Designing Accessible Websites

A Research Paper submitted to the Department of Engineering and Society

Presented to the Faculty of the School of Engineering and Applied Science

University of Virginia • Charlottesville, Virginia

In Partial Fulfillment of the Requirements for the Degree

Bachelor of Science, School of Engineering

Jonathan Hail

Spring 2023

On my honor as a University Student, I have neither given nor received unauthorized aid on this assignment as defined by the Honor Guidelines for Thesis-Related Assignments

Advisor

Joshua Earle, Department of Engineering and Society

Introduction

"Up to 1 in 4 (26 percent) adults in the United States have some type of disability." ("Disability Impacts All of Us", 2023). With the increasing number of people with disabilities and the number of older citizens increasing, having websites that are accessible is more important than ever (Adam & Kreps, 2006). Alison Adam and David Kreps describe that according to a survey in the UK, less than 20 percent of websites meet even the most basic accessibility standards. There is currently legislation in website accessibility, it is relatively new and hasn't made a huge impact in ensuring websites remain accessible (Adam & Kreps, 2006). This paper will use literature review to portray the history of accessibility, the current state of accessibility in websites, and how accessibility can be improved and ensured.

History of Website Accessibility

One of the earliest legislations that put accessibility forward was The Rehabilitation Act of 1973 (Sapega, 2020). This required federal agencies to not discriminate against disabilities, which required accessibility features in front facing user design. This was later interpreted to include websites, which increased demand for design of accessible websites. Later, Congress passed the Americans with Disabilities Act in 1990. This act required more accessibility features broadly, which includes a lot of physical accessibility features such as wheelchair ramps. However, this also applies to websites, as multiple judges have ruled that this applies to them as well. One of the instances of this is in 1996, when the Department of Justice ruled that Title 3 of the ADA includes websites as a public accommodation that must be accessible. However, this act is not specific on what qualifies a website as accessible, and can be difficult to prove one way or another.

In 1998 Congress amended the Rehabilitation Act of 1973, as the internet wasn't as prevalent when this act passed Congress (Sapega, 2020). Congress revised Section 508 of the Rehabilitation Act of 1973 to specifically include that not only federal agencies but any companies that want to do business with the government must have accessible digital assets. This clearly includes that websites are required to be accessible, and for more organizations than solely the government. In 1999, new guidelines for web accessibility were released by Congress, with the first iteration of Web Content Accessibility Guidelines (WCAG). World Wide Web Consortium (W3C), a group founded by Tim Berners-Lee to set standards on web accessibility, founded these guidelines. However, these guidelines were merely guidelines, and not required for websites.

The next major legal step for accessibility in websites was in 1999/2000, with the Maguire v Sydney Organizing Committee for the Olympic Games (SOCOG) Australian court case (Sapega, 2020). This case argued that SOCOG discriminated against blind people as their website was not accessible to purchase tickets. Maguire ultimately won, which caused Australian government websites to shift to adopt WCAG design.

In 2008, W3C published WCAG 2.0 (Henry, 2005). W3C designed WCAG 2 guidelines to be perceivable, operable, understandable, and robust. These ensure there are clear rules to ensure accessibility. Future versions of WCAG released include 2.1, which was released in 2018, and the 2.2 draft, which is planned to be released in 2023. One impact of this is the inclusion of WCAG 2.0 guidelines in Section 508 of the Rehabilitation act of 1973, which was added by Congress in 2018 ("Section 508 of the Rehabilitation Act of 1973", 2022). This also affected sections 501 and 504, which prevents federal employers from discriminating against

people with disabilities, and requires federal agencies and activities to provide reasonable accommodation to people with disabilities.

One of the recent examples of legislation is the VA Website Accessibility Act of 2019. This act passed the House of Representatives in March, 2020. In it, it describes that there are over 130,000 legally blinded veterans in the US, and the Rehabilitation Act of 1973's section 508 requires VA websites to be accessible to this population (VA WEBSITE ACCESSIBILITY ACT OF 2019, H.R. 1199, 116th Congress). However, the bill found that Virginia is in a state of perpetual adoption, and a lot of state websites remain inaccessible. Some of the features that make it this way include poorly formatted tables, small user interface items, improperly labeled items, pop-ups that interfere with navigation, and more. This shows that even though there is legislation that requires governmental websites to be accessible, there still may be action required to ensure that the websites follow suite.

Importance of Website Accessibility

This topic is important to ensure that new technologies can be utilized by the widest audience as possible, as it is important to include people in technologies that can empower them. If more developers are aware about differing disabilities that prevent a certain population from using their software, they could add more design features that allow the wider audience to user their software. This would be mutually beneficial for both developers and the general public as the developers would increase the audience of their software and it would allow the public to have equal opportunity to put said software to use. Adding more accessible design to websites would also impact a large amount of the population: "A 2003–2004 study commissioned by Microsoft and conducted by Forrester Research found that over half—57 percent—of computer

users in the United States between the ages of 18 and 64 could benefit from accessible technology" (Grieves, 2009). As these features are widely beneficial, it can also cause an increase of users ("Why Web Accessibility is Important: 4 Reasons to Create Accessible Content", 2022).

While accessibility features can help bridge the gap between the general populace and users who rely on such features, these features can benefit all users. Features that help make software more accessible can also be of use for convenience, and generally improve software quality. These features can be categorized into several areas, and can benefit people in different ways. Some of these include closed captioning on videos and proper formatting to fit on various devices.

Websites with accessible features can also cost less to maintain ("Why Web Accessibility is Important: 4 Reasons to Create Accessible Content", 2022). When making websites accessible, developers optimize code behind the website, which could lead to performance improvements by reducing unnecessary resources and load. Reducing this load could also decrease the cost to keep the website running. This improved experience can be beneficial by making the website more popular as well.

How Website Accessibility Could be Improved

Software accessibility could aid a large number of people that have a disability. Disabilities that affect one's ability to access websites can be classified into the following: vision, dexterity, hearing, and cognitive (Grieves, 2009). Implementing features in the software design process could mitigate each of these categories. For vision disabilities, an option to increase font size or style, and screen readers can help increase ease of access (Grieves, 2009).

For dexterity, features like on-screen keyboards or other methods of input can decrease strain of the precise mechanical input of a keyboard (Grieves, 2009). For hearing disabilities, options for volume control, text captioning, or sign language let users who have trouble understanding audio in software to understand the language properly (Grieves, 2009). Finally, features that assist with cognitive disabilities are a simplified user interface, intelligent suggestions for user input, reminders for user action, and reading/learning aids (Grieves, 2009). These features not only help those who need them to use the software equally, but it also helps other users generally using the application. For example, many people enjoy using closed captioning in software such as Netflix to assist in clearly understanding everything said.

To change existing websites from inaccessible design for the better, the HTML behind the website must be changed to reflect this. There exist validators such as the W3C HTML validator (to help check if website code is accessible) to assist in this process, however, most change requires a human to do it themselves (Harold, 2008). Accessibility can be tested by trying to access the website in question in multiple different ways that represent how users with different disabilities would navigate the website. This includes making sure that it is easy to navigate if it can't be seen, if it had to be navigated through voice, and if it is being accessed through different devices (mobile, desktop, text only). Alternatively, the website can be tested by users who have such disabilities to ensure they are properly able to access it. Some strategies and design choices that can be made to convert websites to be accessible like this include converting images to text in a way that is understandable, adding labels to forms, standardizing inputs so they can make more sense to users and browsers alike, table descriptions summarizing content, and introducing better navigational subsections to the webpage.

However, making every website accessible can prove challenging. Due to the increased complexity of newer websites and nonlinear content (pages that are more dynamic and complicated than just text), even if the website follows the guidelines set by WCAG, visually impaired users could have some difficulty navigating the content (Raufi et al., 2015). These issues include: navigational context of current text position, excessive information of reading unnecessary content such as headers and links., and lack of information to identify menu bars. Bujar Raufi and coauthors proposed a set of techniques to overcome these issues. These are displaying content without text allowing it to be transformed or removed to be more accessible, reformatting text to make it more linear, add metadata (such as text that describes an image in the background) to facilitate screen readers, and using proper serialization on important information such as links in the text. These techniques solve the previously mentioned issues and would make websites easier to understand to the visually impaired.

A lot of websites and laws use WCAG guidelines as a standard, however, WCAG is not without its flaws (Abuaddous et al., 2016). WCAG are merely guidelines, and following all of them doesn't guarantee that the end website will be completely accessible. Furthermore, there isn't a significant amount of evidence proving that WCAG 2.0 improves the experience of disabled users. The guidelines put forward by WCAG 2.0 are also hard to navigate and implement by web developers. This can prevent websites from becoming more accessible.

Even when managing the accessibility of a website in development, there is an abundance of challenges which can decrease conformance (Abuaddous et al., 2016). Government enforcement of accessibility features in website in some countries can be lacking, which decreases pressure on companies developing these websites to ensure the website is accessible.

This coupled with not many resources for training on website accessibility could result in the website not being as accessible as it could. This issue can be exacerbated for websites designed in developing countries. There is even less accessibility awareness, training, resources, and manuals for developers to utilize.

Another possible way to improve website accessibility is through the use of artificial intelligence (Abou-Zahra et al., 2018). Artificial intelligence has many possible applications to improve website accessibility, both on the side of the user and developer. Some methods to improve the user experience through AI proposed in Abou-Zahra et al. (2018), are to detect objects on websites with advanced interactive designs, conveying visual information from augmented or virtual reality on the web into text, and adjusting the accessibility settings of a website to better suit the user. These methods utilize current AI technologies, including image recognition and natural language processing. AI could provide a substantial benefit in these areas, as it could significantly reduce the time it would take for the creators of the content to implement it themselves. Shadi Abou-Zahra and coauthors proposed methods to improve accessibility on the side of development. These methods include ensuring websites have a way to edit AI generated text for accuracy and using AI to aid in the visual design of a website to promote accessibility.

The current and proposed uses of AI are beneficial; however, artificial intelligence is currently an emerging technology, and there are some downsides to it. For example, sometimes it can produce inaccurate or completely wrong information (Abou-Zahra et al., 2018). AI may also cause issues with sensitive data. Also, to create an algorithm to utilize AI, it must be trained on a data set. The quality of this data set is important, as having certain biases in the data can affect the overall quality of the content the AI is being used to provide. These issues could

improve in the future though, since AI advancement has been happening at a rapid rate. Even with these downsides, AI could still be incorporated into a solution that improves experience for users by increasing the amount of accessibility.

Testing for Website Accessibility

To ensure accessibility features make their way into modern websites, it is required to test for them. This can either be manual or automated. Manual testing is where specific features are tested individually by a human, whereas in automated testing they are ran by a script. The benefits of automated testing are in they are created once and can be run as the software is in multiple stages of development. However, there are some issues with relying solely on automated testing: "the problems of inaccessibility are further compounded by a reliance on automatic checkers, which cannot possibly verify the accessibility of a web site without a human check" (Adam & Kreps, 2006). Thus, manual testing is still important. However, not all features must be checked manually, and manual testers can work in more niche areas (Palani, 2019). Automated testing can take place using tools such as Selenium, which can test if the website conforms to accessibility principles as it's developed, which has been best practice since 2017 (Palani, 2019). In conjunction with tools such as JAWS and NVDA (screen reading programs), developers can use automated testing to receive good coverage (Palani, 2019).

Currently, there are a variety of methods used to measure accessibility of websites. They include expert review, user testing, subjective evaluations, and barrier walkthrough (Brajnik, 2008). Expert reviews are the most common, which reviews the website based upon WCAG guidelines. These are generally more cost-effective, since they can utilize automatic testing. However, this method is not always the most reliable, and can also produce issues that are not

the most important, taking away from developer time. User testing identifies these lesser important issues less frequently. The issues identified also will represent the target audience better, as it would be tested by potential users. However, it can be difficult to find users with disabilities to test the accessibility of the website with, and important issues can still be overlooked.

There can be some challenges faced when implementing tests for website accessibility. Very large websites, websites that have a large number of different states, and third-party content on the websites can make testing for web accessibility difficult (Sajka et al., 2020). The first one is an issue because websites can use a lot of HTML, which can be automatically verified that it is to a certain specification, but not that the meaning stays the same. As Janina Sajka and coauthors put it: "text on a web page marked as contained in a paragraph element may not trigger any failure in an automated test, nor would an image with alternative text equal to 'red, white, and blue bird', but a human will identify that the text needs to be enclosed in a heading element to reflect the actual use on the page, and also that the proper alternative text for the image is 'American Airlines logo'" (Sajka et al., 2020). This could possibly be resolved through using libraries for such images that are pre-checked for accessibility in the alt-text. This information would make the website more understandable to the visually impaired when using a screen reader.

The next challenge mentioned was large websites having a lot of changing elements that make it difficult to ensure all states remain accessible. These elements can include user content and data, locational differences, and rendering differences by device and browser (Sajka et al., 2020). This can be mitigated through the use of unit testing and integration testing. Unit testing tests one particular element of the website to where other information doesn't affect it, and can

be used to decrease the likelihood of there being a condition where the website becomes less accessible. Integration testing tests features together, to check if the features interact in a way that could make it less accessible. These methods can help prevent situations from happening where inaccessible features arise during use that wasn't apparent on release, but cannot ensure it doesn't happen either.

Third-party content can also make a website less accessible. One example of this is website advertisements (Sajka et al., 2020). However, there isn't a way to make sure that the content included in these is accessible for users, as only the third party can change their own content. One way to lessen this impact is though metadata such as alt text. This third-party content is also a mandatory feature of some websites (Hackett et al., 2003). This can be seen in content that allows the presentation of multimedia content such as: video, sound, presentations, movies, and more. Content such as these often aren't supported by browsers used by people with disabilities.

Conclusion

Accessibility in software is important because issues excluding populations should be resolved, as they can have certain political consequences, some directly and some indirectly. The direct consequences of this would be the immediate exclusion of the potential users from using the software. This is not ideal as it not only excludes potential users for the developer, but also is preventing certain people from using it. The indirect consequence of this is whatever the software could be enabling users to do, such that potential users that cannot access the software are also excluded from the end use. Steps are being taken in law and in practice to ensure this happens, and hopefully it will reach a point where everyone can use the same resources equally.

Bibliography

Abou-Zahra, S., Brewer, J., & Cooper, M. (2018). Artificial Intelligence (AI) for Web Accessibility: Is Conformance Evaluation a Way Forward? Proceedings of the 15th International Web for All Conference, 1–4. https://doi.org/10.1145/3192714.3192834

Abuaddous, Hayfa. Y., Zalisham, M., & Basir, N. (2016). Web Accessibility Challenges. International Journal of Advanced Computer Science and Applications, 7(10). https://doi.org/10.14569/IJACSA.2016.071023

Adam, A., & Kreps, D. (2006). Enabling or disabling technologies? A critical approach to web accessibility. Information Technology & People, 19(3), 203–218. https://doi.org/10.1108/09593840610689822

Brajnik, G. (2008). A comparative test of web accessibility evaluation methods. Proceedings of the 10th International ACM SIGACCESS Conference on Computers and Accessibility, 113–120. https://doi.org/10.1145/1414471.1414494

Disability Impacts All of Us Infographic | CDC. (2023, January 5). Centers for Disease Control and Prevention. https://www.cdc.gov/ncbddd/disabilityandhealth/infographic-disability-impacts-all.html

Grieves, J. (2009). Engineering Software for Accessibility. Microsoft Press. https://learning.oreilly.com/library/view/engineering-software-for/9780735642102/

Hackett, S., Parmanto, B., & Zeng, X. (2003). Accessibility of Internet websites through time. Proceedings of the 6th International ACM SIGACCESS Conference on Computers and Accessibility, 32–39. https://doi.org/10.1145/1028630.1028638

Harold, E. (n.d.). Refactoring HTML: Improving the Design of Existing Web Applications. Retrieved December 8, 2022, from https://learning.oreilly.com/library/view/refactoring-html-improving/9780321552044/

Henry, S. (2005, July). WCAG 2 Overview. Web Accessibility Initiative (WAI). https://www.w3.org/WAI/standards-guidelines/wcag/

Palani, N. (2019). Advanced Selenium Web Accessibility Testing: Software Automation Testing Secrets Revealed. Momentum Press. http://ebookcentral.proquest.com/lib/uva/detail.action?docID=5742318

Palani, N. (2022). : Development and Testing Best Practices. Auerbach Publications. https://doi.org/10.1201/9781003299431

Raufi, B., Ferati, M., Zenuni, X., Ajdari, J., & Ismaili, F. (2015). Methods and Techniques of Adaptive Web Accessibility for the Blind and Visually Impaired. Procedia - Social and Behavioral Sciences, 195, 1999–2007. https://doi.org/10.1016/j.sbspro.2015.06.214

Sajka, J., Korn, P., & Hall, C. (n.d.). Challenges with Accessibility Guidelines Conformance and Testing, and Approaches for Mitigating Them. Retrieved December 8, 2022, from https://www.w3.org/TR/accessibility-conformance-challenges/

Sapega, M. (2020, March 21). The History of Digital Accessibility and Why it Matters. TPGi. https://www.tpgi.com/the-history-of-digital-accessibility-and-why-it-matters/

Section508.gov. (n.d.). Retrieved February 26, 2023, from https://www.section508.gov/manage/laws-and-policies/

United States, & United States (Eds.). (2006). Research-based web design & usability guidelines (Version 2). U.S. Dept. of Health and Human Services : U.S. General Services Administration.

VA WEBSITE ACCESSIBILITY ACT OF 2019. (2019). VA WEBSITE ACCESSIBILITY ACT OF 2019, H. Rept. 116-127, 1–6.

Why Web Accessibility is Important: 4 Reasons to Create Accessible Content. (n.d.). Retrieved March 15, 2023, from https://www.boia.org/blog/why-web-accessibility-is-important-4-reasons-to-create-accessible-content