Chapter 10: MAINTAIN THE WEB SITE

What is maintenance and why should I do it?

Maintaining the site

Once you have created your site you need to keep it running. Building the site is like giving birth to a child. Maintaining the site is like raising the child and giving the support necessary to help her grow and develop for years to come. Site maintenance includes identifying the resources you need, training people at your site to modify existing documents and graphics and to add new ones, troubleshooting when there are problems, and conducting ongoing evaluation of your site. You’ll need a Webmaster (web site mother/father) to care for and feed your site.

Why does maintenance need to be done?

If you don’t properly maintain your site after it’s built, it will fail in the future. Computers break down, new models are introduced, browser software changes, HTML standards evolve. Most importantly, your organization (school, business, university) will change as time passes. You will need to modify information in existing documents, add new documents, delete dated material. At some point you will have to consider a major redesign of your site to better match the needs and goals of your changing organization. All of this work is part of maintenance. If you don’t do it, your Web site will “break down” and no longer be useful or usable.

What should already be done?

Needs analysis

You began the design of your Web site by conducting a needs analysis. You determined who the your users and stakeholders are for your site. You identified their needs and wants.

Paper prototyping

Next you created a rapid holistic prototype of your Web site on paper. The content of the prototype 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 want to read or to do. 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 fixed major problems in your design, and then conducted further usability tests until no further major problems emerged.
Computer prototyping

You then 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 in the initial computer prototype, you did need to consider user navigation issues, to deal directly with the interactive nature of your site, and to wrestle with limitations of HTML, computer browsers, computer displays, etc. You conducted two or more rounds of usability tests of computer prototypes. Again, you tested the site with people who are representative of your target population. And you tested until you were confident that you had identified and fixed major design problems with the computer prototype.

Building the site

You built your Web site on the computer. In the production phase your team created the actual HTML files, graphics, and other files. You named your site and optimized its performance to minimize response time as perceived by the user. You made templates and other standard parts of your site to optimize production paths and to facilitate future maintenance. You did extensive bug testing to make sure everything was working right and looking good on numerous computer platforms and browsers. You documented your site to facilitate further maintenance. And after your site went on-line, you celebrated and informed others of your new site.

What’s next?

Maintenance. You need to keep your Web site going.

Maintaining the site: How to do it?

Web maintenance is like child rearing -- only your Web server will never leave home after it grows up!

You will need to identify resources for maintenance, including information providers. You may need to train people in your organization to do routine maintenance. And somebody will need to perform routine tasks to keep the site running, troubleshoot problems, support your information providers, and to conduct on-going evaluation of your Web site.

Identify resources for maintenance

Ideally, you will set up the support system for the care and feeding of your Web site during the design and production phase. In practice, you may be so focused on getting your site on-line and working well you may have forgotten that the job doesn’t end there. If there is much useful information available at your Web site, it will be changing over time, and someone will have to update files with the new information. Your site should be checked periodically to make sure that every element is working. In this boom time when browsers are being upgraded every few months, somebody needs to recheck the site with the new Web browsers to ensure that pages still show up correctly, or at least acceptably. Web servers go down, people send e-mail regarding
the site and expect to get answers, stakeholders ask for revisions or new Web pages, people within the organization change jobs and their names need to be replaced in the site. The list of tasks goes on and on.

**Key stakeholders who don’t know what it will take**

The stakeholders who provide the resources to support your Web site are often people who don’t know what it will take. The Web is new enough that many of us who are actually developing and maintaining sites are still learning what it takes. As part of your design team’s effort take pains to educate decision-makers on the need for, and the nature of, ongoing support. Even if you don’t know everything that your organization will need to provide, you can still alert those key decision makers.

None of us knows exactly what long term Web site support will entail. Your Web site baby is brand new and the Web itself is barely an adolescent. For now, keep key stakeholders informed about the specific needs you have now and any changes you foresee in those needs.

**What can you do to help maintenance efforts?**

**Generate enthusiasm for the project as you go along**

People are more willing to pay for an exciting project and projects are exciting when we have ownership in them. It’s important to get the stakeholders involved as “owners” of the design early in the process, and to keep them involved.

When the design phase is over, you will likely need support from many parts of your organization to keep your Web site viable. Individuals all over the organization who feel ownership in the site will make sure that updating and maintenance are carried out for their parts of it.

**Involvement during design.**

When you did your initial needs analysis, you contacted stakeholders to find out how the site could help meet their needs. When you developed your paper prototype, you included stakeholders in the usability tests (after all, they are some of the users of your site). Likewise, when you did your computer prototyping, stakeholders were involved.

You can also update stakeholders during planning meetings. Invite them to attend. Give them printed progress reports so they can see how the design is coming. As you get into the production phase, get feedback from your stakeholders on the graphics and the look and feel of your site. You want them to be proud of it, not be complaining about it, so listen to what they say.

**Involvement during production.**

During bug testing, ask your stakeholders for comments about problems they might be experiencing as they take your site for a “test drive.”

And be sure to get the go-ahead on the final production version from your organization’s head administrator. Avoid last minute requests for changes, especially requests for changes that violate your usability findings, by stakeholders who have not been part of the process by keeping your stakeholders “in the loop.”

---

Our Dean can brag to other university administrators that our site received 40,000 hits per day this past month, and that 9 out of every 10 are coming from outside the university. That equates to over a 1.25 million hits per month.

**Box 1.** The Dean of the School of Education at Indiana University can be proud of our Web site.

---

**Define the skills it takes to do different parts of the maintenance work**

In your organization, there are different kinds of jobs that require different kinds of knowledge and skills. Not everyone is a technical specialist. Not everyone is boss. Not everyone vacuums the floors or empties the trash. All of these roles are important to your organization and the same is true for maintaining the Web. You don’t need a bunch of computer scientists to do all the work, although you may want a programmer on your maintenance team if you need to have PERL and Java programs written well.

**Updating documents**

Maintaining and updating existing HTML documents does not require a very large skill set. Most people who know how to do word processing can learn the basics of HTML in a half-day workshop. One of the best ways of keeping information up-to-date is to make the folks who care the most responsible for making those changes. This decentralizes the overall Web publishing effort and, while it may lead to some short term gaps in information, is ultimately likely to be the most efficient way to go.

Web development tools have been improving over the last few years, especially tools with WYSIWYG editors (e.g., Netscape’s Composer and Microsoft’s FrontPage). With these editors your Web maintenance staff can update your site without having to delve into the HTML coding that is automatically created. These editors are easy to use and in many ways very similar to modern word processors. However, we have found that Web sites and documents may need to be customized in ways that WYSIWYG Web editors cannot adequately handle. The only solution is to tweak the HTML codes to fix the problem in these kinds of situations.
System administration

On the other hand system administration of a Web server does require special skills. You want someone who knows how to do Unix system administration: who can issue new user accounts, reset passwords, backup everyone’s disk files to tape, reboot the system if it goes down, and seek out – and understand – new technical information if it is required for proper maintenance of the server.

Even if your Web server is running an operating system different than Unix, it will need administrative attention on a regular basis. You will need someone who has the knowledge and skills to work with the operating system, Web server configuration, and associated software.

Visual design

Not everyone is good at doing graphics and visual design. If you do not have such persons in your organization, you may want to hire or contract with Web artists.

Computer programming

Not everyone is good at computer programming. If your site requires processing of Web forms, and you do not have a facility already available through your server, you’ll want someone who knows how to write those CGI programs that will process the data correctly and reliably.

Summary

What your organization needs may differ from what we’ve discussed thus far, but you should identify key roles that need to be filled for maintenance of your Web site. Getting people qualified to fill those roles is part of the support you will be educating your stakeholders to provide.

Argue for a centralized Webmaster

However well-meaning and competent the efforts of distributed information providers, your Web site will fall into disrepair if no single person looks out for the design as a whole. New design issues will inevitably arise which were not addressed in the original plan. The central person -- the Webmaster -- may not decide these issues but should be able to call together interested parties to make the required decisions.

Every site has levels and elements which cut across the entire organization (searching, indexing, guidelines and resource pages). These levels are critical for the site and yet don’t belong to anyone unless there is a Webmaster looking out for them.

A Web site without a Webmaster is like an orphaned child. With no parent to look after it, it could very well be short-lived.

If necessary, train people to do routine maintenance

When identifying the various roles and skill sets needed to maintain your Web site, it may be to your advantage to train some people in your organization for certain kinds of tasks they were not doing before. Normally, one of the biggest tasks is to keep information
During the development of the site for IUB in the summer 1995, I learned a great deal about the campus that I didn’t know before -- and I’ve been here for 30 years -- just by testing links from our top-level information structure. Going back 2 months later I was amazed to see how existing sites had changed, and how many more new Web sites were added.

Box 2. Learning about the site by checking it.

A decade ago relatively few people knew how to do word processing and regularly did electronic mail. Nowadays, people who do e-mail and word processing are the norm, not the exception. Creation and modification of Web documents are on par with these skills. Empowering people in your organization to learn Web skills may facilitate your organization’s goals.

Just don’t underestimate the resources it may take to keep a Web site checked on a regular basis and train your extended maintenance team. Our Ed Tech Services unit is now getting more questions about Web development than anything else. They recently trained all their technology staff to answer routine questions from faculty and staff. You will need to be prepared to support the people who are checking the site and performing document updates.

What is included in routine technical maintenance?

Technical maintenance of your site includes keeping the site running, troubleshooting problems, supporting information providers, and conducting on-going evaluation.
Keep the site running

Leasing.
If you lease your Web site, the service provider keeps the site running. You normally want your site to be available 24 hours a day, 365 days a year, with minimal down time. You want your files regularly backed up (you don’t want to lose documents if hardware fails, or someone inadvertently deletes files). Most importantly, you want folks who request files from your site to see them in their Web browsers. If your users get too many “server busy” messages too often -- because your Web site is running on a server which cannot keep up with the overall demand -- or they have to wait too long for files to be sent across the Internet, then you could be losing potential business, turning away new customers, failing to recruit potential students, or eroding user confidence in the dependability of your Web site.

Owning your Web site.
If your organization owns and runs its own Web site, then you should have staff who can meet routine. Your site should be available 24 hours a day. When you are sleeping at night, folks half way around the world may want to access your site during their daytime waking hours. If you have invested in good computer equipment, placed it in a controlled, secure and well-protected space with its own emergency backup power, use proven operating systems which can handle large volumes of time-sharing tasks (such as Unix) and proven server software. If your server is set up properly it will pretty much run itself. Still, you’ll need someone to do routine system administration including regular backups of the computer hard drive(s), monitoring the processes running on the system to make sure you haven’t run out of disk space or developed some other problem, and processing log files or e-mail complaints to the Webmaster.

Troubleshoot problems

In the fall of 1997 our current Web server, which had been running continuously without problems for about 2 years, began crashing. We were getting about 30,000 hits a day. We had also instituted an asynchronous Web conferencing system earlier in the summer. In early September, we found ourselves restarting our Web server 2-3 times a day, following unexplained exhaustion of system resources.

Examine server error logs.
Examine error logs generated by the Web server program. Making such a routine inspection, for example, I noticed that many “could not fork process” errors occurred in the logs prior to the crashes. After we rebooted the server, these errors would disappear for a while, and then within the next 12 hours would begin to reappear again with increasing frequency until the next crash. These data led me to conclude that our Unix operating system was maxing out its resources.

This led me to examine the Web server configuration files. I began to modify some parameters having to do with the number of child processes being started by the server to handle many simultaneous user Web requests, and how long the processes would be permitted to continue. After making some changes, I would monitor the error logs
to see if the number of “could not fork” errors was reduced or eliminated. This seemed to help for a while, but then – like dandelions in the spring – those errors kept springing up again after a while.

It turned out that we needed to tweak a Unix system configuration parameter in order to allow our Web server enough latitude in generating processes to handle Web requests. On hindsight, the Web conferencing system was running a lot of CGI scripts which caused the problem to become more severe. The problem had existed, as it turns out, for some time. For unexplained reasons we had needed to restart our Web server about once every month or two. Now that we have tuned our server and operating system, I am happy to report that our server has not crashed for the past 6 months.

Re-test external links.

Another class of problems is the validity of links to organizations outside of your Web site. Routinely check such links to see 1) if they are still working and 2) whether the external site is still providing the content that it did when you added the link to your own site.

You may recall the problems caused when Web sites move to new locations or change their names. Unfortunately, sites do disappear and the external link that was working yesterday may be broken today. You won’t know it is broken unless you check periodically on all these links to external sites. Don’t forget to actually look at the content of the site to make sure it’s still relevant and something you want your site to be pointing to.
Re-test internal links.

Don’t assume that all the internal links within your site are working, just because they were all working last week or last month. Sometimes errors are introduced into HTML documents by text editors which do conversions without the developer knowing about it (such as inserting a blank space in the middle of a once-correct URL). Sometimes developers don’t test their documents thoroughly after editing and reinstalling them on the server. However the problems happen, users don’t usually care – they are simply frustrated if they can’t view the documents they want.

Summary.

Link testing is like immunizing a child or taking her to regular dental check-ups. Do it to prevent problems before they become costly or irreversible. And be prepared for the unexpected. The Web is a dynamic system. Your Web server can be running just fine, and then wham! If your organization and your clientele depend heavily on your Web server, then you must be prepared to deal with emergencies.

Support information providers

The skills required to maintain a Web site are not entirely the same as those required to design it. A passion for detail and a belief that “thoroughness is next to godliness” characterize the ideal maintenance specialist. In addition, look for a person who is motivated improve the site continuously, who asks, “What could we do to make this a little easier or a little faster?” A maintenance person should be able to communicate clearly with the information providers who contribute to the site -- putting out regular reports on the “state of the site,” new resources they should know about, and changes they will have to accommodate.

Don’t overlook the role that information providers play in maintaining a site. In a distributed or decentralized organization bring information providers together on a regular basis to recognize the daily efforts they make to update a site, inform them of any new developments at the site, and find out how what they need in the way of on-going support. If your maintenance strategy focuses only on the central person responsible for the site’s upkeep, you haven’t tapped all the resources that a maintenance effort requires.

Support for “non-tekkies”.

If your information providers are not technical people, make it easy for them to contribute their information to the site in a way that conforms to the standards you’re working hard to maintain. Template files, directories of graphics that can be copied directly from the Web, and “boilerplate” text will all make the lives of low-tech information providers easier. Such helpful aids will also encourage them to participate in responsible care and feeding of the site.

You may need to set up a process whereby information providers contribute their material by e-mail or through a Web page form. Then a technically skilled member of the maintenance team can publish the appropriately marked-up HTML document in the correct directory on
your site. If you have sophisticated technical support yourself, forms-based submissions from information providers can be automatically turned into HTML documents by a program created specifically for that purpose.
**Figure 43.** An example of a “standard graphics” page at Indiana University.
Figure 44. An example of a Web form for submission of information by information providers at the Web site.
Figure 45. An example of guidelines for Web developers at Indiana University.
Support for “tekkies”.

If your information providers are technically adept and your organization is composed of semi-autonomous or highly differentiated units, you may have a different maintenance problem: How do you accommodate the diversity of your information providers while still preserving consistency of presentation and navigation throughout your Web site? Keep your mandatory guidelines simple and few. Focus on the guidelines that affect the usability of pages and the ones that protect integrity of the upper levels of the site (see the IUB guidelines at URL http://www.iuinfo.indiana.edu/policy/).

Conduct on-going evaluation of the Web site

Dealing with browser software and computer platforms as they evolve

One of the frustrating things about software development in the “old days” was that just about the time you finished a project, a new computer model was introduced or the next version of some operating system came online. If you were lucky, your software still worked on the new platform, but it already looked dated in comparison to programs using newer technology like higher resolution monitors, more simultaneous colors, windows, and icons. If you were unlucky, your software didn’t work on the new equipment or operating system, and you had to revise it -- sometimes spending as much time for revisions as for the original development.

When the World Wide Web came along, there was some hope for interoperability. The greatest feature of the Web was and is (for now) that it should not matter what kind of computer a person is using, or what operating system it is running. Documents should still be rendered correctly and have a similar appearance, while maintaining the structures that HTML provides. The great hope is that what we produce today for the Web will still work 5 years, 50 years, or 500 years from now.

The introduction of the free graphical browser (Mosaic) at the University of Illinois, fueled the exponential growth of the Web in the early 1990s. Now we see commercial software developers like Microsoft attempting to push Web standards in directions that could give them a competitive business advantage. The good news is that browsing programs like Netscape Navigator and Internet Explorer are getting better and better. The bad news is that there are non-standard HTML tags or embedded scripts that will work on some browsers, but not on others. There is a conflict between improving the HTML standards and achieving business proprietorship.

What this all means for you is that, as Web browsing software evolves, you need checking out all the documents and links at your site to make sure they still look reasonable and work without problems no matter what browsers your users are using.

If you have been conservative in your approach to the development of Web documents at your site, then you should be fairly safe that you won’t need to make major changes in HTML document markups at
your site as time passes. On the other hand, if you’ve decided to use some of the latest features supported only by certain browsers, you run the risk that such features might not be supported as HTML standards in the long run. If that happens, you might have to mount a major overhaul of your entire Web site. If you do choose to use browser-dependent features, other Web browsers may not be programmed to interpret these features and will ignore them. Users will be confused by the resultant rendering -- which could be literally a mess.

Most users -- unless they are Web-knowledgeable and have high-end equipment to support themselves -- do not download the latest version of the hottest browsers all that often. For example, our School of Education receives a significant number of hits from Netscape 2.x, even though it is a dated browser. In March 1997, two-thirds of our education Web site visitors were using “older” browser versions (Netscape 2.x and 3.x). Less than one-third were using Netscape Communicator 4 or Internet Explorer 4.

As with design and production, to make sure that your pages are still working with newer Web browsers: test often.

Dealing with HTML and other standards as they evolve

HTML standards are evolving, though not as fast as some private companies who develop browser software would like. In 1998 HTML version 4 is the standard which Web browsing software programs are expected to support. XML is on the horizon and is envisioned as the solution to extending HTML indefinitely. XML will likely result in major changes in how we technically design our Web sites. While the shift may be arduous, it is likely to make maintenance much easier in the long run.

Holding your own against “feature creep”.

Be realistic about the resources you will have to maintain a site after the original design team has packed up and gone home. If your original design relied on graphical menus and other fancy features, you will need someone who can recreate or edit those features on new pages.

Some Web designers believe their sites must incorporate the latest and greatest new features. For some organizations that could be essential (e.g., companies that produce multimedia products), because they want their target audience to see how capable they in implementing those particular features.

We believe that it is more important to pay attention to your usability data. If your users have been saying to you that they want information from your organization quickly and efficiently, then you should be trying to achieve that goal rather than adding bells and whistles just because they are now available. If a new feature of HTML does enhance the goal of efficient retrieval, by all means consider it. But don’t hesitate to rely on your usability test results to show that people can find what they need at your site without a lot of fancy extras. Anything that appears in your site will need long term maintenance, and people with the skills to perform that maintenance.
Dealing with user comments and further revisions

Make it easy for users to send you comments or ask questions. Provide an HTML “mailto” which allows them to send e-mail to someone at your Web site, assuming that their browser can do so. Make sure the email address appears on the screen so that users who can’t use the mail function directly still know where to contact you. Another way to solicit comments is to provide a link on every page leading to a Web form where users can simply type their comments and questions.

In addition to including a “mailto” on regular Web pages, there are two other places on our Web site where we have found it valuable to allow user comments and feedback. When a browser attempts to retrieve a Web document that does not exist on the server, it is helpful to customize the server’s standard error message. “Error 404 -- File not Found” can frustrate users. In Figure xx, we have replaced this rather cryptic default error message with a customized error message (see http://education.indiana.edu/errors/notfound.html). In addition to being more informative, users are provided with links to the Web site table of contents, to our search engine, and they are invited to submit feedback to the Webmaster.

[*** insert new figures here from the SoE File Not Found custom page, and the feedback form for unsuccessful searches ***]

We have also found it helpful to include a feedback link on pages generated by our Web server’s search engine. We invite the user to tell us what they were looking for, how they had been searching, and what were the results. See Figures xx (a search results page with no found links) and xx (the feedback form: http://education.indiana.edu/errors/searchfb.html). Such comments can provide insights into what users are unable to find at our Web site. This can help us to better index our site and to make other changes in the content of Web pages to make them more “findable” by the search engine.

Somebody in your organization, possibly the Webmaster, should read these comments and questions regularly. The comments users end you can contain valuable clues about problems with your site that you are not aware of.

Check the logs for hits at your Web site

Another routine maintenance check is of records for pages at your Web site that are being accessed a lot. Look for trends over a period of time (e.g., six months). Don’t be too concerned about day-to-day variation.
Ted wondered why our hit count was down about 3,000 hits per day on our education server on August 7, 1996, until he watched the evening news on TV. Turns out America On-Line was off-line for nearly that entire day, shutting out 6 million subscribers from their usual access to the Internet. He’s also noticed that hit counts drop during the holidays, such as around Christmas and New Years.

In the spring of 1998 Ted noticed a sharp increase in hits to pages from our Center for Adolescent Studies – especially dealing with teenage mental health and violence. This appeared to be correlated with media coverage of several shootings at public schools that received national attention.

**Box 3.** Hit counts can fluctuate daily.

If you do see significant, long-term trends of decreasing access at your Web site, you may want to investigate. If there are no obvious explanations (e.g., file protections set incorrectly by accident or oversight, so nobody can access those files on the Web), talk to representatives of the target population and to your stakeholders. It may be time to re-evaluate your Web site. You may want to conduct a further needs analysis or do some further market research.

Ted was invited to make a Web presentation to a group of employees from a local newspaper. They had heard about the process we used to re-design the IU Bloomington Web site in the fall of 1995. During the discussion that followed, he learned that the newspaper was concerned about a decrease in subscriptions in the 18 - 25 year old segment of the population. Healthy subscription rates encourage advertisers to buy space in their paper and folks to place classified ads. Failing to reach this younger generation, the newspaper publisher is evaluating alternatives such as a monthly magazine that targets these youth specifically -- and you guessed it -- a Web site for the “newspaper”.

**Box 4.** A local newspaper re-evaluates its target audience.

**Summary**

If you’ve been employing principles of user-centered design in the development or redesign of your Web site, then the idea of conducting on-going evaluation should feel pretty natural. You’ve seen the benefits of such an approach, so why stop paying attention to your users now?
Building and Maintenance: What to do next?

As we write this book, the Web is only about six years old. It has been an exhilarating roller coaster ride so far. Those of us involved closely with Web development feel we’re part of something really special -- something the world has never experienced before.

The printing press and, more recently, television have brought about widespread change on an international level. We’re more aware now than ever of what’s going on in other parts of the globe. We’re experiencing a transformation in communication that is unrivaled in the history of civilization.

Alan Kay, one of the developers of the graphical computer interface at Xerox PARC in 1970s that was later adopted by Apple Computer and others, says: “The best way to predict the future is to make it.”

Join us on the Web. Let’s make the future.