

# The Ultimate Testing Checklist » SitePoint

By Shirley Kaiser | July 26, 2006 | Server Side Essentials

**Testing plays a critical role in the development of your web site and its long-term maintenance. While smaller web sites – especially those with more limited budgets – may not need to follow the formal testing procedures that are required for large-scale, commercial web sites, every site needs to be thoroughly tested to ensure that it's error-free, user-friendly, accessible, and standards compliant.**

Testing should be completed during each phase of a site's development. Two of the most costly web project mistakes are to delay testing until just before launch, or not to test at all. Testing during production makes it easier to locate and resolve errors, and minimizes the chance of existing bugs being replicated throughout later stages of development. Early and continued testing can eliminate the need for the costly redesigns and other major fixes that can result from overlooked errors.

This chapter's checklists, extracted from SitePoint's new release, *Deliver First Class Web Sites: 101 Essential Checklists*, will help you test your site both during development, and after. Download this checklist, along with others covering SEO and content management — you'll also receive .pdf versions of the documents for immediate use in your web projects.

## Getting Started

### ***Document your baseline web site testing requirements.***

Make use of the preliminary data you collected in Chapter 2 to help determine and document your baseline site testing requirements. For instance, the information you collected on the browsers and connection speeds that your visitors will use, and those visitors' skills and age groups, can be used as a basis for your testing plan.

### ***Source and install all necessary tools.***

You can conduct tests very cheaply using free and low-cost testing tools. Most browsers are free or offer conditionally free versions, and you can download and install multiple browsers – including several versions of Internet Explorer for Windows – on one computer. Alternative device simulators and emulators, such as those for cell phones and PDAs, can also be downloaded and used for free, and most shareware and commercial software creators allow a 30-day trial of their products for testing purposes. Some web design-related discussion lists, as well as the SitePoint Forums, welcome testing requests and posts that solicit bug reports and feedback.

### ***Provide acceptable testing protocols.***

Ultimately, the design and layout for each page of the site must be both aesthetically pleasing and functional across multiple platforms and browsers.

One of the difficulties associated with testing design elements on different platforms is that the test results can be subjective. For example, a designer may create a CSS-driven site that works perfectly on newer browsers, but doesn't look that hot in older browsers such as Netscape 4. The tester may interpret this as a failure point rather than acceptable "design degradation."

To avoid the time and effort involved in re-submitting pages for testing, spell out acceptable cross-browser, cross-platform differences in advance – including protocols for supporting old browsers.

#### *Use Annotated Screenshots to Determine Display Protocols*

*At the end of the final prototyping stage, ask the designer to provide a series of screenshots that have been annotated to identify the differences in the ways the site displays on different browsers and platforms. These screenshots can be used in later testing to identify the displays that are deemed acceptable on each platform. Testers can then compare the protocols and screenshots with their findings.*

#### **Set up a staging server.**

A staging or test server that's identical to your web site's live server environment should be used for testing and development purposes prior to your site's launch.

#### *Automated Programs for Larger Sites*

*Larger web sites typically require more robust testing procedures that test large sets of records, values, and content. You might consider using an automated program, such as Badboy, eValid, or TestComplete for such purposes.*

### **Good Testing Practice**

#### ***Systematically test individual pages.***

Thoroughly test your site one page at a time, making sure each page is solid before proceeding. Carry out a comprehensive test of the final version of the site in the same way.

#### ***Track bugs and confirm fixes.***

Document the details of discovered bugs and their resolutions.

#### ***Regression test, especially when fixing bugs.***

Regression testing, which is also known as verification testing, is the process of retesting pages or sections of a site to make sure that a recent bug fix hasn't broken some other aspect of the site, or reinstated bugs that were fixed previously. Conducting and documenting regression tests will allow you to identify any parts of the site that break, and document the causes of those errors and how you resolved them. If a bug resurfaces at some point, it will be spotted immediately and resolved quickly. (See [Regression Testing](#),

Webopedia Computer Dictionary).

### ***Validate the markup for each individual web page.***

Validate the markup for all your web pages to ensure that each uses structural, compliant markup. This will allow you to confirm that you've used the correct markup for the specified DOCTYPE, fix any typos you may have made, and resolve any syntax errors. Validating your markup as you build a page, rather than waiting until you've created an entire page (or worse, an entire web site), makes it easier to isolate, find, and correct markup problems.

#### *Using Markup Validators*

*Some HTML editors integrate validation tools such as W3C's HTML and CSS validators, the WDG HTML Validator, and the CSE HTML Validator (an HTML/XHTML syntax checker and validator), which allow you to validate your markup conveniently while you work.*

*The CSE and WDG HTML Validators also include helpful batch processing features. So, for example, although I validate each page with the CSE HTML Validator and the W3C's validators as I work, I use CSE HTML Validator's batch processing to recheck multiple pages simultaneously. I also use batch processing as a final check when I've finished developing all the pages of the site.*

*Ensure that the validation tool you use is based on W3C Recommendations.*

### ***Validate all CSS.***

Validate all the CSS you use on the site, including each external style sheet, plus all embedded and inline CSS.

### ***Conduct load testing to stress-test programming technologies and server hardware capacities.***

You need to make sure that the programming technologies you've used, and the capacities of your server hardware, can cater to higher levels of traffic than you expect the site to receive. Taking this approach will help you ensure that your site remains online and fully functional at all times. Load testing procedures should include a trial run – tests that usually involve the use of automated scripts that emulate multiple simultaneous user sessions. The trial run will reveal how the programming technologies and server hardware you've used will cope with traffic spikes.

#### *Matching Hosting with Projected Server Loads*

*Your ISP should match your hosting package with your projected server loads if you're using a web hosting service, so you should let them know what your loads will be. Your web host might recommend a different web hosting package that offers greater capacity if your load testing reveals that your existing setup has the potential for problems as traffic levels increase.*

*If you're managing your own server(s), check whether your server software provider offers free load testing software – many do. You might also consider using dedicated load-*

*testing software products, such as those offered by WebPerformance and NetMechanic. In addition, numerous open-source performance analysis software tools are available to help you assess, monitor, and manage your web site's performance. (See also Stress Tools to Test your Web Server, Microsoft.com.)*

## **General Testing**

### ***Test your web site on multiple browsers and platforms.***

The fact that your web site looks good and works well in one browser doesn't mean it will look as good or function as well on other browsers and/or platforms. Even if you're developing an intranet web site for an organization whose employees use the same browser and operating system, the hardware and software used by those employees can and will vary. In the long run, it's better to develop your web site on the basis of the W3C recommendations, then to follow up with extensive cross-browser, cross-platform testing. Stay informed of the release of new or updated browsers and operating systems so that you can keep your cross-browser, cross-platform tests current.

#### *Which Browsers and Platforms to Test?*

*Check your server logs – as well as other on- and offline resources – that cover browser statistics, operating systems, and emerging Web use trends, so that you know which browsers, versions, and operating systems your web site visitors use now, and are likely to use in future. As part of your approach to site maintenance in the long term, plan to check your server logs weekly (monthly checks might be fine if you run a smaller web site with low traffic volumes) to follow changes and trends in your browser statistics. Take this data into account as you hone your cross-browser, cross-platform testing procedures, to ensure that you're testing all of the browsers and operating systems that your visitors use. Here's a list of the popular operating systems and browsers on which you might test your site.*

#### **Operating systems**

- *Macintosh OS X*
- *Macintosh OS 9*
- *Windows XP SP1 and SP2*
- *Windows 2000*
- *Windows 98*
- *Linux*

#### **Browsers for Macintosh OS X**

- *Safari 1.2*
- *Mozilla 1.6*

- *Firefox 1.0*
- *Opera 9*
- *Opera 8*
- *Opera 7*
- *Internet Explorer 5.2*

### ***Browsers for Macintosh OS 9***

- *iCab*
- *Internet Explorer 5*

### ***Browsers for Windows XP***

- *Opera 9*
- *Opera 8*
- *Opera 7*
- *Mozilla 1.7*
- *Firefox 1.0*
- *Netscape 7.1*
- *Internet*
- *Explorer 6.0*
- *Lynx browser*

### ***Browsers for Windows 2000***

- *Opera 9*
- *Opera 8*
- *Opera 7*
- *Mozilla 1.7.3*
- *Firefox 1.0*
- *Netscape 7.1*
- *Netscape 7.0*
- *Netscape 6.2*
- *Netscape 4.78*

- *Internet Explorer 6*
- *Internet Explorer 5.5*
- *Internet Explorer 5.0*
- *Lynx browser*

### ***Browsers for Windows 98***

- *Internet Explorer 4.0*
- *Lynx browser*

### ***Browsers for Linux***

- *Konqueror 3.0.5*
- *Mozilla 1.6*
- *Opera 8*
- *Opera 7*
- *Emacs/W3*
- *Netscape 7*
- *Netscape 4.8*

### ***Test page optimization with every update.***

Optimizing your pages right from the start can help ensure that your site design and images support fast page load times, and that the site's development is smooth and efficient. Whenever you add anything to your web site – new content, new images, or new web pages – check that content's optimization to help keep the site streamlined. If you're not yet familiar with web site optimization, see Chapter 11, *Web Site Optimization*.

#### *Document Weight and Load Time Tools*

*Some HTML editors include features that tell you a document's weight, or a page's load time at various connection speeds, and most image editing software products indicate file size load times at various connection speeds.*

*There are also free online tools that calculate document weight, composition, and page load times, and even offer recommendations for optimizing web documents. Siteoptimization.com's helpful Web Page Analyzer is one such tool.*

### ***View pages on a variety of displays.***

You may find that your beautiful design, which is wonderfully contrasted on an LCD monitor, is impossible to read on a CRT monitor, and completely unusable on alternative devices. Different displays will not always interpret your site's design and colors consistently, so it's

worth testing your initial design, as well as the site's ongoing development, in a variety of displays.

### ***View pages on different screen resolutions and with various color settings.***

Some HTML editors provide a feature that allows you to check your pages at various screen resolutions and with a range of color settings. Similar tools can also be downloaded as plug-ins for some browsers. For example, the highly recommended Web Developer Toolbar extension for Firefox includes a customizable window resizing tool, among many other useful features. (For more on Firefox's tools for web developers, see Chris Pederick's article, [Web Developer Extensions for FireFox and Mozilla.](#))

#### *Consider Nonstandard Resolutions*

*Screen display sizes may or may not reflect the actual dimensions of the viewable web page. Many users don't expand browser windows to fill the entire screen (especially on larger monitors), and various toolbars and side panels can take up portions of the screen width and height. For example, on an 800x600 display, a browser that has a side panel and a couple of custom toolbars across its top would display only 500x400 pixels of a web page – less if the browser window isn't maximized. Be sure to conduct testing that accounts for this wide range of variability – not just the standard maximum screen resolutions. Table 14.1, which shows screen resolutions of various devices, and Table 14.2, which shows the color depth settings of different displays, should help your testing.*

*Table 14.1. Typical screen resolutions and displays (in pixels)*

[a] Lisa Gade,  
palmOne Treo 650  
Palm OS Smartphone,  
Mobile Tech Review  
(December 10, 2004).

Computer displays	640x480, 800x600, 1024x760, 1280x1024, 1440 x 900 (iMac G5 17" display), 1680 x 1050 (iMac 20" G5 display)
WebTV/MSN TV	544 x 383
PDA, Pocket PC with Windows Mobile, smartphones	240x160: Palm; 320x240: Palm, Pocket PC with Windows Mobile pre-2003, v2; 640x480: Windows Mobile pre-2003, v2; 320x320: Palm OS smartphone <sup>a</sup>

*Table 14.2. Typical display color depth settings*

[a] Browser Statistics:  
What Is the Trend in  
Browser Usage,  
Operating Systems  
and Screen  
Resolution?

Computer displays	Newer displays: 24-bit, 32-bit, 65,000 colors, 16,777,216 colors; Old displays: 16-bit, 256 colors, 65,000 colors <sup>a</sup>
PDA, Pocket PC with Windows Mobile, smartphones	Pocket PC: 16-bit, 65,000 color screens; Old PDAs: 8-bit, grayscale <sup>b</sup>

W3Schools (January 2005).

[b] What is a Pocket PC (PPC)? What models are out there? Mobile Tech Review (no date).

In addition to the displays mentioned in Table 14.1 and Table 14.2, there are standalone tools – such as BrowserCam, an online subscription service – which provide screenshots depicting the way your content will appear in a wide range of screen resolutions and

browsers.

### ***Check for adequate color contrast.***

A good way to check color contrast is to change your browser display to grayscale, or print the web page in black and white. A tool that can be very helpful for checking your design's color contrasts is [Vischeck](#), which simulates colorblind vision, allowing you to see your pages as a colorblind user might.

### ***Test the functionality of external and embedded scripts and functions.***

Some scripts and functions might work fine in one browser, though they may not work correctly, if at all, in other browsers or devices. Minor tweaking can resolve many cross-browser, cross-platform functionality issues. If older browsers or alternative and adaptive technologies don't support some functions, you may wish to provide alternative methods to allow those users to access those capabilities some other way.

### ***Test all links, including navigation.***

Ensure that all links work properly, and point to the correct location. Link checking software can check for valid, broken, and redirected links. Some HTML editors, such as [Macromedia Dreamweaver](#) and [HomeSite](#), offer link checking capabilities. Free tools are also available – try [Xenu's LinkSleuth](#) and [W3C Link Checker](#). See [WebsiteTips.com's](#) section, "Link Checkers, Maintenance" for more. Note, though, that link checkers aren't intuitive enough to check whether links direct to the correct location – be sure to check for this manually.

### ***Check error pages.***

Intentionally enter incorrect URLs into the browser's address bar to check that the appropriate error pages (404 errors, etc.) are in place.

Make sure that the error pages include helpful information and links that help users find what they seek. Keep in mind that, sometimes, visitors will incorrectly guess a domain name or slightly misspell a URL. Help those visitors by accepting common misspellings, typos, incorrect case sensitivity, different terminology for the same items, and predictable domain name errors: provide suggestions to redirect users to the proper page. (Matthew Linderman with Jason Fried, *Defensive Design for the Web: How to Improve Error Messages, Help, Forms, and Other Crisis Points* (Indianapolis: New Riders, 2004), 99, 224.)

### ***Test all downloads.***

Check to ensure that all your download links point to the correct files, and that the download files exist.

### ***Test the search feature***

Ensure that the search feature functions correctly, and that search results are accurate and helpful.

## **Solid Security**

***Clear your Cache!***

*Be sure to clear the browser cache, including cookies, before each test.*

***Check that digital certificates and SSL URLs work correctly.***

Your SSL URLs should begin with `https://`. Also, when a visitor accesses a web page protected by SSL, most browsers will show a "lock" icon at the bottom of the browser, like that being shown by IE 6 for Windows in Figure 14.1.

*Figure 14.1. The "lock" icon displaying at the bottom of the browser window*

SSL is commonly used to encrypt online transactions and other sensitive data, as well as intranets and extranets.

Verisign and Thawte are two popular companies that issue SSL certificates.

***Check that all pages requiring SSL access are accessible only via SSL.******Test the security of restricted areas.***

Users might share or try to guess the URLs of content and downloads in protected areas of your site. Restricted content URLs might appear within publicly available referrer logs. People might even share usernames and passwords without authorization or permission. As such, it's critical that you anticipate the kinds of security breaches that might take place, and test methods for their prevention.

***Hack your Site: the Ultimate Security Test***

*Download popular hacking tools to use in testing, to see if your protected areas can easily be comprised. Those managing larger-scale web sites might even consider outsourcing or hiring a hacking expert for testing purposes.*

*Ensuring security and data integrity – especially in terms of your confidential data, including customer credit card information – is critical for promoting and maintaining trust among your site's visitors. Don't assume that your web site's security is always okay. Test regularly to make sure it remains secure.*

***Test forms and form controls.***

Check to ensure that forms are submitted correctly, and that they're only submitted when the correct information is entered and required fields have been completed. Review form error messages to ensure that they are helpful and informative within the context of the form itself.

***Test online shopping facilities.***

If your web site includes a shopping cart or similar functionality, thoroughly test back-end operations to ensure that all transactions are secure, and everything runs smoothly.

## Accessibility Testing

If you plan to make your site accessible, and include accessibility testing in every phase of your project, chances are good that by the time your site is ready to launch, it will meet the W3C's WCAG Guidelines, as well as any other accessibility guidelines that it must address.

### Conducting a Preliminary Review

The W3C recommends a two-phase process for testing a site's accessibility. First, developers should conduct a preliminary review of the site that includes testing a few pages to get an idea of the site's accessibility. Then, they should undertake a thorough, comprehensive review that includes an evaluation of every web page (or representative pages on larger web sites) with a variety of tools and users who have disabilities. (W3C, *Evaluating Web Sites for Accessibility*, W3C Web Accessibility Initiative (November 14, 2002).) The checklist below is a guide for conducting a preliminary accessibility review.

#### ***Select a random sampling of pages to test.***

You should test every page of a small web site. For larger web sites, select a random sampling of pages that includes your homepage, at least one or two pages from each section, and especially important or popular pages, such as the contact page and the sitemap.

#### ***Use a graphical browser to test sample pages.***

#### ***Use a voice or text-only browser to test sample pages.***

A text-only browser, such as the free Lynx text-only browser, can provide valuable insight into your web site's accessibility.

#### ***Use two evaluation tools.***

Table 14.1 and Table 14.2 list tools that you could use to evaluate your site.

#### ***Manually examine representative pages using the checkpoints from WCAG 1.0, your government accessibility guidelines, or other requirements.***

#### ***Summarize the results.***

If your results summary indicates a number of accessibility issues, you might wish to make site-wide changes before you conduct a comprehensive review. Conduct another preliminary test after you implement these changes.

### Conducting a Comprehensive Review

A comprehensive review is much more thorough and detailed than a preliminary review. Here's a checklist that will help you ensure you don't miss a thing!

#### ***Identify, determine, and document site-wide conformance requirements and levels.***

You may include WCAG 1.0 Level 1, local government requirements, and company guidelines in your documentation, for example.

***Test the site's accessibility conformance.***

Test every single page of smaller web sites. For larger web sites, test a representative sample that includes the homepage and all especially important or popular pages, pages from each section, pages for each different layout, and pages generated from databases.

***Use at least two different accessibility evaluation tools.***

Manually examine representative pages using the checkpoints from WCAG 1.0, your government accessibility guidelines, or other recommendations.

***Use a graphical browser to test sample pages.***

***Use a text-only browser and a voice browser to test pages.***

***Ensure that interface elements are operable with multiple input devices.***

For example, test all interface elements with a mouse, with keyboard only, and with adaptive technologies. If you use JavaScript to create navigation rollover effects, ensure that links work without JavaScript, provide HTML text links, or implement both.

***Proofread all content.***

Read over each page of the site to check that all the text is clear, simple, and appropriate for your web site's target audience, as well as those segments of the market that have cognitive or learning disabilities.

Make sure that your text is clear, simple, and written for the Web, then boost its accessibility by allowing for those with disabilities.

*How Complex is your Writing?*

*Clear Language and Design (CLAD) offer a great Reading Effectiveness Tool that's designed to assess the literacy skills that users will require in order to understand your text.*

*For helpful insight on cognitive and learning disabilities, see Roger Hudson, Russ Weakley and Peter Firminger's article, [An Accessibility Frontier: Cognitive Disabilities and Learning Difficulties](#).*

***Conduct usability testing with a diversity of participants.***

If possible, conduct usability testing with people who have a variety of disabilities, technical expertise, and varying levels of familiarity with your web site. Use a variety of adaptive technologies, if possible.

There's no better way to assess your site's accessibility than to have disabled users test the site. It should be noted, for example, that a sighted person will not use a screen reader, such

as JAWS, in the same way as a blind person who uses the tool daily.

## Testing with a Graphical Browser

You can do a fair amount of accessibility testing with your browser. Opera is especially conducive to accessibility testing, since the browser makes it quite simple to turn off images, animations, style sheets, JavaScript, frames, and so on. Ian Lloyd has written a great tutorial about using Opera for testing a site's accessibility.

There are also helpful accessibility testing extensions, sidebar panels, and toolbars for other graphical browsers, including Mozilla, Firefox, and Internet Explorer. As well as visiting your favorite browser's web site, see Derek Featherstone's [Testing Tools for Developing Accessible Web Sites](#) – a helpful list of annotated browser links.

### ***View pages without images.***

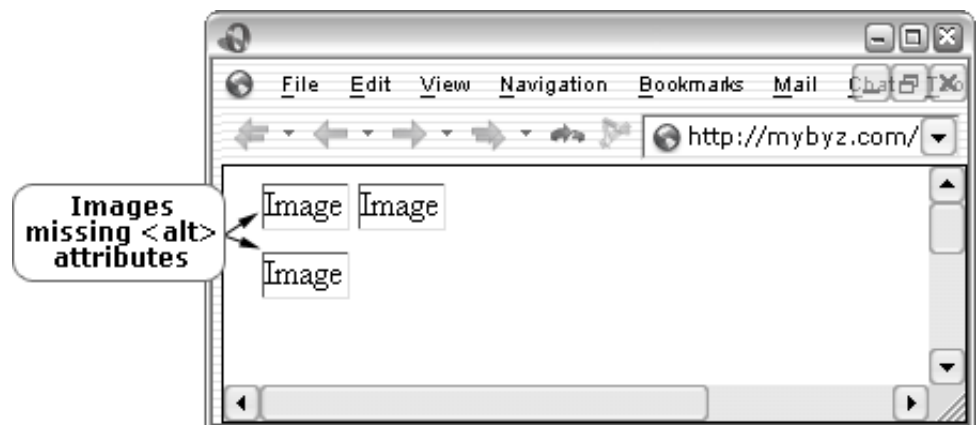
As long as you've included the `<alt>` attribute for each image (for use by browsers that don't support images, or visitors who are browsing with images turned off), the browser will hide decorative images that have blank `<alt>` attributes. However, if you don't include `<alt>` attributes, browsers will not hide the images, and screen readers will read the word "Image," as indicated in Figure 14.2.

Figure 14.2. Viewing images with blank `<alt>` attributes in Opera 7 for Windows

### ***View pages without style sheets.***

***Check that your pages allow users to vary the font sizes displayed.***

***View tabular data with tables disabled or in a browser that doesn't offer table support.***



This process will indicate how usable your tabular data might be for visitors using screen readers, text-only browsers such as Lynx, and older PDAs.

### ***View the web site with new technologies turned off.***

These "new technologies" include JavaScript, Java, sound capabilities, and plugins. Your page should remain functional and accessible when these technologies are unavailable.

## Testing with a Voice or Text-only Browser

If you don't have handy access to a voice browser, you can use Lynx, a free text-only browser, or Delorie's online Lynx Viewer, for preliminary testing. For comprehensive testing, the W3C recommends testing with a text-only browser and a voice browser. (W3C, *Evaluating Web Sites for Accessibility*, Web Accessibility Initiative, W3C (November 14, 2002).)

### ***Check that web site navigation is available and works properly.***

If your web site uses images – including image maps – for navigation purposes, text hyperlinks should appear in their places (via `<alt>` attribute text, and possibly also in the `<title>` attribute text, if needed). If you use Flash or Java applets for navigation, your site should include alternative text navigation links to help ensure accessibility.

#### *Helping Everyone get Around*

*Keep in mind that this approach also helps search engine bots to crawl and index your web site. For more on optimizing your web site for search engine crawlers, see Chapter 11, *Web Site Optimization*.*

### ***Check all hyperlinks.***

### ***Check that decorative images don't appear in text-only browsers and aren't read by voice browsers.***

As we discussed above, all decorative images, such as bullets and design-related images, should have blank `<alt>` attributes so that text-only browsers and voice browsers ignore them.

### ***Check that the information is presented in a way that's comprehensible when read serially.***

Lynx allows you to simulate serial access, so that you can make sure the information you've presented is understandable to users of voice and text-only browsers. (Joe Clark, *Building Accessible Websites* (Indianapolis: New Riders Publishing, 2003), 236.)

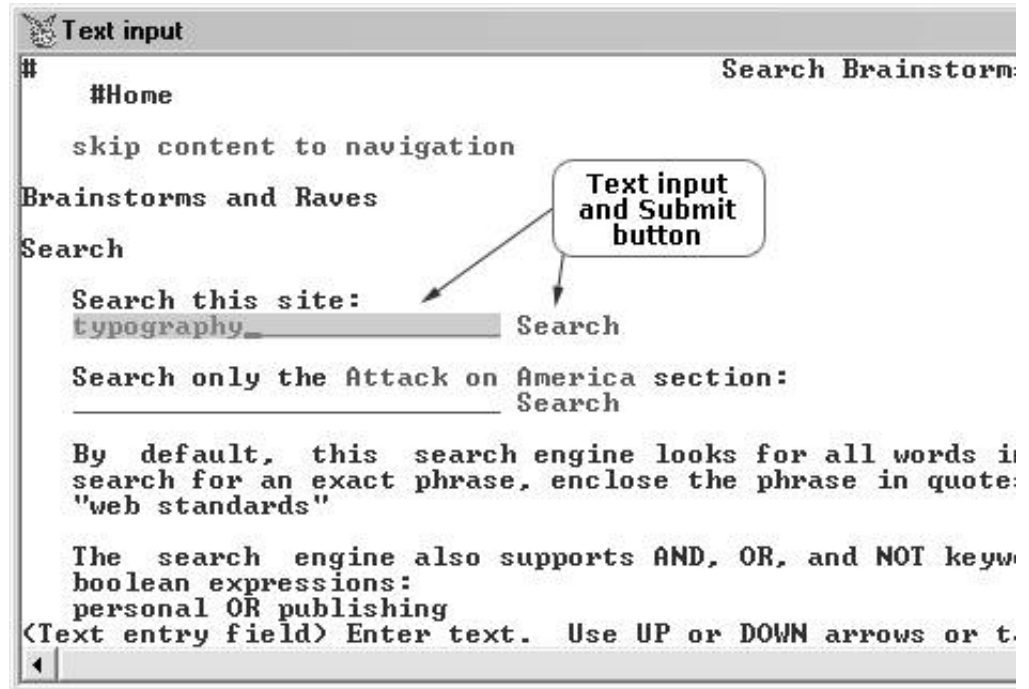
### ***Test form controls.***

Figure 14.3 depicts the testing of an accessibility-friendly form with Lynx 2.85.

*Figure 14.3. Testing a form in Lynx*

## Usability Testing

If you find yourself short on money, time, or both, you might be tempted to consider skipping the usability testing phase. Rather than skipping this process altogether, it's far better to test on an abbreviated, informal scale. It's better to conduct a number of small, fast tests during development, rather than one big test at the end – especially if that final usability test is to be scheduled just before the site's launch.



The main goal of usability testing is to make sure your final-version, live web site is user-friendly and works well for your visitors. Three main dimensions of usability testing are effectiveness (testing to see if the user can accomplish desired tasks), efficiency (testing to see how much effort is involved in accomplishing the desired tasks), and satisfaction (testing to see if the user has a satisfactory experience and will return). (Dey Alexander, Quick Wins with Usability Testing (presentation, Web Workshop Series at Monash University, Melbourne, 2002), slide 3.) Use the following checklist as a guide to determining and conducting your usability tests.

### **List test objectives and concerns.**

Be specific, making note of tasks and areas of the web site that might be of concern – for instance, forms that might be difficult to complete or use, or the methods by which users are expected to find certain products. Be sure to specify these age groups and computer skill levels that you wish to test.

To help determine your objectives and concerns, check with your clients, management, designers, developers, and others involved in the project.

### **Choose test approaches.**

Numerous approaches have been developed to testing various aspects of sites' usability. In planning your usability tests, it's important that you research and consider the available approaches in light of the resources and time you have available.

For example, you may choose to use personas or card sorting to help test your web site's

information architecture – one single, but critical aspect of your site's usability. By creating personas, you can document the steps typical visitors would take to accomplish particular objectives, then test those steps – looking specifically at the information architecture, navigation, and forms, for example, that that persona might use to achieve that goal.

Card sorting can be a helpful way to understand how your web site visitors will likely understand and explore the information on your site.

### *Conducting a Card Sorting Test*

*To conduct a card sorting test, give users index cards on which are written the titles of pieces of content, features, or categories. Include brief descriptions on each card, but don't hint at any information structure or classification. If you plan to add more content to your web site in future phases, it can help to add those specific elements to the cards if you can.*

*Be sure to brief your test users, explaining that there are no right or wrong answers. Ask them to read each card carefully and group them into piles that make sense to them. If the users aren't sure about a card, they can set it aside and come back to it whenever they wish. When they finish, ask them to label each pile.*

*Compare the results with your web site's information architecture as one way to determine whether or not further tweaks or changes are required.*

### **Recruit users for testing based on your target market.**

Test users should represent your target audience profiles. For example, if your typical users aren't very technology-savvy, but you test with technology-savvy users, your test results won't uncover potential technology-related problems, such as confusion with drop-down menus or multiple browser windows.

A test group of five users is considered large enough to provide information that will help improve your site's usability.

### **Prepare an NDA (non-disclosure agreement) ready for users to sign prior to starting your tests.**

### **Determine testing location(s).**

Jakob Nielsen and other usability experts talk about conducting testing in usability labs. Indeed, labs can be helpful when you're testing a highly sensitive and secure web site, you have significant hardware components or specific hardware that must be used for testing, you anticipate long testing sessions (more than a couple of hours or so), and you have a project for which you'd like a team of observers to watch the tests. (Julia Houck-Whitaker, *Remote Testing versus Lab Testing*, Bolt | Peters, Inc., (February 3, 2005).)

Alternatively, there can be advantages to "field" testing – testing the site in an office or home setting, using testers' own computers at these locations. This approach eliminates the costs of a usability lab while allowing you to observe how your web site will work for typical visitors. In addition, testers may be more comfortable and relaxed using their own

computers. However, Jakob Nielsen reports that field studies tend to be expensive, and it can be challenging to get permission to conduct them at company locations. (Elizabeth Neal, *Why You Don't Need a Usability Lab*, SitePoint (October 8, 2004).)

Consider also the possibility of conducting remote testing. For example, you could be at your office and use the telephone to verbally communicate instructions, obtain user feedback, and engage in discussion. Unless you have online video cameras set up strategically for testers' computers, though, you'll miss the important observation part of your testing. You'll need to encourage them to "think out loud," and remember to phrase your responses and questions to remain as neutral as possible. Remote testing can be quite effective for large web sites, those with ecommerce capabilities, web applications, and intranets. (Houck-Whitaker, Velda Bartek, and Deane Cheatham, *Experience Remote Usability Testing, Part 1*, IBM developerWorks (January 1, 2003).)

### *Videotape your Tests*

*Videotaping tests can be helpful, as the videos can facilitate later review, allowing you to see details that you may not have noticed during testing, and to show the tests to others on the project team. Time-stamping the videos will make it easier to reference key findings later.*

### *Top Testing References*

*For more on testing approaches, see Janice Fraser's *The Culture of Usability: How to Spend Less and Get More from Your Usability-Testing Program*, and Lane Becker's *90% of All Usability Testing is Useless*.*

### ***Create a relaxed atmosphere for testers.***

Rather than making people think they're in a lab being assessed, decorate your test area as if it were a typical office setting or perhaps a living room. The SitePoint article, *Why You Don't Need a Usability Lab*, has more on this.

### ***Establish realistic tasks for users to perform and identify what, specifically, you will measure.***

For example, to test a web site that offered numerous articles categorized by topic, you could set tasks that require users to find specific articles or topics. From these tests you could collect data on the time the test subjects needed for each task, whether or not the navigation was helpful, whether the local search was used, and if the results were accurate and helpful.

### ***Provide simple, clear instructions only.***

For accurate test results, do not provide any hints.

### ***Allow users to perform their own tasks in addition to your pre-defined tasks.***

Users will often come up with tasks you haven't thought of, and by pursuing these tasks, can uncover important subtleties and problems with a site. (Carolyn Snyder, *Paper Prototyping: The Fast and Easy Way to Design and Refine User Interfaces* (Amsterdam: Morgan

Kaufmann Publishers, 2003), 330.)

### ***Create a post-test questionnaire.***

Consider what, in particular, you'd like feedback on, and create a questionnaire that addresses those issues. Also, include areas for comments and suggestions. You may choose to make a questionnaire with a 1–5 ranking system (1 being poor and 5 being excellent), and that asks users to rate the following:

- their overall satisfaction with the web site
- their overall satisfaction with the design and layout
- the ease with which specific tasks were completed
- the ease with which specific information was found
- whether navigation labels, page titles, headings, and other terms were easy to understand and accurate
- whether icons and other images were helpful and easy to understand
- whether downloadable items downloaded quickly and satisfactorily
- whether password-protected areas were easy to log into and use
- whether errors occurred, and, if they did, whether appropriate and helpful error messages were encountered
- whether product information was sufficient, or there was too little or too much information
- the ease with which they completed ecommerce transactions
- whether or not the user would return to the web site
- whether or not the user would recommend the web site to others
- suggested improvements

Using a ranking system with areas for comments gives you valuable data. You can analyze the ranked responses, and use the comments and explanations for further helpful insight.

### ***Analyze test results.***

You may decide that it is necessary to meet with your entire web site project team following each testing session to discuss what worked, what didn't, and what the possible solutions may be.

Concentrate first on high-level functionality issues, such as global navigation, links, and page layout; then focus on specific areas and recommendations. (Kelly Goto and Emily Cotler, *Web ReDesign 2.0: Workflow that Works* (Indianapolis: New Riders / Berkeley: Peachpit Press, 2005), 230.)

You may also wish to create a report on the results and recommendations to present to the designers and developers and, if applicable, to clients.

## Summary

This chapter emphasized the importance of checking your work and testing the site throughout every phase of its development and long-term maintenance. There were checklists to help you test your designs, content, and markup; procedures for cross-browser, cross-platform testing; and discussions on testing labels, functionality, security, accessibility, and usability.

You can download this chapter of *Deliver First Class Web Sites: 101 Essential Checklists*, along with all the SEO checklists in .pdf format, ready for use with your next project. To find out more about the book, check out the *Table of Contents*.

## Written By:



## Shirley Kaiser

Shirley Kaiser owns SKDesigns, a company focused on creating visually appealing web sites that are based on web standards and universal accessibility, and are optimized for speed and search engines. Aside from her web work, Shirley is a professional musician with Bachelor of Arts (B.A.) and Master of Arts (M.A.) degrees in Piano Performance.

## Related Posts