A review of "Web Design on a Shoestring" by Carrie Bickner

By Bruce Lawson

The title of this book is both accurate and simultaneously misleading; it does teach how to make web sites on a shoestring, but it also feels like it's going to tell you how to use that copy of FrontPage that you got with MS Office, together with some clipart and animated GIFs in order to make a table-heavy, midi-muzakked "killer homepage".

I believe that, at some point during the writing, it was codenamed "how to make a site on a shoestring that looks a million dollars" which would be more accurate (although a little cumbersome and too big to fit on the book's spine!). It's a book for web designers with a low budget (whether in the public sector, freelancers or – let's be frank – practically everyone these days).

The author, Carrie Bicker, has an excellent pedigree of working on some good, non-homepage, sites: she works for The New York Public Library, where she is The Digital Library’s Assistant Director for Digital Information and System Design. From the autumn of 1999 until the spring of 2002, she was The Branch Libraries' Web Coordinator. The best known project from this stint is probably the NYPL Online Style Guide, which shows how to author web sites in valid XHTML 1.0 Transitional and Cascading Style Sheets, co-written with Jeffrey Zeldman of The Web Standards Project, so it’s a fair bet that, as a librarian, she doesn’t have a monster budget, but is adept at making industrial strength sites.

Let's look at the book in more detail:

Chapter One – Secrets of a Successful Shoestring Project

1. **Uses the resources at hand.** You've got PHP on a machine and know how to use it? Then use it; don't decide ASP.NET will be better and spend time installing it, learning how to use it!
2. **Is managed by a small group of decision makers.** Remember, a camel is just a horse that was designed by a committee.
3. **Has a clear focus.** Few things eat budgets faster than wasted man hours changing tack half way through, navel-gazing or not specifying a project so scope-creep sets in.
4. **Dares to do less.** This is a phrase from Tim Bray, the co-inventor of XML, who says that technologies that are successful do one of two things very well, rather than trying to be a jack-of-trades. It becomes Bickner's mantra throughout the book. Basically, if you've got limited resources, do what you can do well, and don't attempt unnecessary frills like flash intros, cunning 360° panoramas on a live web cam of the stock room of the furniture shop that the web site is for. You don't need it; it doesn't add value to the user, and it'll eat your
A review of “Web Design on a Shoestring” by Carrie Bickner

Budget faster than my daughter can make herself chocolate-sick on Christmas day.

A project such as this, with in-house photography, simple and clean design can be seen at http://www.nodltd.com/ronco/index.shtml

Chapter Two – The Pound Wise Project Plan

You don’t need my summary of this! You can read this chapter below starting on page 5!

Chapter Three – Usability on the Cheap

If you’re on a budget, you won’t be able to hire Jakob Nielsen, that’s for sure. And the danger in getting designers and clients to do usability testing is like asking engineers to write manuals – impossible to do well because they’re too close to the project. Any web designer who neglects the needs of the site’s audience at their peril, but unfortunately, it’s often missed out of shoestring projects as it’s perceived to be really expensive. This chapter shows some strategies for on-the-cheap usability testing.

Chapter Four – Why Good Copy Counts

This chapter writes Bickner “is for web professionals who are responsible for many aspects of web production”. Basically, no matter how great a site looks, no matter how intuitive its navigation, if the reader can’t understand the content, it’s wasted time and effort - and loads of typos never inspires confidence, either. She discusses how to write for the web (be careful of just using the text from printed company literature; what’s readable on the page might not be readable on the web) and HTML ways to enhance copy: a customised 404 page, the pages’ title tags, the links and title attribute of the links can all enhance the copy on your site.

Chapter Five – The Design: Looking Good for Less

This was an eye-opener for me, as my design skills are of a similar standard to my lion-taming abilities. Image editors and typefaces are dangerous weapons in the hands of geeks like me – and that’s what happened in the early days of the web; half a dozen different typefaces on each web page, in different colours, animated gifs (because you can!) all make for an amateurish, cluttered feel. To return to Bickner’s mantra, dare to do less. She explains how to match up typefaces, which types (serif/ sans-serif etc) are good for text and headers, and how to use text-in-images for cross-browser integrity of “branding” fonts.

The Art and Photography section was well thought-out. Once I’d read it, everything seemed to be common-sense, but before I’d read it, I’d never thought of some of the points she made – which is the kind of feeling I get when I know I’m learning some good lessons. Points like, if you’re using free clipart/ photography, choose
pictures which harmonise well together, as well as just being individually appropriate. If the pictures you want seem to clash, try running them through your favourite image editor and filtering or tinting them similarly – and crop them to the same size and give each an identical border. It’ll make the whole site appear much more professional. There’s also an excellent list of resources on the web of cheap, good quality art-work. Each of those is now bookmarked in my favourites.

One aspect of this I disagreed with; Bickner writes that you shouldn’t stint on the tools you use, and recommends Adobe Photoshop which costs $649 from Adobe. Photoshop has snob value, sure, and is the only tool of choice if you are making 10 by 5 meter advertising hoarding pictures, high-resolution print magazine pictures, or serious image manipulation (cleaning up seriously damaged images, for example) but for cropping, converting, tinting and saving for 72 dpi web graphics, Fireworks is fine. Otherwise, PaintShop Pro from Jasc costs just $79, a fraction of the price of Photoshop, and can accomplish the common web image tasks just as well as Photoshop for a fraction of the price. If you’re really hard-up, the GIMP is free. (Early in the new year, we’ll be running a DMXzone tutorial series on how to make use of free or low-cost software in your web development; the GIMP is one of the programs we’ll be looking at).

**Chapter 6 – Content Management on a Tight Budget**

Choosing a CMS is a tricky business; commercial software exists ranging in price from free-for-personal-use, to over $100,000. A CMS is seriously worth considering if you are handing the web site over to the client for content updates in the future (although you’d probably recommend Contribute), if you will have more than one person updating content, so you need to be sure that you versioning and check-out, or for a myriad other reasons. One thing about a CMS is that it’s always easier to add it at the start when there’s not much content, than migrate existing static content into a CMS at a later date.

This chapter has an excellent check-list to ensure that you don’t over-specify and get a CMS that is way beyond your needs. There are also comparison charts that cover a range of possible CMS choices for shoestring web sites.

If you’re considering a CMS for a site, this chapter is worth the price of the book by itself.

**Chapter 7 – Save Time and Money with Web Standards**

If you’re new to Standards, this is a very good introduction to the subject that doesn’t preach because Standards are the latest fad, but shows you how to implement XHTML and CSS because they can save you time – and hourly pay is often the largest component of the shoestring budget. Does what it says on the tin.
A review of "Web Design on a Shoestring" by Carrie Bickner

Bruce Lawson

Chapter 8 – Bang-For-Your-Buck Hosting and Domains

In the main, aimed at a U.S. audience, who have the misfortune to go through Network Solutions/Verisign, but because the chapter names no names, and suggests a process and some useful pointers, the advice given on how to choose a good hosting company is applicable anywhere.

Executive Summary

This book definitely does what it says on the cover and more; the sites you'll create if you follow its advice will be better looking than the majority of current web sites out there on the web. Even if you are an experienced developer or designer, there's sure to be some useful tips you'll pick up.

It's nice to see New Riders publishing practising what their author preaches; the book is cleanly and carefully laid-out in black and white (to keep the cover price down, presumably) and lacks the typographic excesses that polluted some previous books like blink tags on a Geocities web page.

This slim book doesn't feel very big, and it isn't cheap at $24.99, but that's much cheaper than most computer software available and will certainly impact as much on your quality and productivity as most labour-saving software packages do. I recommend buying a copy, as it is full of good ideas, links, tips and project management suggestions that is almost certain to save you several multiples of the cover price.
Chapter 2: The Pound Wise Project Plan

Chapter Checklist

1. Dare to do less.

It will be tempting to load up your project with every wonderful idea that occurs to you and your colleagues. Remember that your budget is small; you are better off if you scale back and do a few things well. Save the expensive ideas for a later phase of production.

2. Write a short project goals document.

If your project plan is vague, your small budget will be eaten up by indecision, rethinking, and patch-up work. Make sure that you start the job with a clear project goals document.

3. Create a functional requirements document.

It pays to be explicit about each of the functional requirements for the site. If you do not have a list of functional requirements in place, you will pay through the nose, wasting time and money on production work that you don't need.


Take the time to list technical requirements such as target browsers and system specifications, and use this list to test the site as you build it. If you wait until it's finished to test it against your technical benchmarks, your changes and bug fixes will cost far more than they have to. Never put yourself in a position where you have to pay for expensive post-production fixes to easily avoidable problems.

5. Keep documentation nearby.

It is important not only to write project goals and functional and technical requirements, but also to read them—and make sure your colleagues read them, too. I like to keep this documentation short and pin it to the bulletin board, where it serves as a constant reminder of what I'm trying to achieve.

Oh, I realize it's a penny here and a penny there, but look at me: I've worked myself up from nothing to a state of extreme poverty.
—Groucho Marx in Monkey Business (1931)

The promise of this book is to show you how to build a wonderful site on a shoestring budget. Although these pages share important techniques for working through each phase and aspect of a shoestring site, the overall success of your project depends on the planning work that you do at the outset. After all, the success of any project is the result of good organization and a straightforward concept. This is especially true for shoestring sites. When you have little money, you can't afford to take on the bloat that accompanies ill-defined goals and poorly organized work plans.
This bloat often goes unnoticed in big-budget sites. Larger budgets afford web professionals more opportunity for indecision and lack of direction. An unclear path leads to a major eleventh-hour revision; labour is wasted on last-minute fixes for problems that could have been avoided through better planning.

One of the pleasures of working on a dime a day is that you are required to run lean. Shoestring web professionals, forced to take a focused, stripped-down approach to a project, are a bit like adventure travelers who can’t afford to pack their rucksacks with frivolities. The planning phase of web site production is like packing for a trip. When you plan for a small-budget site, you need to be selective about what goals and requirements you bring along.

**Careful Planning Pays**

In 1996, mountaineer Rob Hall led a team of climbers on what turned out to be a disastrous expedition to the top of Mount Everest. Commercial tourism was big business, and what were arguably underqualified climbers had been able to buy a trip up what was becoming an overcrowded mountain. One such climber, a wealthy socialite, brought an espresso maker along on the expedition. She survived the journey—no thanks to the extra weight she was toting, which at such altitudes and temperatures posed a serious and foolish risk. But the lady and her coffeemaker became a symbol of the excess that commercial Everest tours had become. That espresso maker is exactly what web professionals need to look out for as they approach web sites with limited budgets.

Overstuffed backpacks have undermined many sites, no matter what the budget is. Before any journey, it is tempting to load up on all of the fabulous things that you might use. But low-budget travellers must pack with care. Think of your shoestring site’s development as a trip up a tall mountain, where any bit of extra weight is a burden that you have to carry. Pack too much, and it will cost you.

I have never done any mountaineering, but I used to rock-climb. Most of my crag time took place in a climbing gym. My climbing friends and I scaled 70-foot slabs made of plywood and synthetic rock formations. The gym was pricey, and I think it made most of its money by selling beverages. Bottled water was about $3 a pop. These were my graduate student days, and I did not have money for extras, so I brought tap water from home. Pretty simple stuff, but I probably saved hundreds of dollars each year that I climbed there.

**Budget Threat: Treating Small-Budget Sites Too Casually Can Be Expensive**

I have enjoyed a modest supplemental income from creating small-budget sites in my free time. I learned the hard way that neglecting the planning phase of such site costs money in the long run. By failing to plan and budget for basic things, such as the cost of new fonts, the time to produce web-ready images, and the labor of authoring content that was supposed to have been created by someone else, I have worked more hours without pay than I care to add up. Whether you are working freelance, in an agency, or in-house, don’t let poor planning eat away at your budget this way?
So often when web professionals approach a small site, they don’t take the time to plan. A low dollar amount can lead people to think, "This is a minor project that I can do in my off hours. It is a casual job that will evolve on its own." Don’t ever put yourself in this position! You will pay dearly for taking that attitude. If you are working on a flat fee, you will not recover your costs. If you are working on a billable-hour schedule and you pass that cost on to the client, shame on you! If you are working full-time as part of an in-house team, you will pay personally by working late and on weekends to make up for time lost to lack of vision. One way or another, lack of planning will cost you big-time.

**An Ounce of Prevention**

If you plan well, your project has a better chance of staying on budget. Specifically, you need to address three steps formally and rigorously:

1. Define the goals of your site.
2. Define the functional requirements.
3. Define the technical requirements.

None of these steps needs to be overly involved, but you do have to take the time to write each definition in a document. These documents will help you and your budget stay on track.

**Budget Threat: Scope Creep**

An unstoppable phenomenon is common to all web sites: scope creep. The development of a web site is hard to pin down. New ideas surface, and needs change. The scope of your project expands exponentially as new ideas and new "needs" are added. Scope creep can be good or bad, depending on how you manage it. The trick is to direct these shifts by making conscientious adjustments to the definition of your site and by carefully adjusting the budget to fit the new needs. Clearly defined project goals and carefully articulated functional and technical requirements will help you stay sane and in the black as you begin to experience scope creep.

**Shoestring Project Goals**

Before you begin to work, sit down and take the time to write out what you are trying to achieve with your site. Let’s work with a hypothetical example. The following is a freelance scenario, but the principles apply to any web-development circumstance.

**Sample Site: Project Goals for the Something Blue Site**

Imagine that an artist, Mary Hoy, has asked you to create a site based on her last exhibition, Something Blue. She can pay only a pittance. You like her work and figure that a pittance will help pay the bills, so you take her on as a client. Mary has never been involved in web site creation, so has no way of shaping the scope of the site. You correctly anticipate that her lack of experience might lead her to change her goals and requirements over time. When she sees what her work looks
like on the web, she’ll be inspired to try new ideas or to add functional requirements to the site. You have anticipated these problems. To protect yourself and the budget, you formalize the project’s goals:

**Project Goals for the Something Blue Web Site**

Something Blue will be an online exhibition of the works of artist Mary Hoy. The site will include 15 of the 40 images in her bricks-and-mortar exhibition (also called Something Blue) that took place at the Good Girl Gallery in Studio City, California, in July 2003. In addition to the 15 images, the site will include 2 interviews with the artist, her resume, and her contact information. The purpose of the site is primarily to promote her work, to give Hoy an online “calling card,” and perhaps to lead to a sale or two, although no selling will be done directly on the site.

This is a good example of a budget-sensitive project goal document because it is specific. First, you have established the genre of the site: This is an exhibition, not a retrospective or an e-commerce site. Second, you have quantified the scope of the project. The difference in the cost of producing 15 scans versus 40 scans might add up to quite a bit. You are probably going to be the one doing the scanning and the image editing. This will take time, and creating a schedule around a specific number of images will help you stay on track both for the schedule and the bottom line. The same is true for the interviews and the contact information; you can anticipate the number of web pages that you will have to build. As you put your budget together, you can attach a dollar amount to each major site element.

**Budget Threat: Keep the Number of Decision Makers Small**

As explained in Chapter 1, “The Secrets to a Successful Shoestring Project,” one of the secrets to creating a successful shoestring web site is to keep the number of decision makers small. The more people you involve in the goal-setting phase, the more diffuse your project goals might be. Good resource allocation requires a carefully defined project. If a large group of decision makers is keeping you from nailing down the project goals early in the game, try to pare the number of decision makers to a few key people. If you don’t, you will spend big bucks on unfocused work that goes nowhere.

The project goals document cannot save your budget or your sanity if it is not shared with the site’s stakeholders. In the Something Blue scenario, the only decision maker is your client, Mary. Most sites, however, involve more stakeholders. You must communicate with all your stakeholders without turning them all into decision makers. The more levels of approval that are required to get the site moving, the more involved and expensive the process will be. More important, the more people are involved in a creative process, the more diffuse the focus can become. Distinguishing between stakeholders and decision makers is not a license to be cavalier; don’t blow off the big wigs because, if you ignore them up front, they will be inclined to kill your project just before it launches. Talk about a waste of time and money! Make sure that your group of decision makers is small, but be politic in selecting the composition of that group.
Chapter 2 of “Web Design on a Shoestring”: The Pound Wise Project Plan
by Carrie Bickner

Definition: Client

In this book, I use the term client to refer to anyone who has engaged you for web production. This could be a single client who has hired you to do freelance work, your direct supervisor, or in-house colleagues who depend on you to develop the company’s site.

When you have formally established your group of decision makers, be sure to share the project goals document with the whole gang. It does not matter whether they are clients or colleagues. It does matter that you all agree on the goals of the site. If the goals change later, someone will have to pay for the extra time and materials; if you have formal sign-off from clients, colleagues, and bosses, you lessen your chances of getting stuck with the tab.

A formal document that nails down the initial scope places you in a much better position to ask for more resources when, for example, artist Mary Hoy decides that she wants 30, not 15, images of her work. You should never have to pay the price when someone else has a change of mind.

Nail Down the Functional Requirements Early

Establishing goals is the best way to keep control of a shoestring site, but you need to buttress that work with clearly defined functional requirements.

Let’s return to Mary’s Something Blue site again and create a functional requirements document. Again, this example assumes that you are a freelance web professional, but the same process can and should be applied by professionals working in any environment. You can start by interviewing your client about what she wants the site to do. The result of your interview will be a broad wish list that might include the following:

- I would like my visitors to be able to search the site for images and, when they click on them, be able to zoom in and see each piece, down to the detail of the paint strokes.
- My users need to be able to scan the images quickly and then pick the image that they want to see in more detail.
- My mother has a slow dial-up connection to the web, and I want her to be able to look at my site. I hate sites that take a long time to load.
- It must be easy for users to contact me; I want them to be able to call or send me an email if they want to buy a piece or show my art.

Clearly, not every item on such a wish list will make it onto a site with such a small budget. Your job as the web professional is to let the client know that you have heard all of her ideas and that you take them seriously, and to pare the wish list to a set of functional requirements that can be delivered on budget. You’ll need to massage the wish list into a functional requirements document that can be accommodated by the resources at hand.
The result of this message should look something like this:

**Functional Requirements for the Something Blue Web Site, Version One**

The Something Blue web site has possibilities that are broad and exciting, but a few of these are too ambitious for the initial launch. Taken all at once, the potential requirements would result in a budget that exceeds the resources at hand. We have boiled them down to a few complementary requirements that can be achieved within the current budget and timeline. Some of the more expensive requirements that did not make it into this proposal should be saved for a second iteration of the site.

**Possible Requirements for Next Phase of Web Development**

<table>
<thead>
<tr>
<th>Requirement</th>
<th>Notes</th>
<th>Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>Users must be able to click on a small version of an image to see a larger image.</td>
<td>This is an economically viable alternative to the desired pan and zoom technology that would allow users to zoom into each image. It will not allow users to zoom into such detail as paint strokes, but it will afford a nice size of viewing area. The larger image will be about 600 pixels wide, a size that accommodates most computer screens. An added benefit of this alternative is that users with slow Internet connections will not be forced to view the large image unless they choose to.</td>
<td>$</td>
</tr>
<tr>
<td>Users must be able to quickly scan all images by scrolling through a strip of small thumbnail-size images.</td>
<td></td>
<td>$</td>
</tr>
<tr>
<td>The site must be quick-loading and easy to use on a slow Internet connection.</td>
<td></td>
<td>$</td>
</tr>
<tr>
<td>Users must have quick access to contact information.</td>
<td></td>
<td>$</td>
</tr>
<tr>
<td>The site must have a search engine.</td>
<td>This could be expensive and might not be necessary now because you are working with such a small number of images. But you can make this an item for another phase of production, perhaps when you add many more images to the site later.</td>
<td>$</td>
</tr>
</tbody>
</table>
Chapter 2 of “Web Design on a Shoestring”: The Pound Wise Project Plan
by Carrie Bickner

Note this well: If you can’t get your client to scale back the requirements to fit the budget, this is your chance to walk away. Nailing down an unambiguous vision of what will and will not be on the site is absolutely essential. If you can’t get there, everyone will be unhappy in the end. You will work long hours that you will not get paid for, and your final deliverable will not satisfy your client.

I can’t emphasize enough that the most important factor in creating a site on a shoestring budget is to consciously restrict the scope to include a limited set of achievable functional requirements. Remember that less is more; it is much better to do a few things well than to try to do too many things and fail.

### Spinning Straw into Gold

Even with a big budget, you will find it hard to include every item on a client’s wish list. Knowing which requirements to table for the short term will help you deliver focused sites on time and on budget. The items that you table for a second iteration might butter your bread later. A happy client might call you in the future when she has more cash to expand her site. Many freelance web professionals are staying in business these days by working with long-term clients who provide a steady, if small, stream of work. Help yourself develop a list of clients like this, and you might begin to enjoy a little stability.

Every time you are asked to expand or change the scope of your site, pull out the functional requirements document upon which you and the client agreed, and discuss how you might best modify this list to accommodate the new request. You might need to drop one requirement for another. Let the client know about these trade-offs, and ask her to help you evaluate them. This is also your time to talk about the additional resources that are necessary to add to the list.

As discussed earlier, the scope of your site might change as work progresses. If Mary the artist tells you at the halfway point that she also wanted to sell art from the Something Blue site and now she needs a shopping cart, you will be able to show her what this change means to her bottom line.

Each time a client change compels you to modify the functional requirements document, give the document a new version number. Keep the older versions of the document so that you can see how the project is evolving (or devolving). If version four of the document differs significantly from version one, make sure you have the resources to carry out the project as it has come to be defined.

Often the new ideas that clients and bosses come up with will be easy to accommodate, and you will not have to ask for more resources. Update the document anyway, adjust the version number, and make sure your client or boss sees it. If you do this consistently, the client will at least understand that all changes to the scope of the project must be acknowledged formally. This will also discourage frivolous changes and help keep you from having to pay the price for scope creep.
Establish Your Technical Requirements

Defining your technical requirements early is just as important of a money-saving device as defining your project goal or functional requirements. In the first years of my career as a web professional, I worked on a few sites whose technical requirements were left vague. This lack of clear technical requirements sometimes came back to haunt us. In one instance, we hired a consultant who developed a beautiful site for us. The site was wonderful in the current version of the Netscape browser, but it broke when we viewed it in Internet Explorer. Our target browsers had not been formally identified, and the consultant had assumed that the site needed to work only on Netscape. (As you can tell, this is not a recent story.) After a little tooth pulling, our consultant adjusted his markup and code so the site would work on a wider set of browsers. He also picked up the bill for the extra hours of labour. If we had defined our target browsers early, however, he would have saved time and money, and we would have saved the administrative overhead that was required to facilitate the fix.

Clearly defined technical requirements can keep you from burning up your budget on these kinds of post-production fixes. Explain your needs simply, in plain English. When you have created this language, you can use it again in subsequent Requests for Proposal (RFP) and other web-development documents.

If you are a client, make sure that you turn in this information to your vendor. If you are the vendor, you might have to help your client determine technical requirements. The initial work might seem time-consuming, but it will save heaps of time and money in the long run. Here are some technical requirements that I have been using that have successfully saved me time, money, and stress.

Technical Requirements Checklist

Some of the items that should be included on your technical requirements document include these:

- **Target browsers and operating systems** — Do you care about 4.0 browsers? Only modern browsers? Handheld devices? Do you care whether your site works on a PC, or are your users on Macs? Be sure to list these receiving devices explicitly, accurately, and carefully. Use this list to test browser performance as you go. (Don’t wait until the project is finished to test it in the targeted browsers and devices. Fixing problems that late in the game is far more costly than spotting them earlier in the process.)

- **HTML and CSS** — Do you have a particular HTML and Cascading Style Sheet specification in mind? (Hint: The answer is “yes.” I go over this topic and how it can save you money in Chapter 7, “Save Time and Money with Web Standards.”)

- **Scripting** — Be explicit about how your professionals should use JavaScript. I personally believe that every function must work when JavaScript is turned
off. This keeps a broader range of users happy, including those who lack access to JavaScript-capable browsers.

- **Servers and databases** — Make sure that you are explicit about the server on which the site will run, and the middleware and database applications that are supported.

- **Bandwidth requirements** — Take the time to determine whether your audience is primarily using low- or high-speed Internet connections. If most are on some kind of broadband connection such as DSL or cable, you will not have to worry about speed optimization as much as you would if most users were on dial-up connections. On intranets, everyone might be on the same high-speed network. With public sites, as of this writing, at least half of your visitors are likely to be at dial-up speeds of 56K and less. Web Site Optimization's free online bandwidth report keeps track of public connection speeds at work and at home (http://www.websiteoptimization.com/bw/).

After conducting a survey based on this checklist, it's time to sit down and write your technical requirements document. Keep it short, sweet, and easy to read. Listing the technical requirements formally will protect you from paying for development that you can't use. The specifications discussed in the following chapters are not comprehensive; they are simply examples that are intended to get you thinking about these cost-sensitive issues.

**Good Planning Pays**

Now that you have your project goals, functional requirements, and technical requirements in place, you are ready to begin production. Your life will be much easier and your work much more efficient because you have taken the time to plan. You will have to make adjustments as you go, but with each shift you'll create good documentation. This will help you ask for more resources if you need them and will help you better allocate the resources that you do have.

This chapter is only the tip of the iceberg; there is much more to learn and discuss about site planning and the web production process. I have given you a few tools that will help you move into production, but if you have more time for study, I highly recommend Web ReDesign: Workflow that Works (New Riders Publishing, 2001), by Kelly Goto and Emily Cotler.