The Do’s and Don’t’s of Accessibility

Michael Larsen
mktesthead@gmail.com

Abstract

Accessibility is a large topic and one that often gets a variety of approaches to deal with. Often it is seen as having to focus on a large checklist (the WCAG standard) and make sure that everything complies. While this is a great goal and focus, often it is overwhelming and frustrating, putting people in the unfortunate role of having to read and understand an entire process before they feel they can be effective.

My goal is to help condense this a little and give some key areas to focus on and be effective in identifying Accessibility issues quickly and helping testers become effective advocates. We will look at ways to find issues, advocate for them and help make strides to greater understanding and focus moving forward. We can use a little to provide a lot of benefits.

Biography

Michael has worked on a broad array of technologies and industries including virtual machine software, capacitance touch devices, video game development, and distributed database and web applications. He currently works with PeopleFluent, located in Raleigh, NC, USA. He writes a software testing blog called TESTHEAD (http://mktesthead.com/).

Michael served as a member of the Board of Directors for the Association for Software Testing from 2011-2015. He was their Treasurer and then their President. Currently, he helps teach their Black Box Software Testing classes. Michael is also the current producer and a regular commentator for The Testing Show, a podcast produced for QualiTest (available in Apple Podcasts, Google Podcasts, and Spotify).

Copyright Michael Larsen August 15, 2021
1. **Introduction**

Accessibility can be defined as “The design of products, devices, services, or environments for people with disabilities. Accessibility allows for design compatibility with a person’s assistive technology”.

Accessibility utilizes the WCAG standard and other regulatory documents as required in various counties to determine that our software complies with suggestions and regulations. The WGAG standard is large and getting larger all the time. It is understandable that people would look at the checklist, feel deflated, and wonder if they would ever be able to understand it all, much less actively use it.

To this end, I want to share some ideas based on key areas and help people get their hands dirty quickly, get some quick wins, and develop skills that will help interested testers and developers approach Accessibility more effectively.

2. **Why Do We Need to Deal with Accessibility?**

If we are lucky enough to live a long and complete life, we are likely to experience a primary or secondary disability of some sort: visual, auditory, motor, cognitive, or a combination of any of those listed.

Nearly 1 in 5 people have some form of disability in the USA. More than a billion people in the world today experience some form of disability. Even those who don’t consider themselves disabled have some sort of disability – even if it is simply wearing glasses. It makes social and economic sense to plan for a better user experience by incorporating design philosophies that address Accessibility.

3. **Situational Disabilities**

Persistent issues (low vision, low hearing, limited mobility, cognitive disability) are considered primary or chronic disabilities. However, there are situations people without chronic disabilities face. We refer to these as "situational disabilities".

Examples of situational disabilities include:

- Background noise (hearing)
- Distracted tasking (cognitive)
- Small text/non-scaled web page (vision)
- Foreign language (literacy)

Have you been in a country where you can’t read the language or even the writing? This is akin to the frustration people with disabilities feel using systems not designed for Accessibility.

One of my favorite examples of Accessibility and Inclusive design is IKEA's assembly instructions for any product. Pictographs and images overcome literacy issues. Also, in most cases, only one set of instructions is needed, rather than repeating the same series of instructions multiple times in several languages. This likewise cuts down on the cost of repetitive printing, which has a bottom line effect as well as a customer experience improvement.
4. Ten Principles of Web/Mobile Accessibility

Jeremy Sydik’s book “Design Accessible Web Sites” focuses on “Ten Principles of Web Accessibility”. Those ten principles are:

1. Avoid making assumptions about the physical, mental, and sensory abilities of your users whenever possible.
2. Your users’ technologies are capable of sending and receiving text. That’s about all you’ll ever be able to assume.
3. Users’ time and technology belong to them, not to us. You should never take control of either without a really good reason.
4. Provide good text alternatives for any non-text content.
5. Use widely available technologies to reach your audience.
6. Use clear language to communicate your message.
7. Make your sites usable, searchable, and navigable.
8. Design your content for semantic meaning and maintain separation between content and presentation.
9. Progressively enhance your basic content by adding extra features. Allow it to degrade gracefully for users who can’t or don’t wish to use them.
10. As you encounter new web technologies, apply these same principles when making them accessible.

These principles are helpful when it comes to designing applications and they are also helpful when it comes to framing how they should be tested.

5. Mobility Issues

Mobility issues are often the most visible of disabilities. A person may have missing or immobile limbs or extremities, or they may have a loss of fine motor control. These conditions can be present from birth, but mobility impairments also are the most common secondary, or situational, disabilities. Injuries to the wrist or hand that require immobilization, such as bone breaks, can significantly affect the way people interact with their systems and how they input information.

A variety of conditions, such as multiple sclerosis, Parkinson’s disease, diabetes, stroke, myelopathy, and arthritis, may make it difficult to control a mouse or a stylus, or otherwise effectively use a standard keyboard.

There are several approaches that can help an application be more accessible to those with mobility disabilities.

Types of mobility disabilities

Primary mobility disabilities include the following:

Paralysis: A variety of conditions can result in a lack of movement in the extremities. Birth defects such as spina bifida and cerebral palsy can have a significant impact on the ability of extremities to operate, with a gradation of minor fine motor control issues to full immobility.
Limb loss or absence: Meromelia is a birth defect where one or more limbs may be lacking a part or only partially formed. Diseases or injuries also can require the amputation of limbs and extremities.

Ensuring your application is usable by those with mobility issues

The goal for truly inclusive design is to engineer your application in such a way that assistive technologies aren’t even always necessary. Here are a few suggestions, many of which do not require assistive technology, that you should check to ensure your application is accessible by those with mobility issues.

- Ensure all functions can be accessed with the keyboard

This is specific to web applications and regular computer systems. Interactions that focus heavily on mouse usage will be a challenge for people with a limited ability to move a mouse or to specifically control and target its pointing interface.

Making sure that keyboard options are available and clearly indicated will help considerably. It also helps to provide keyboard commands that are standard and do not require complex sets of keystrokes to execute.

- Use tabs and skip links to get to the main content

It’s very helpful to those with mobility issues to have the ability to use the tab key to navigate, as is making the first tab item a skip link, so as to bypass the navigation elements, if necessary. Verify that the tab order is consistent with the way the material is presented, most often left to right and top to bottom.

- Test your product with a capacitive touch stick

Many users will need an alternative to keyboard input. Both with traditional computer systems and mobile devices, I use a touch capacitance pen or pointing device and hold it between my lips or teeth. The rubberized end acts as a pointer and a substitute for a finger to interact with a typical computer keyboard. It does have some limitations in that I can only press one key at a time, so using “sticky keys” and other time-based key combinations helps me interact with a system.

- On touchscreen devices, closing my lips around the barrel of the pen also allows me to use the human body’s electrical connectivity to utilize the rubberized end on the screen.

This helps extend the ability of a classic touch stick to a smartphone or tablet and register the touch events the same way another user would with their finger.

- Drive your application with voice-recognition software:

For individuals with limited or no mobility in their arms or hands and where movement with a touch stick may be impractical, voice recognition software is a huge advantage. Voice recognition tools allow users to use their voice to execute commands, navigate sites, and fill in the information, as well as compose longer text entries.

I find spending time trying to navigate an application through a voice-recognition application to be both enlightening and frustrating. Commands for voice-recognition software are standardized to common keystrokes for typical computer actions, so making sure that my software can effectively respond to voice commands is important.

6. Vision Issues

When developers plan to make an application accessible to those with visual impairments, often the first thoughts are about screen readers and making sites and applications available to people who are blind. While that is an important aspect, it should not be where the discussion about accessibility stops. Vision is a broad area, and there are a number of accessibility issues in the area of vision. Here are some examples of primary visual disabilities:
Refractive error: Focusing light on the retina. This includes nearsightedness, farsightedness, and astigmatism. In many cases, glasses can remedy this, but some cases are too severe for glasses to help.

Cataracts: A clouding of the lens in the eye that can be slight to severe.

Macular degeneration: A condition that causes blurred or no vision in the center of the visual field.

Color blindness: A situation where vision may be normal in most means but the ability to see certain colors is hindered.

In addition, there are a variety of visual situational disabilities that many of us may deal with:

- Having our eyes dilated as part of a vision procedure.
- Stepping outside into the bright sunlight and trying to see something on our phone or device.
- Corneal abrasion: Think of getting sand or dust blown into your eyes to the level that you can’t wear contact lenses for a few days.

Each of the situational disabilities described above is temporary. Still, each of them would benefit from accommodation, and we should make the applications we develop accessible to as many people as possible.

Design Considerations

How can we make applications more accessible to people with visual impairments? Below are some design suggestions, many of which do not require any assistive technology.

- Use appropriate color contrast.

Using a distinct contrast of colors is especially important when dealing with color blindness, but this design practice is helpful for everyone. The usual combination is dark text against a light background, but this can be reversed if using a dark background. The Web Content Accessibility Guidelines (WCAG 2.1 is the current recommendation) encourages a color contrast of 4.5:1 for medium-sized bold or unbold text and 7:1 for small, unbold text.

- Enable scalable fonts.

By allowing the user to magnify and scale the font size for their needs, users can more easily read the text in a variety of situations.

- Put all the information on the page.

If you have to navigate a site with a vision impairment, it’s frustrating to need to download information as separate documents. In some cases, it’s necessary, such as with binary files or installable executables, but if dealing with text, where possible, make it available within the page itself.

- Avoid using color alone to display meaning.

Green means go, yellow means proceed with caution, and red means stop. For users with normal vision, these color cues are common and can communicate a lot of information quickly. But for a color-blind user, this meaning can be missed. There isn’t a single type of color blindness, so where possible, use text or graphics to display and convey greater meaning than color.

- Make your pages as linear as possible.

Having pages that flow from top to bottom in a single-column format is easier for people who need to increase magnification on a page. The text will align with the magnification level, and the pages will display similarly to how they will at normal magnification. Spreading the content out over multiple columns would make it more difficult to read.
or in a side-to-side format makes it difficult to see where the content is, and navigating the page could be confusing.

- Do not separate elements that go together

By grouping text boxes and buttons that relate to each other, we can make sure that context is not missed and that actions that need to be performed in a certain order are easy to interpret. Think of a form layout that needs to be magnified by 200 percent. If you can’t see the button that goes with a particular text box, it’s a good bet that your users will miss it, too.

**Considerations for Screen Readers**

While the modifications discussed above benefit everyone, there are some situations where assistive technology is needed. Screen readers are applications that take the text that appears on a screen and reads it aloud for the user. There are a number of techniques that can be used to help make this process easier, and as an added bonus, these additions can be checked with automated scripts.

- Describe images and provide transcripts for video

The alt tag is very helpful for displaying additional descriptive meaning for an element. Any image that appears on the screen that is meaningful to a user should have a descriptive alt tag. For a video, transcripts can go into greater detail about what is appearing on the screen.

- Make content easy to navigate

Set the tab order so that the user can more easily navigate to the most important information. The first item should be a link at the top of a page that says, “Skip to main content,” and allows the user to bypass the navigation elements on the page.

- Use HTML5 options to make pages more structured

Use the options available through HTML5 such as article, nav, sidebar, etc. These labels give a much better idea of what is being navigated. Use div tags sparingly, and if they must be used, give an alt tag description that can provide context.

- Provide a keyboard-only option to complete a task

It is common to rely on a mouse or another pointing input device to complete actions, but for screen reader users, a mouse is useless. Navigation and interactions are done via keyboard. If a workflow can’t be completed with just a keyboard, users who rely on screen readers will not be able to use your application.

- If there are links, make the link text descriptive

If a page has multiple links that say “Click here,” it is easy to lose the context of which link goes where. Be descriptive with the link, and define specifically where it will take you.

- All images have an alt tag and have descriptive alt text

This is a simple first step and it’s one that is easily handled by any tool that can examine a web page. If there is an alt tag, there needs to be alt text associated with it. Granted, not every image is going to be meaningful to the user: Some are purely decorative or are repeated inside of a page. If an image is part of the page flow and is relevant to the content being displayed, it is important to verify that there is alt text for these images. For images that are just there for a sense of page aesthetics, null tags can be created.

- Tables have “and” values

Tables are challenging for screen readers and for keyboard navigation. Check that the tables are properly descriptive so that they are easier to navigate and provide context for the data in each row or column. Search for and tags in each table and identify any instances where they are not used.
- Inputs have labels

When using forms, it's important to identify all of the controls and what they represent, especially for assistive technologies like screen readers. The `<label>` tag helps to present that control to the user effectively. Example: If a form is looking for an email address, and the input type is "text" with a name and id of "emailaddress", adding `<label for="emailaddress">` will provide that extra level of context for a screen reader.

## 7. Auditory Issues

Discussions of accessibility for applications are usually dominated by visual considerations. Unless we are dealing with sound directly, it's common to think less about the audible aspects of our sites and applications. However, there are numerous audible components of applications that we can improve, and provide an experience that more people can enjoy whether they are dealing with partial or total hearing loss. The key point to remember is that if there is a way to make content perceivable to people who can't hear, we should take the steps to make that happen.

### Primary Auditory Disability

Like in the previous section regarding visual disabilities, auditory disabilities can be separated into primary (persistent) and secondary (situational) issues. And as with visual loss or limitation, hearing loss or limitation can have various stages and causes. When I consider hearing loss, I try to break it up into ranges:

- **Mild hearing loss**: Difficulty hearing sounds lower than about 30 decibels. If there is background noise, it may be hard to understand someone speaking.
- **Moderate hearing loss**: Difficulty hearing sounds lower than 50 decibels. To put this in perspective, that's approximately the volume of a refrigerator running. A hearing aid is likely to be helpful in this kind of situation.
- **Severe hearing loss**: Difficulty hearing sounds lower than 80 decibels. To put this into perspective, that's the volume of an average washing machine while it is running, or the average alarm clock. At this level, hearing aids may not be helpful.
- **Profound hearing loss**: Difficulty hearing sounds lower than 95 decibels. That's about the volume of a passing subway train.

In addition, there are a variety of auditory situational disabilities that many of us may deal with:

If you are in a noisy restaurant or bar and someone calls you on the phone, you may not be able to hear the phone ringing, even with normal hearing, much less hold a conversation with a person due to the noise level in the room

Ear infections can happen to all of us as a result of sickness, and often they can have a profound effect on hearing

These situations, again, are temporary or situational, but they would require users to have an option to perceive the audible information in an alternative way. Below are a few considerations for testers to verify the ability of hearing-challenged people to perceive audible content, many of which do not require assistive technology.

- Are there text alternatives where possible?

If a video is being displayed, the application should provide closed captions. If presenting a podcast, there should be a text transcript of the content. Additionally, aim to keep the message straightforward, and don't veer into inside jokes or figures of speech that may not be able to be interpreted as written. This is good advice for audio content in general.

- Is the page design smooth and simple?
If there are places in the application where the content is meant to be heard, that should be clear to the user. Follow the suggestion above to make sure there is a way to provide text content for the audio.

- Are audio signals used alone as cues?

If an alert relies on a sound, the application should display a message as well; or, if on a mobile device, vibrate to alert the user.

- Are there a variety of ways to communicate?

In an age where texting seems the most common way to communicate, many organizations still default to using a phone to handle issues. Allow for other ways to let this communication happen.

- Is the content structured clearly?

Too much badly formatted information can be overwhelming for many people, and this is also true for auditory impairment. Well-structured information, with clear headers, bulleted or numbered lists, separation of content, and a lack of clutter can help make pages and applications easier to deal with.

- The alt tag isn’t just for screen readers

The alt tag is the simplest and most straightforward option and can be used with many elements. It’s typically used with images to describe through screen readers, but an alt tag can also be used with audio content to describe what it is. There are limitations to this approach, however, as this would not be appropriate for a long presentation.

- Use track and VTT files for longer text

If we take advantage of the video and audio tags inside of pages, we can also leverage the track tag to provide a text equivalent of the audio that is being presented. The track tag takes a variety of options. Common examples include kind (what the track tag represents), src (the file that contains the content saved as a vtt file), and srclang (which defines the language/locale to use, such as “en”, “de”, “ja”, etc.). If track files are used with audible content, this makes for an easy test to confirm their existence and verify that the breadth of language options we offer are being represented.

- Using the transcript tag to identify transcripts

For truly large text files, the transcript tag is still considered one of the best options. However, the ability to associate a transcript with an audio and video file depends on how the pages and content are structured. A newer feature in HTML5 allows for extending the tag to include transcript as an option. Example: Within a video tag, if the application references a video file, you can extend the track definition to include the kind “transcript” and make a link to a document. The link can be on the same page or it can be in an external document if desired. Once the “kind” transcript is associated, look for the transcript tag as part of a test to verify that it has been included along with the video file. If the goal is to make sure that the content you have worked so hard to create can be shared with (and purchased by, in many cases) the broadest number of people, it makes business sense to allow for multiple options to perceive that information. Just because your application’s users can’t hear the audio content doesn’t mean they would not benefit from it. By giving an alternative and, hopefully, comparable experience, your sites and applications can be useful for more people.

8. Cognitive Issues

Cognitive disabilities cover a variety of conditions and experiences, but because they are not always obvious, they often get overlooked and are perhaps the least understood disabilities.

When it comes to addressing auditory, mobility, or visual issues through your website, the approaches toward accessibility are more straightforward. For visual issues, a screen reader can help. For auditory issues, using closed captioning can replace audio output. Mobility issues allow for voice operation or other methods to enter information and interact. But due to the number of possible causes for cognitive
disabilities, coming up with a set of solutions is challenging. And the goal for truly inclusive design is to engineer your site in such a way that assistive technologies aren’t always required.

Bearing both of these factors in mind, there are several approaches that a tester can look for to verify that sites are accessible by those with cognitive disabilities.

**Types of Cognitive Issues**

As with visual or auditory impairment, there are primary and secondary, or situational, cognitive disabilities. Primary disabilities include the following:

**Down syndrome**: A congenital disorder that comes from having all or part of an extra copy of chromosome 21 (trisomy 21) that involves both physical and cognitive abnormalities

**Autism**: A developmental disorder frequently manifested by challenges with communication, forming relationships, and understanding concepts that are not literal

**Asperger syndrome**: An autism spectrum disorder that appears through difficulties with nonverbal communication and social interaction and frequently manifests a need for repetitive patterns of behavior

**Dementia**: A general term for a variety of symptoms, typically resulting in a decline in memory and a change to a previously normal personality and reasoning

**Dyslexia**: A disorder that makes learning to read and interpret words difficult, manifesting in mixing up letter or word order

**Dyscalculia**: A difficulty in understanding arithmetic, numbers, and mathematical concepts

There are also a variety of examples where cognition can be situationally impacted. These secondary disabilities could include the following:

- Difficulty when having to interact with a foreign language, particularly one that uses a different alphabet
- Being in a distracted, stressed, or emotional state
- Trouble understanding when wading through dense, minimally formatted & punctuated text

Here are a few considerations for testers to verify that an application is accessible by those with cognitive disabilities. Note that many of these do not require assistive technology.

- Is content structured with clear headings, sections, paragraphs, and lists?
  Too much information badly formatted can be overwhelming for many people, and this is especially true for those dealing with cognitive issues. Create well-defined headers, employ bulleted or numbered lists, and use space to separate content.

- Does the user interface avoid overly bright or contrasting colors?
  This is in conflict with advice that would be given for dealing with visual impairment. While it is a benefit to those with visual disabilities, such a stark contrast could be shocking or unsettling to someone with autism or another spectrum disorder. I also recommend limiting or removing elements that flash or otherwise draw attention in a jarring manner.

- Are text alternatives provided?
  Having an audio or video file that goes with a text article can help users with dyslexia or dyscalculia better understand what is being presented. Many people with dyslexia use screen readers to help them digest and understand text passages, so the same techniques used to maximize the ability for screen reader use for those with visual impairments will also benefit those dealing with cognitive issues.
- Are there prompts and visual support for actions?

For many users with cognitive issues, it is not uncommon to lose track of what they are doing in the middle of a workflow. Being forced to remember a variety of steps to accomplish a task may be asking too much for many users. By providing prompts to highlight the previous and next steps, we can help them keep track of where they are and what they should be doing.

- Use breadcrumbs to identify where a user is in the page hierarchy

By implementing a breadcrumb option, you can validate that pages are appearing in the correct order, as well as determine that they can be navigated by following the breadcrumb trail as a series of links. This works in conjunction with testing the navigation elements on pages. Having a consistent design for both navigation elements and breadcrumbs can help users know where they are and what they are doing.

- Provide options for text resizing

When users can rescale the text they are viewing, it can make for a more comfortable reading experience. Having buttons that cannot be resized will cause pixelation when they are enlarged, in some cases making them illegible or indeterminable. By looking for elements that use the CSS box model and inline styles, those elements can be resized, keeping their actions recognizable to the viewer.

- Use the `<ABBR>` and `<ACRONYM>` HTML tags

It is common to see a variety of abbreviations or acronyms used across web pages, particularly in the software industry. These may not be understood by many readers, and not just those with cognitive disabilities. But if you use the `<ABBR>` and `<ACRONYM>` tags, users can hover over the letters and see what each abbreviation or acronym means in full text. You can make tests to search for all acronyms or abbreviations and determine whether accompanying tags and titles are provided.

- Reformatted justified text

If a paragraph uses full justification, words will appear equally spread out along the line, instead of centered or aligned left or right. This can have the unintended consequence of making the material difficult to read for many people. A simple test that looks for and highlights paragraphs that are formatted with `<p align="justify">` can help pinpoint where there may be difficulties.

Making our sites usable by the largest possible group of people is an important goal. Unfortunately, it is impossible to make every site completely accessible to those with cognitive impairments due to the range and varying severity of issues, but that should still be the aim.

**Conclusion**

The design decisions made early in the life cycle of products have the potential to make them excellent solutions to issues they face, or genuine nightmares to use. The farther along a product gets in its development, the more difficult it is to make modifications to its design.

Think of the applications that you would want to use, and think of yourself in the future, with the possibility that a significant disability (or disabilities) may be part of your everyday experience.
References


Census Bureau Reports. Nearly 1 in 5 People Have a Disability in the U.S. https://www.census.gov/newsroom/releases/archives/miscellaneous/cb12-134.html

