Accessibility Best Practices

Plan, Develop, and Test Inclusive Design

iSoftStone | MOVING FORWARD TOGETHER

Contents

Inclusive Design Makes Better Experiences. For Everyone.

Accessibility is on your roadmap, but is it part of your daily sprint?

One in four US adults with a disability

Inclusive design makes a better product for everyone, not just those with impairments. There is no reason to bury accessibility to the bottom of your backlog. Yes, it is often a large lift for a developer, but you should be planning, designing, and building for accessibility right now.

Inclusive Design Changes Lives

Technology touches every part of our lives, and that is never going to change. For many people with disabilities, technology is itself one more barrier. There are new, exciting developments in accessibility every week, and in fact, there has never been more opportunity to build inclusively. Knocking these barriers down is life-changing for people with impairments.

- Only one in four adults with disabilities have a job.
- Americans with disabilities earn about two-thirds the median income of those without disabilities.
- Because of accessibility barriers, Americans with disabilities are 3 times as likely to never go online.

Legal Mandates for Accessibility

If you still need a reason to blow the dust off your accessibility strategy, then know that in 2010 the Department of Justice ruled that the Americans with Disabilities Act (ADA) also included access to digital and online content. Since then, many cases have gone to trial over accessibility issues.

In the European Union, the European Accessibility Act has already levied millions in fines against some tech companies.

There is a legal tidal wave building that will require access to apps, sites and platforms for all users.

We've put this guide together to help you get started with accessibility strategy, best practices, and testing.

Note: We reference the Web Content Accessibility Guidelines (WCAG) throughout this document, but WCAG is **not** the law. The laws that govern accessibility are different. In the US, it is the Americans with Disabilities Act. Following WCAG is highly recommended, however, and is the industry standard at the time this material was published.

ACCESSIBILITY QUICKSTART

Simple steps that deliver the biggest impact to users.

Low Effort, High Impact Issues

UI or Actionable Features Unavailable via Keyboard
Some users never touch a mouse when navigating the web. With this in mind, it becomes increasingly apparent that every UI must be navigable and actionable by keyboard-only controls. At a minimum, before releasing a feature or going live with UI, use the tab key to navigate the page or program and confirm that users can reach and activate all aspects of the page via keyboard-only controls.

Relates to WCAG 2.1 standard: 2.1.1 Keyboard

Keyboard Traps
This one is straightforward. In the instance that a user navigates to a component of the page, the user must be able to move focus away from that component using only keyboard commands. As you can imagine, it would be quite frustrating if you had navigated to an element and were stuck there until you closed or refreshed the page. These issues will be found frequently when testing for keyboard focus to see if all UI is accessible via keyboard but may require some additional footwork (shift+tabbing, activating elements that cause flyouts to appear, etc.)

Relates to WCAG 2.1 standard: 2.1.2 No Keyboard Trap

Unknown Elements
Navigating a page and hearing only "Button", "link", "checkbox", etc makes taking action on these elements complete guesswork. Additionally, if form fields do not inform users of their purpose or have associated labels, it can be difficult and sometimes impossible to know what is necessary to complete the task. Define each element programmatically so users can understand what element they've navigated to, what kind of control it is, and understand the purpose.

Relates to WCAG 2.1 standard: 4.1.2 Name, Role, Value and 1.3.1 Info and Relationships

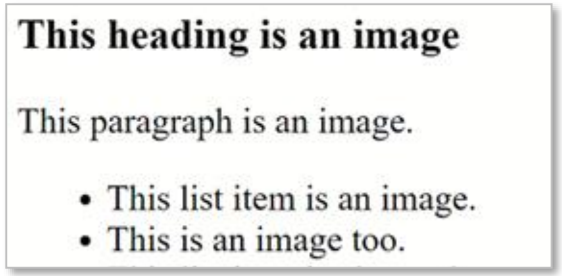
Low Effort, High Impact Issues continued

Lack of Form Validation
After filing out a form, users will look for a "Submit" or "Next" button to complete the task or move on to the next step. If no feedback is given upon submission, or users are not informed of any errors that may have occurred, they will be unaware how to fix the issue or know if the task was completed successfully. Form validation can be tricky but there are many articles and documentation to help you out.

Relates to WCAG 2.1 standard: 3.3.1 Error Identification and 3.3.3 Error Suggestion

Images of Text
When navigating to an image of text, unless the full contents have been added to the image alt text or long description, users will not be aware of what is portrayed in the image. This also effects users that magnify the page or utilize high contrast settings as images can get blurry and won't resize properly upon magnification (also will not respect OS text customization settings) and will not be effected by high contrast settings.

Relates to WCAG 2.1 standard: 1.4.5 Images of Text


This heading is an image

This paragraph is an image.

- This list item is an image.
- This is an image too.

Flashing Content
Thankfully this is not a common issue but has the highest potential for damage and danger to users. Users who are susceptible to seizures may have problems if any element flashes more than 3 times in any one second. It's best to not have flashing content at all, but if it's necessary, make sure it occurs less than 3 times per second.

Relates to WCAG 2.1 standard: 2.3.1 Three Flashes or Below Threshold

Easy-to-Fix Accessibility Issues

Making your website accessible may seem like a daunting task; below are a few of the easiest to fix, most common accessibility issues website creators can make that will greatly increase accessibility. These directly relate to the Web Content Accessibility Guidelines (or WCAG 2.1), so if implemented properly your site will adhere to accessibility standards.

Add appropriate alt text to images
Make sure your alt text properly describes the image. Stay away from subjective adjectives such as "beautiful", "scary", or "cool". Also, do not mention "image" or "picture" in the alt text as screen readers will already inform users that the element is an image.

For images that are purely for decoration purposes, write the alt text as *alt="";* this will hide the content from screen readers which is fine if there is no relevant information that can be gathered from purely decorative elements.

Relates to WCAG 2.1 standard: 1.1.1 Non-text Content

Make sure your links properly describe their purpose
When users navigate to a link, it must be apparent what the purpose of that link is. Coming across something that states "click here" or "more" does not help users determine if they want to select that link or continue with their current page. I recommend making the link purpose clear with the link text alone as it is not much more challenging to accomplish and is much more accessible than the link only making sense in conjunction with the surrounding context.

Relates to WCAG 2.1 standard: 2.4.4 Link Purpose (in context) and 2.4.9 Link Purpose (Link Only)

Easy-to-Fix Accessibility Issues continued

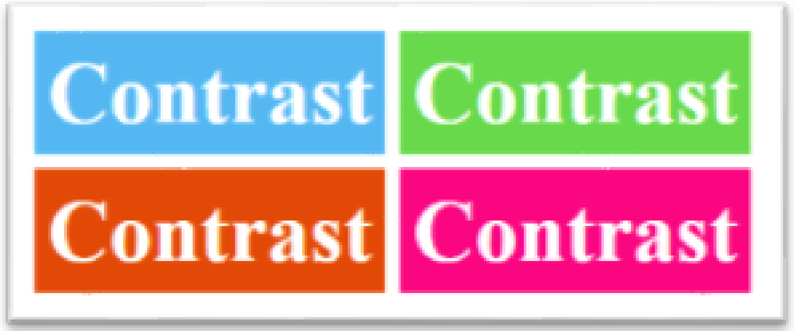
Use proper color contrast
When designing your page, make sure the text you are utilizing has enough contrast. WCAG states that text and images of text have a contrast ratio of at least 4.5:1 while large text (18-point font or 14-point bold font and up) must be at least 3:1.

Some exceptions include incidental content (such as inactive UI or decoration) and logotypes which currently have no restriction. There are also requirements for non-text contrast such as UI components and graphical objects which must have a contrast ratio of at least 3:1.

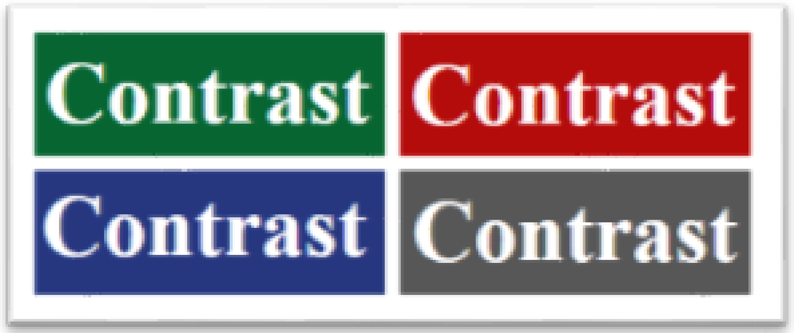
If you're unsure of the contrast of your text, [Colour Contrast Analyser]() is a free, easy to use software to verify the luminosity contrast ratio.

Relates to WCAG 2.1 standard: [1.4.3 Contrast (Minimum)]() and [1.4.11 Non-text Contrast]()

Contrast Failure



Contrast Success



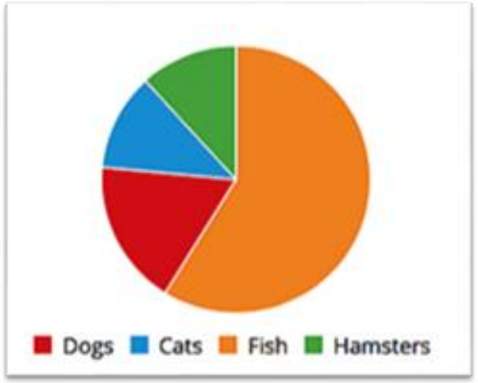
Easy-to-Fix Accessibility Issues continued

Don't use color as the only means to understand something
As visual users and creators, it's very easy to accidentally utilize color to convey meaning but this can limit many individuals from being able to understand the information. Any time you are using color, other methods to understand the purpose need to be implemented. This can be achieved with visible labels, shapes, or textures.

Relates to WCAG 2.1 standard: [1.4.1 Use of Color]

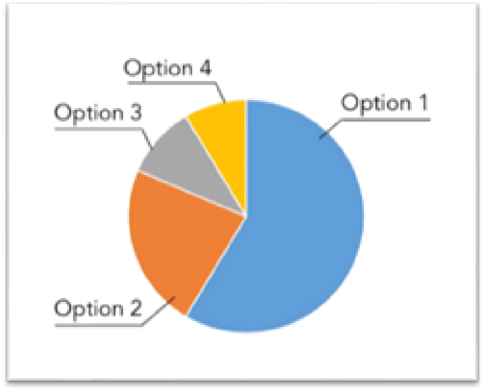
Failure

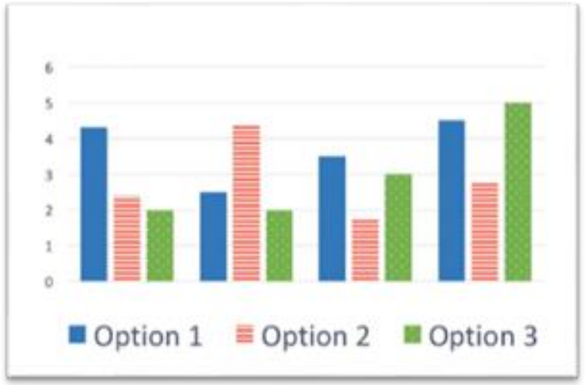
Users with color blindness may not be able to understand what is represented in the graph to the right



Success

Users with color blindness can still understand the important information in the graphs below due to the choice in labels and/or graph structure.





Easy-to-Fix Accessibility Issues continued

Add "skip links" or "skip to main" functionality
Frequently, sites have helpful navigation, search boxes, or toolbars/menus that can provide users access to other locations within the site. While helpful, these links are repetitive and can be quite cumbersome to navigate through every time users access a page. Adding a "skip links" or "skip to main content" feature allows keyboard-only users to quickly bypass this repetitive content on a website to get to the main content of the page.

For source code and further information on skip links, please navigate to the following resources:

- How to implement skip links
- "Skip Navigation" links

Relates to WCAG 2.1 standard: 2.4.1 Bypass Blocks

Make sure your page title is informative and relevant
The page title should properly inform users of what the page purpose is. It's the first thing users will hear when navigating to a new page and can also help them find pages they were previously on if they navigated to another tab or program. Put the most relevant page information first and be as descriptive as possible while still being succinct.

Examples:
<title>Contact Me </title>
<title>Homepage </title>
<title>Videos </title>

Relates to WCAG 2.1 standard: 2.4.2 Page Titled

ACCESSIBILITY BEST PRACTICES

How iSoftStone designs and delivers accessible content.

Designing for Visual Disabilities

People that are blind or have other visual disabilities form the core audience for our accessibility efforts even though they are not the largest audience of users with disabilities.

However, the discipline, design, and testing processes that have sprung from accessibility for people with visual disabilities have had an impact across the industry.

In this section we discuss our best practices for designing for common assistive technology, such as screen readers. These best practices extend to other types of inclusive design that we will explore later.

ARIA Labels

At iSoftStone, we use ARIA to develop for screen readers when native HTML doesn't work, but we only use it when there are no better options. ARIA allows web elements to be tagged so they can interact with screen readers and other assistive technologies.

There is almost always a better native solution than ARIA.

ARIA has, in our opinion, become a band-aid for overworked developers. It has never been a substitute for proper site design and there is an extreme danger with ARIA. If you don't use ARIA correctly, you will create a poor experience for someone using assistive technology.

No ARIA is better than bad ARIA.

Keep these rules in mind when using ARIA

- Use native HTML with element or attribute semantics already built into it.
- Don't change native semantics with ARIA unless it is your last option.
- Interactive ARIA controls must work with only keyboard input.
- Avoid using *role= presentation* or *aria-hidden hidden true* on a visible keyboard-focusable element. This can make the element invisible to assistive technologies, especially in browsers already designed to work with assistive tech.
- All interactive elements must have an accessible name.

Best Practices for Low-Vision and Blindness

Ensure that your HTML passes W3C validation tests. If your code is W3C compliant, it will pass many of the accessibility standards as well.

HTML tags must reflect proper content hierarchy. Keep the hierarchy consistent and simple. Remember, a visually-impaired user has a much higher cognitive load when using your site or app.

Alternative text for images, video content, audio content, and embedded objects are suggested for important elements. Decorative images can be tagged alt="".

Have a consistent, concise heading template in place for all pages.

Design for high contrast. WCAG contrast ratio requirements vary between 3:1and 4:5:1. We often see a 7:1 ratio for web and mobile apps.

Make sure that your site does not depend on JavaScript alone to function.

Understand, implement, and use WAI-ARIA (the Accessible Rich Internet Applications suite) for web development when it makes sense. Remember that no ARIA is better than bad ARIA!

Keep your most critical information in text format. Likewise, keep the text short and to the point.

All important information must be provided on keyboard focus.

Allow users to skip sections of your site, such as links that "go to primary content" or "go to my order."

Surface the most important information to the top to reduce cognitive load. For example, if a product in an ecommerce site is out of stock, make that information that first thing a user realizes when they access the product.

Forms should be designed with accessibility in mind: associate labels with fields.

Designing for Cognitive Disabilities

Designing and developing for cognitive disabilities can be challenging.

First, these disabilities are rarely understood and their impacts range between individuals. It is unreasonable to expect accessibility standards to meet every possible scenario for cognitive disability, but there are design and development elements that can be implemented.

Second, the changes required to be inclusive for cognitive disabilities can be significant. Often, these suggestions may require a re-write of a site or app with attention to information architecture, navigation, and other features. It is typically not as simple as adapting your work for a screen reader.

The good news is, changes made for cognitive inclusivity are changes that improve everyone's experience.

In this section we outline our best practices for building designs that improve the experience for all users.

Best Practices Cognitive Disabilities

Memory Disabilities

Memory disabilities are not uncommon. They can be caused by serious disease such as Alzheimer's, or simply a mild disorder brought on by age. Here are our tips for designing for users with memory issues:

- Focus on navigation – standardize designs, controls, buttons, menus, and placement across the entire site or app.
- Provide ways to easily start an action over without losing information. We like the use of breadcrumbs and reminders.
- Use prompts and feedback to give users clear guardrails for decision making.
- Avoid having users re-enter information. When possible, allow the user's device to access stored information - especially for form fills.
- Provide summaries for complex processes and make a process map easily accessible while the user completes the task.
- Create information chunks and avoid long scrolling pages. Instead, use single pages with one idea per page.
- More clicks aren't a bad thing, if they make sense and are easy to click. Keeping concepts simple and isolated are much better for memory-impaired users.

Learning Disabilities

Learning disabilities can range from attention disorders to issues around reading and comprehension. Below are some best practices around learning disabilities:

- Use plain, simple language.
- Use a mix of media: images, text, and video.
- Do not hide critical information. Avoid extraneous information altogether.
- Use images and icons to help users remember content.
- Keep decisions to a minimum – focus users along a narrow decision path when possible.
- Avoid menus that appear or disappear with mouseover.
- Allow users to control media – do not have auto-playing videos or sound.
- Use a high contrast, uncluttered design to avoid distractions.

Designing for Mobility

Most of our focus is on building designs that work well with screen readers, but largest number of people with disabilities in America is around mobility issues.

Even people without disabilities face mobility issues throughout the day. If you are holding a coffee in one hand and your smartphone in the other, you have a temporary disability – you only have the use of a single hand to navigate the phone.

For users with more severe disabilities there are several assistive technology options. From eye-tracking technology to sip-and-puff assistive devices, we will look at design needs specific to mobility issues.

Best Practices for Mobility Issues

New mobility-assistive technology is released regularly. Stay relevant with:

- Microsoft Eye Control
- Microsoft Gaze Interaction Library

Buttons and clickable icons should be large and prominently displayed.

Follow best practices for content hierarchy and structure that are accessible for screen readers.

Avoid menus and elements that activate only with mouse interactions, such as display on mouseover.

Implement speech recognition throughout your app.

Do not demand precise interactions on your site or app.

Do not use short timeouts.

Limit typing, scrolling, and clicking.

Designing for Auditory Disabilities

Users who are deaf or have other hearing disabilities are often the least impacted when it comes to accessing sites and apps. They often do not require assistive technology to use a site, so designing for auditory disability can be easier and faster.

Best Practices for Auditory Disabilities
Provide transcripts of video and audio media.

Allow users to engage with your site via text. Do not force a phone call or speech-controlled method for interaction.

Do not use sound cues as the only means of interaction, notification, or channel for critical information.

ACCESSIBILITY TESTING

How iSoftStone tests accessible content.

iSoftStone Accessibility Testing Process

iSoftStone follows three basic steps when testing for inclusive design.

Define What Will be Tested
We call these "core scenarios", or the most common user flows in the product, app or website. This includes any "gating" flows such as the user having to log in, sign-up, or purchase.

Build a Testing Matrix
There is a wide variety of test tools and browsers, and not all work together. We build a testing matrix that defines combinations of platform, operating system, and versions for testing. These combinations determine what software can be used for accessibility testing.

Test the Product
We start with automated tools – many of which are reviewed in detail in this section. Then, manual testing is conducted by certified accessibility testers. It is not uncommon for the results of automated testing to relay what is broken but then fails to place the errors in a larger cohesive context.

You can learn more about iSoftStone's accessibility testers here: [7 Things You Need to be a Better Accessibility Tester]()

Our Favorite Testing Tools

Accessibility extensions for Chrome

aXe by Deque Systems. This automated tool makes quick work of accessibility testing, generating a list of errors while on site. Deque Systems also offers more advanced, paid accessibility testing tools that we love. This tool has recently joined forces with Microsoft to release the open source Accessibility Insights.

Chrome Accessibility Developer Tools. Adds an accessibility audit, and an accessibility sidebar pane in the Elements tab, to Chrome Developer Tools.

WAVE Web accessibility evaluation tool. WAVE is a web accessibility evaluation tool developed by WebAIM.org. It provides visual feedback about the accessibility of your web content by injecting icons and indicators into your page.

WCAG Accessibility Audit. Exposes Google Accessibility Developer Tools in an easy-to-use UI.

WCAG Luminosity Contrast Ratio Analyzer. Pick colors from your Web, compute contrast, get suggestions. Preview with different types of impaired vision simulators.

Web Developer extension by Chris Pederick. The Web Developer extension adds a toolbar button to the browser with various web developer tools.

Tenon Check. This extension adds a button to your browser. Click it, and the page you're currently viewing will be tested for Accessibility against WCAG 2.0 using Tenon.io.

Our Favorite Testing Tools continued

Web Based Tools
Gunning Fog and Flesch Reading Ease Test. Reading-ease tests determine how easy copy is to read and understand. The easier copy is to understand, the more accessible it is - especially to users with cognitive impairments. Remember, users who rely on a screen reader face stressful cognitive loads on some pages – easy-to-read copy helps them navigate a site.

Desktop Applications
Colour Contrast Analyzer. This tool allows you to quickly test for color contrast. Correct color contrast ratios are important for many vision-impaired users. The tool provides a pass/fail assessment using the WCAG 2.0 color contrast guidelines. It also simulates certain conditions such as dichromatic color-blindness and cataracts. You can see how your web content appears to people with less than perfect vision.

Photosensitive Epilepsy Analysis Tool (PEAT). PEAT scores animations and videos to see if they are likely to cause seizures.

Total Validator. Total Validator is an HTML validator, an accessibility validator, a spell checker, and a broken links checker all rolled into one tool, allowing one-click validation of your website.

Who We Are

iSoftStone helps global brands build, test, and launch complex projects. Accessibility is a core component of every project we do, and the primary objective of the case studies highlighted below.

If you need a partner to make your site, app, or product conform to the strictest WCAG standards, we will help.

We have both near-shore and off-shore teams, including access to thousands of testers to meet the highest standards and quickest turnarounds.

Our staff in Kirkland, Washington is focused on not just our client's projects, but on developing new tools to help the accessibility community.

Want to know more?
We keep it simple – just contact our client services team and we'll see how iSoftStone can make your next accessibility project a wild success.

John Baron
johnbaron@isoftstone.com
425 216 6300

Microsoft: Sign Language Application Project
We are partnering with Microsoft Research to build a platform to collect American Sign Language (ASL) videos and data to support Microsoft's efforts to improve Accessibility services. We've created the data management, video capture, and backend components for this project.

Microsoft: Internal Accessibility Compliance
When Microsoft needed help bringing millions of internal documents, SharePoint and Team sites, and other internal web and mobile assets to WCAG AA compliance, we were their first choice. Our team of PMs, testers, developers, and designers have delivered fully compliant content that meets or exceeds Microsoft's strict standards.

Come Meet Us

If you are in the Kirkland, Washington area we would love to meet you in person. We have a monthly Meetup group at our corporate offices dedicated to accessibility.

We share best practices, have product demos, interactive activities, and plenty of snacks. Join the community and hear from experts in the accessibility space.