It is also extremely easy to capture text and graphics with a few clicks and drags of the mouse on the Web. You give your organization vastly improved means to confront and stop the unauthorized use of its
materials when you display a standard copyright notice on all your pages.

**Parts of the template that are "hidden" from users but very important**

Your developers need to be reminded about three very important parts of every HTML document:

1. the title,
2. a meta tag for keywords, and
3. a meta tag for description.

**Title:** The title of a document is usually rendered in graphical web browsers in the very top of the browser window, above the buttons and menu bar (i.e., not in the scrollable window of the Web page itself). Although most Web users do not pay attention to the title, it is extremely important to include a unique, descriptive title in every Web page. This is part of the document `<head>...</head>` structure, and goes in between the `<title>` and `</title>` tags. The title should describe the contents of the document in language that would make sense to someone encountering it "out of the blue", much like titles of books in a library or bookstore. The title ought to give the user a good clue as to what is in the document.

Now, why would a title be so important if most users don't pay attention to it when browsing the Web page? (Note that the title should not be confused with headings (e.g., `<h1>...</h1>`, `<h3>...</h3>`) that occur in the body of a Web page and which do appear in the scrollable window).

First, if you have ever done a Web search and you get a list of results, it is the titles that appear as links on the results page, as illustrated here.

In the above example, we did a search for Web documents containing the keywords: web design. Notice the underlined links.
These are the titles of the documents that the search engine found in its index. Normally, users won't know anything about your Web page when seeing the results of a search for the first time, when one of the titles of your pages shows up in the results. A good title will help those users decide whether to follow the link or ignore it. Also, many search engines return a line or two of text below the title. That text is typically from the first part of the document itself, or it is what the document author put in a meta tag for the content description (e.g., <meta name = "description" content = "my page description">
).

Meta tags: Like the title, meta tags also occur in the HTML <head>...</head> block. In addition to the document description, there is also a meta tag for keywords (e.g., <meta name = "keywords" content = "put a list of keywords separated by commas here">). Search engines such as Alta Vista, MetaCrawler, Lycos, Excite, Infoseek, etc., have what are called robots (spiders, web crawlers), which are computer programs which literally go visit Web pages by following links on other Web pages. When a Web page is encountered by a search engine robot, it is indexed for later use by the search engine. Keywords, words in HTML document titles, and words in HTML headings are often given more "weight" in the index, as well as words which occur more frequently within a given Web page.

In short, you can increase the likelihood of your Web page turning up on the results of a Web search by using good titles to describe your documents, and meta tags for the document keywords and description. While these parts of HTML documents are basically "hidden" from users, they are very important for users who do Web searches getting some extra help in finding your Web materials. It also helps to design good headings within the document body, which users do see. Not only do headings help users understand the structure and organization of your Web page, words that occur in HTML headings are also often given more "weight" in search engine indexes.

While this discussion has been a bit technical, delving into some HTML, we feel it is important nonetheless to discuss titles and a few special meta tags because these are often glossed over by many Web developers. How do you make page titles and meta tags? Usually there are features of good Web development tools that make it easy to insert these elements of a Web page often by filling out fields in a short form. Then the tool generates the correct HTML for you.

**Thoroughly test the template before handing it off**

Bug test your templates on a variety of Web browsers and computer platforms until you were satisfied that they work acceptably in a variety of conditions. Failure to carry out this kind of bug testing of the templates can result in the replication of mistakes or errors across the whole site. There are now Web development tools that can make widespread changes to a while set of Web files fairly efficiently, and some Web servers can build documents "on the fly" by inserting standard headers and footers for your organization. Even so, every change that has to be made all across your Web site represents some loss of time and, if the error is allowed to go on-line, may represent some loss of credibility.

**Summary**
Whatever the design you create, it needs to inform the user of where they are, what they can expect to find on this page, and where they can go from here (especially to other parts of the context of your organization). The page needs to have an identity, so it is clear to users who drop in from a search engine or a link at another site who you are and where they are.

Once you have a set of well-tested templates, the people on your production team can proceed to make the final graphics and add the content elements to every individual page. They won't have to make the big design decisions as they go because those are embedded in the site structure and templates. If you bring production help in late in the project, be sure to go over the previous work that has been done or you may find that they are changing elements of the Web pages that you spent time and usability effort to design. If you have keep these production people in the communication loop from the beginning of the process you can be more confident that they will create the individual pages successfully because they will understand the rationale and the usability data behind your design decisions. You may want to invest in demonstrating your templates to other stakeholders at this time too, so that you know they will recognize and be satisfied with the resulting pages.

The Next Generation of the Web

Parts of the advice we have given above about templates will become somewhat obsolete as the Web evolves. It will still be important to have descriptive titles, standard headers and footers, and links to the context in which a Web page exists. However, the methods that Web site developers will employ will change.

We remind the reader that the Web is still very young, relatively speaking -- less than a decade old at the time we are writing. One major strength of the Web is interoperability. Millions of folks from all over the world can use it with different computer systems with different operating systems -- even different languages and symbol systems. Standards for HTML, which browser programs follow when rendering Web pages, make this interoperability possible. Interoperability has helped to make the Web very popular. This popularity in turn has resulted in demand for functionality that exceeds what HTML was originally designed for. We have seen versions of HTML standards jump from 1.0 to 2.0 to 3.2 to 4.0 in a matter of a few years. We have also seen competition among Web browser software producers create non-standard features of HTML. Unfortunately, this defeats the goal of interoperability.

These experiences, coupled with user demands for more and more functionality, have led the W3 Consortium to consider a new set of standards. The new standards will make HTML extensible, and are referred to as XML (extensible markup language). Since these new standards are under development as we write, we only note some important features that will make development and maintenance of Web sites easier than it now is.

For example, if you use the "template" approach that we describe above, and you decide to make major renovations at your Web site, you need to modify all of your Web files. You may want to change the overall appearance of Web pages, but the content may not change
significantly. XML will make this much easier to manage, since it separates (or unbinds) the style of a Web page from its content. Under XML, you would only need to modify the style definitions, and presto! the appearance of your Web site changes everywhere to follow the new style.

Furthermore, you can develop different styles to accommodate different user needs. This will become increasingly important as different kinds of information appliances connect to the Web, which render information in new and different ways. For example, listening to a Web page is different than visually reading it. You could define a style for "listeners" (e.g., for car-drivers or visually impaired people) which is different than a style for "readers". Yet the content of your Web files would remain the same. You wouldn't need to go in and make changes in the HTML in each file as you are now required to do. Instead, you would edit your style definitions. You could have a style that accommodates those slower communications systems we discussed above, and a different style for users with faster devices; one for tiny-screen devices, and one for large-screen devices; and so forth.

XML-capable Web browsers will also be able to render conventional HTML. When XML Web browsers become widely available, you should consider building your Web site to take advantage of these extended capabilities. This chapter has addressed issues that are relevant at the time of this writing.

Stay tuned.

**** Add a link here to updated versions of this chapter on the Web itself.