

No #FAIL Accessibility Testing

Angela M. Hooker

Office of Citizen Services and Innovative Technologies

US General Services Administration

Now hear this!



- Given our time limitations, I can't show every type of test, but I'll try to mention each step needed for testing.
- *Compliance*, in some respects, is subjective—determine if your project can be *used*; but do use a checklist to provide steps. **Test, Test, TEST.**

Wait a minute ...



- Unless you're evaluating an established product, if you're being asked to address and test accessibility at the end of a project, you're going to pull out your hair!
- Plan to test accessibility during the development of your project.
- It's easier to test and remediate in small increments than waiting until everything is built and later discovering that the infrastructure is weak.

Rules for your testing mission



- Be **thorough**
- Be **ruthless**
- **Protect yourself** from negative repercussions **and your agency** from lawsuits
- Remember: It's all about your **users' needs**—not your agency's needs

Testing Case Study

Note: No disrespect is intended to any tool or site mentioned or tested today. We're all about education.

Validate the code

- Validate all code (HTML, CSS, scripts, frameworks, etc.):
 - Sometimes problems that occur when using assistive technologies are due to coding errors.
 - Don't worry if your site uses a technology approved by the W3C (such as [ARIA](#)), even though the [W3C Validator](#) counts the use (not the *form*) of this technology as invalid.
 - Just because it doesn't validate doesn't mean it's inaccessible. Just because it validates doesn't mean it's accessible.

Review the code

- After validating, *read* the code:
 - We expect automated tools to pick up poorly formed or misused code, but not all do.
 - Make sure tags are used properly—*semantically*—so that users know what to expect: For instance, don't use `<blockquote>` for formatting, since screen reader users will expect that the text within that tag is a quote.
 - Check that the code is in the proper order.
 - If the code isn't right, it doesn't pay to proceed with other testing until the code is corrected.

Automated testing

- Use an automated tool: It can find issues quickly that would be time consuming to locate during a code review.
- The tool you use will depend on your constraints:
 - Are you testing templates? A sample page? A large site? Is your site behind a firewall? Will it check all standards you require?
 - WebAIM WAVE toolbar for Firefox or WAVE online
 - Deque Worldspace (tests single pages [online, local files on your PC, or code you cut and paste into the tool], plus Flash implementations and PDFs—up to WCAG 2.0 level AA [not up to AAA])
 - Deque FireEyes
 - iCITA Functional Accessibility Evaluator—very (overly?) strict

Color contrast

- Hopefully, you've already worked with the project designer to suggest an accessible color scheme, and tested for any issues.
- Don't forget to test the contrast of text on top of background images (generally, automated tools won't pick up those instances).

[Juicy Studio Color Contrast Analyser](#), [WebAIM Contrast Checker](#), [Accessibility Color Wheel](#)

Skip link(s)

- Check for a method to skip repeated navigation:
 - Is the skip link always visible or is it visible when it gains focus?
 - Some people think that using headings (<h1> ... <h6>) is sufficient, but that doesn't help users with motor issues, since they're unable to navigate by headings (unless they're using Opera).
 - Are there *too many* skip links?
 - Does the skip link work?
 - If not, is it because of the IE bug (see Jim Thatcher's article on skip links; scroll down to "Sometimes Skip Links Don't Work")?

Headings

- Are the `<h1>` ... `<h6>` headings used properly?
 - Are they in proper order (except, perhaps, the `<h1>` tag on pages other than the home page)?
 - Are there too many `<h1>` tags?
 - Are heading numbers skipped? For example, are there only `<h1>` and `<h5>` tags?

Styles

- Disable the cascading style sheets (CSS):
 - Is the page readable, clear, and usable?
 - Does the page flow/read in the order intended?
- Are there inline styles (they can interfere with user defined styles)?

Images

- Is there alternative text for inline images?
- Turn off all images:
 - Do complex images have alternative text on the page, or is there a link to the content?
 - Are decorative images coded in the CSS, or if not, do they have null alt attributes (alt=“”)?
 - With images disabled, is there sufficient contrast between the text and the background?

Information by color

- Is there information conveyed by color?
 - Check for keywords (“red text is required”);
 - Print the page in black and white; or,
 - Use the Web Developers’ Toolbar to disable colors
- If there is info conveyed by color, is there an alternate method to provide that information?

JavaScript, AJAX, other scripting

- During the project planning, discuss if JavaScript is required to use your site or if you'll provide content by an alternate means when users have JavaScript disabled.

NOTE: According to Search Engine People, it's estimated that, besides bots, 2 percent of people in the United States have JavaScript disabled; however, that number of people can fill New York city two times, or the entire state of Wisconsin; be sure to check keyboard with AJAX.

- Be sure scripts function with assistive technologies (AT)—watch out for “forbidden” event handlers.
- Turn off scripting—is the page functional?

Screen flashes

- Does the page contain anything that causes the screen to flash more than three times per second?
- Are there any graphics that are optical illusions (think of an Escher-inspired graphic) that would appear to be flashing?
- This is of great importance: Don't cause your users to have seizures!

Multimedia, sound, video ...

- Does any multimedia content have synchronized captions?
- Is it fully navigable by keyboard (with clear on-screen focus)?
- Are audio description tracks needed?
- Do audio files have transcripts (ideal for video, too)?

Tables

- Are there tables used for layout?
 - If so, do they have summary attributes (remove them)?
 - Do they (rightfully) use role="presentation"?
 - Is the content properly linearized?
- Are data tables properly marked up?

Frames/iframes

- Do frames have proper markup (frame titles, use of the proper doctype)?
- Has content within frames also been provided by using `<noframes>`?
- Do iframes have title attributes with descriptive information? Is scrolling set to auto? Is the content within the iframes also provided through links nested within the `<iframe>` `</iframe>` tags?
- When navigating by keyboard or assistive technology, do you get trapped inside a frame or iframe (usually because of scripting)?

Forms

- Do all forms have text associated with each form element by using the `<label>` tag?
- Do text labels appear next to their form element?
- Are `<fieldset>`s used properly with legends?
- Has scripting been used to automatically change the page location, using a form element (for example, a drop down box that loads another page when a user clicks on an item in the box)?

Plug-ins and software

- Are there links to plug-ins or software that are necessary for reviewing the content on the page?
- Is the plug-in or software accessible?

Screen resolutions

- Check the page at different screen resolutions:
 - Is there horizontal scrolling?
 - Is any content cut off?

Mouse-free/keyboard navigation

- Use the keyboard alone to navigate through the page completely (tab through):
 - Are you able to navigate through all items?
 - Is the tab order logical?
 - Do you get trapped inside any blocks of content?

Screen magnification/zoom

- Use Opera to zoom (up to 1000%) the page
 - Does the layout contract and expand to minimize horizontal scrolling?
 - Is there excessive horizontal scrolling?
 - Is the page still usable and readable?
 - Are graphics too pixelated to be readable?

Link text

- Does each link make sense when isolated out of context?
- Are there multiple links with the same link titles that go to different destinations?
- Is the link destination clear from the link text?

Content

- Review all content:
 - Does it use plain language principles?
- If content isn't your specialty, get a plain language specialist to review your content—be sure they understand the potential issues faced by people with cognitive disabilities, low literacy, low language proficiency, dyslexia, and autism/Asperger's syndrome.

Cross-browser/platform/device

- Test in several browsers, on different operating systems, and on a few platforms. For example:
 - PC:
 - XP, Windows 7, Vista, Linux
 - IE7 – IE9; Firefox 3.x and up; Chrome; Opera 10 – 11; Safari; Konqueror
 - Mac:
 - Tiger, Leopard, Snow Leopard
 - Firefox 3.x and up; Safari; Opera 10 – 11; Chrome

Mobile

- How does the page appear on mobile devices?
- Is the code optimized for mobile?
- Test on real devices (c'mon—someone in your network has these devices!).
- Review your site in Opera Mini.
- Check your site in the W3C's mobileOK tool.
- View the mobile development resources from the W3C.

Assistive technologies (AT)

- Your site may be coded perfectly and follow all best practices, but if it doesn't work with AT, how will you know? (Believe or not, it happens!)
 - Test with users—sit with a person who uses AT and test your project, so you can see how and if they're able to use your content.
 - Give accessibility a face! Reach out to users and understand why they need a well-developed site.
- Test with voice recognition software, a screen magnifier, and at least one screen reader.

Why Many Accessibility Testing Plans Fail

There's No *Single* Silver Bullet

The notion that testing with a single tool to check the accessibility or compliance of an entire site or software application

=

FAIL

“No tool exists that you can run against your web site (or web page for that matter) in order to assert that it is accessible and/or complies with the Section 508 provisions or the Web Content Accessibility Guidelines, however much you are willing to pay.

“When a web site claims Section 508 Conformance or WCAG Conformance from some tool or other (and many do it), the most it can mean is that the site (or page) passed all of the *automatic* Section 508 or WCAG tests.”

— **Jim Thatcher**, Web Accessibility Testing

More reasons for testing #FAIL

For government websites and web applications, people generally focus on *one* portion of the Section 508 standard (usually 1194.22: Web-Based Internet and Intranet Information and Applications) when other parts of the standard often apply to their project.

More reasons for testing #FAIL

Testers often use JAWS, or another assistive technology—with no other test—to determine accessibility.

See WebAIM's article [Testing with Screen Readers](#), and Clear Helper's post [Stop Using JAWS for Web Accessibility Testing?](#)

Understand your users' needs

- Learn about each disability type and its needs for accessibility:
 - Cognitive
 - Auditory
 - Motor
 - Seizure and Neurological
 - Visual
 - Aging

See Dive into Accessibility's Tips by Disability, WebAIM's Considering the Users' Perspective: A Summary of Design Issues, and W3C-WAI's Web Accessibility and Older People.

Tips and Resources

Again, test with real users

- You'll see your project from their perspective, and learn if their needs are being met.
- Test with people who have different disabilities.
- Use social networking to recruit participants.
- Ask your section 508 coordinator if there are any willing participants within your agency.
- See Dev.Opera's article on [recruiting users for testing](#).

Protect yourself and your agency

- Document all of your testing.
- Create testing logs (contact me for examples).
- Maintain documentation in your project archive.
- If you have a contractor perform testing, make sure they document their testing (put this in the project contract/statement of work) and are able to give you raw data as well as detailed reports.

Just because ...

- ... another agency uses a technology doesn't mean the implementation is accessible.
- ... a vendor says their product is accessible or 508 compliant doesn't mean it is—conduct due diligence testing.
- ... your page/site/application functions properly in one browser doesn't mean it will work in all browsers, other platforms, devices, or assistive technologies.

Tools

- ColorZilla
- Firebug
- FireEyes
- HTML Validator
- Juicy Studio Quality Assurance Tools
- Juicy Studio Web Accessibility Toolbar (Firefox)
- Paciello Group Web Accessibility Toolbar (IE)
- W3C's HTML and CSS Validators
- Web Developer Toolbar by Chris Pederick
- WebAIM WAVE (online, upload, or toolbar)

Resources and articles

- Aaron Cannon/Northtemple's Accessibility Checklist
- Joe Dolson's Automated Accessibility Testing
- Dennis Lembree's 25 Ways to Make Your Website Accessible
- The Pickards' WCAG 2.0 Lite
- Ginny Redish and Mary Frances Theofanos' Observing Users Who Work with Screen Readers

Resources and articles

- Henny Swan's [Setting Up a Screen Reader Test Environment](#)
- Marco Zehe's [How to Use NVDA and Firefox to Test ...](#)
- UW-Madison's [Screen Magnification and the Web](#)
- WebAIM's [The Importance of Human Evaluation](#)
- WebAIM's [WCAG 2.0 Checklist](#)

Rock the accessibility!



Angela Hooker

angela.hooker@gsa.gov

Twitter: [@AccessForAll](https://twitter.com/AccessForAll)