1 Introduction

2 What is web accessibility and why is it important?

1. What is disability?
   Some more general accessibility principles not related ot web a11y
2. Universal design
3.

WHO estimates that 1.3 billion people – or about 16% of the global population – experience a significant disability and this number is growing. Disability is a part of being human and persons with disabilities are different from each other. (World Health Organization, 2022)

2.1 Web accessibility

”The power of the Web is in its universality. Access by everyone regardless of disability is an essential aspect,” said Tim Berners-Lee, W3C Director and inventor of the World Wide Web(World Wide Web Consortium, 1997).
Essentially accessibility in providing equal access to goods and services to everyone regardless or their age, gender or disabilities. These principles can be applied to any various fields. For example in architecture it could be designing buildings in a way that that can be accessed by people who can walk on their own as well as the ones who need to use a wheelchair. In digital products it is more often related to the senses we use to consume content. Everything could be equally accessible whether I want to see or hear it for example.
Web accessibility is standard practices and requirements are defined in WCAG. People who consume web content have different abilities, because of their experience or different conditions that might affect their physical or mental capabilities. Someone might break their hand and be temporarily unable to use a mouse to navigate, or they might have been born blind and rely entirely on assistive technologies to consume content. The same content should be equally accessible for them both.

TODO: Some history. How did accessibility get started what is the history behind web accessibility?

3 Common ways of testing for accessibility?

What are the most common methods for testing?

- Testing with users
- Testing using experts
- Testing by using checkers that run in a browser

4 Automated accessibility testing.

- How can accessibility testing be automated?

- In software development CI/CD (Continuous integration / Continuous delivery) is commonly used. How can we use the same principle for automated testing. Look et this (Sane, 2021)
- What are common strategies used by other companies?

5 Case study

Tried out some automated accessibility strategies in Pipedrive. Pipedrive has been developing a sales CRM using mostly typescript and React. Accessibility has never been high priority and at this point it is not very easy to get started. We have a design system and a React based component library to keep the look and UX consistent. This seems like a good place to start with solving accessibility issues. If a button in the reusable library gets fixed 95% of the buttons in the webapp that our customers use should be improved.
It is commonly advised by accessibility specialist that don't build a component library from scratch. Ours uses some other libraries as the base, but a lot of it custom. Currently, we are adopting inner sourcing – every developer in the company is free to contribute to the library. Automated testing can help keep a better quality standard

5.1 Storybook and a11y-addon, axe-core

- What is storybook and what is it used for?
  Short description of the tool.
- How does this tool work?
  Uses axe-core in the background. How does axe-core work
- What issues are tested for each example?
  Based on HTML tags? What else? There is a standard for these rules so they could be checked better (Initiative (WAI), n.d.).

5.2 Steps

1. Set up automates accessibility issue detection in storybook adding a11y-addon + find a way to generate report of all issues.
   Data gathered from automated testing report:
   - How many occurrences in the accessibility violations report. This will show how many violations where detected from all the examples. Might contain the same issue multiple times.
   - How many unique issues will only count different violations for each component.
   - How many passed checks – this together with violations will show how many things where tested for each component – Most component have more than one story – the list will contain all different passed checks listed
   - How many valid checks – are the passed checks relevant to the component – only count the ones that are related to the component that the example is about.
2. Manual accessibility audit with other team members. We already had the addon set up, and we used storybook preview of components for testing, so we looked at the violations reported by the addon-a11y also.

3. Comparison between manual and automated report

4. Research other possibilities. Testing new storybook test-runner to automatically run all a11y tests. How can we ignore some issues without losing the info.

5. Set up a new solution in a new component library. Will it help to ensure better accessibility from the beginning?

5.3 Future plans/improvements

Test runner that works in the next major version so that the tests can be run the same way as unit tests in CI workflow. This would ensure more visibility and make it easier to keep an eye on the issues.

** All references are listed remove before submitting

References

Byrne-Harber, S. (2021). *Giving a damn about accessibility: A candid and practical handbook for designers.* UX Collective. https://www.accessibility.uxdesign.cc/

Deque Systems. (2021). *Developing axe-core rules* [original-date: 2015-06-10T15:26:45Z]. Retrieved 02/18/2023, from https://github.com/dequelabs/axe-core/blob/develop/doc/rule-development.md

Henry, S. L., Abou-Zahra, S., & Brewer, J. (2014). The role of accessibility in a universal web, 1–4. https://doi.org/10.1145/2596695.2596719

Hinz, K. (2016). *Designing for Accessibility: The Ultimate in UX*. Retrieved 12/26/2022, from https://medium.com/tealmedia/designing-for-accessibility-the-ultimate-in-ux-e366165d0db7

Horton, S., & Quesenbery, W. (2013). *A web for everyone: Designing accessible user experiences*. Rosenfeld Media OCLC: ocn871042774.

Initiative (WAI). (2022). *Introduction to Web Accessibility*. Web Accessibility Initiative (WAI). Retrieved 12/26/2022, from https://www.w3.org/WAI/fundamentals/accessibility-intro/

Initiative (WAI), W. W. A. (N.d.). *Accessibility Conformance Testing (ACT) Overview*. Retrieved 02/18/2023, from %5Curl%7Bhttps://www.w3.org/WAI/standards-guidelines/act/%7D

Interaction Design Foundation. (N.d.). *What is Accessibility?* The Interaction Design Foundation. Retrieved 12/26/2022, from https://www.interaction-design.org/literature/topics/accessibility

Kramer, N. (2018). *A Primer to Web Accessibility for Designers*. Medium. Retrieved 12/26/2022, from https://uxplanet.org/a-primer-to-web-accessibility-for-designers-2c548448c612

Sane, P. (2021). A brief survey of current software engineering practices in continuous integration and automated accessibility testing. *2021 Sixth International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)*. https://doi.org/10.1109/wispnet51692.2021.9419464

Wikipedia. (2022). Accessibility. In *Wikipedia*. Retrieved 12/26/2022, from https://en.wikipedia.org/w/index.php?title=Accessibility&oldid=1124028549
Page Version ID: 1124028549

World Health Organization. (2022). *Disability*. World Health Organisation. Retrieved 12/26/2022, from https://www.who.int/news-room/fact-sheets/detail/disability-and-health

World Health Organization & World Bank. (2011). World report on disability 2011. Retrieved 12/26/2022, from https://apps.who.int/iris/handle/10665/44575

World Wide Web Consortium. (1997). *World Wide Web Consortium Launches International Program Office for Web Accessibility Initiative*. Retrieved 12/26/2022, from https://www.w3.org/Press/IPO-announce

World Wide Web Consortium. (2019). *Accessibility Conformance Testing (ACT) Rules Format 1.0*. Retrieved 02/18/2023, from https://www.w3.org/TR/act-rules-format/