

# 1 Introduction

## 2 What is web accessibility and why is it important?

1. What is disability?  
Some more general accessibility principles not related to web ally
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. (**WHO Disability 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 (**worldwidewebconsortium19**). Essentially accessibility in providing equal access to goods and services to everyone regardless of 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 (Continuous integration is commonly used). How can we use the same principle for automated testing.
- What are common strategies used by other companies?

## 5 Case study

Tested out some automated accessibility testing strategies in Pipedrive. Decided to test out some ways of testing for accessibility in our component library. It is commonly advised by accessibility specialists that don't build a component library from scratch. Ours uses some other libraries as the base, but a lot of is 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

During the time of this testing axe-core improved in evaluating color contrast for disabled elements. At the time of the audit they were evaluated for color contrast issues, but later this has been improved to comply with the change in WCAG standard.

The issue with Modals, dialogs and similar elements is that they never get checked. Could we improve this by changing the examples provided in storybook **DequeSystems2021**.

- 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.
- What issues are tested for each example?  
Based on HTML tags? What else?

### 5.2 Steps

1. Set up automated 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 were 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 were tested for each component – Most components 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