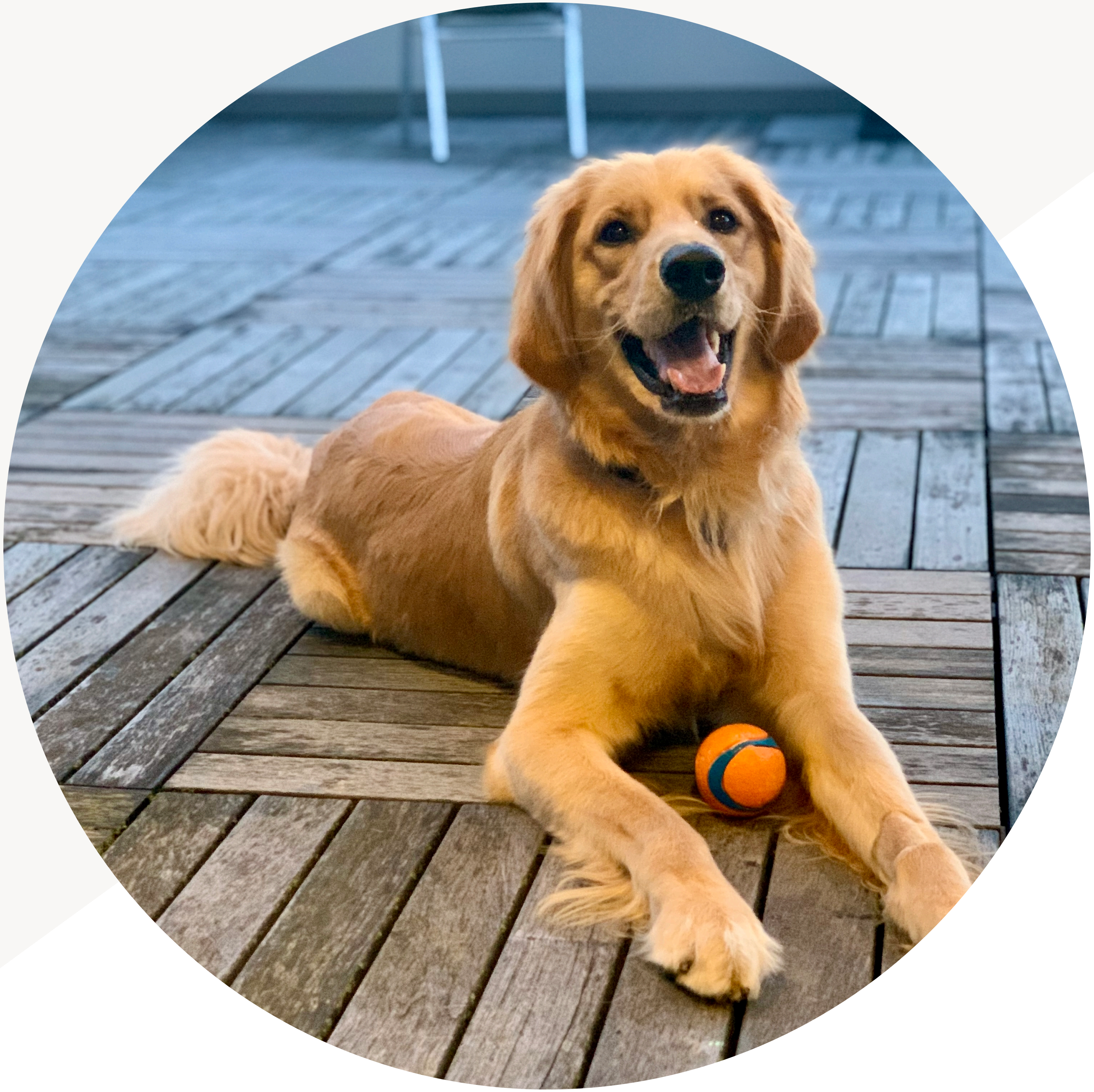
From **Audit** to **Action**

A Four-Step Guide to Accessibility Testing

Kaitlin Bolling • Director of UX Development at Alley

ColorCode: Buffalo 2025



.Alley

STEP 1

Requirements Gathering

Identify and define the requirements needed for the audit. This could include:

- Audit Scope
- Conformance Level (ex: WCAG AA 2.2)
- Evaluation Methods and Tools
- Specific device considerations
- Documented currently known areas of interest
- Deliverables and reports

STEP 2

Automated Testing

What is automated testing?

Automated testing uses software tools to evaluate a website or application against a set of accessibility standards, such as WCAG.

It is your first line of defense and can quicky identify many common accessibility violations.

PROS

- Efficient
- Consistent
- Low barrier to entry

CONS

- Only detects about 30-40% of all issues
- Can't evaluate the quality or appropriateness of accessibility solutions
- Struggles with dynamic content and state changes

How to Conduct Automated Testing

There are many testing tools available. Choose the tools you are most comfortable with, and select at least two for the most comprehensive results.

Automated Testing Tools

- Accessibility Insights (by Microsoft) - Browser extensions with assessment workflo
- AccessLint - GitHub integration for catching issues in pull requests
- ARC Toolkit - Chrome extension for developers
- Axe (by Deque Systems) - Available as browser extension, API, and CLI tool
- IBM Equal Access Accessibility Checker - Browser extension and automation tools
- Lighthouse - Built into Chrome DevTools with accessibility audit features
- Pa11y - Command-line interface for automating accessibility tests
- Siteimprove Accessibility Checker - Browser extension and platform
- WAVE (Web Accessibility Evaluation Tool) - Browser extension and online service

Google Lighthouse

- The Lighthouse Accessibility score is a weighted average of all accessibility audits.
- Weighting is based on axe user impact assessments by *deque*.
- Each accessibility audit is pass or fail.
- More on the Lighthouse accessibility score

Siteimprove Accessibility Checker

- The score is a measure of how well the page performs against WCAG success criteria categorized by Level A, AA, or AAA conformance levels.
- https://www.siteimprove.com/toolkit/accessibility-checker/

STEP 3

Manual Testing

What is manual testing?

Manual accessibility testing involves human evaluators directly interacting with digital products to identify accessibility barriers that automated tools cannot detect.

This process requires testers to simulate the experience of users with disabilities by using various assistive technologies.

PROS

- Identifies issues that automated tools cannot detect (60-70% of accessibility problems)
- Evaluates the actual user experience rather than just code compliance
- Identifies problems with dynamic content and state changes

CONS

- Time-intensive and resource-demanding
- Subject to human error and inconsistency between testers

Keyboard Testing

The primary goal of keyboard testing is to ensure that all interactive elements and functionality of a website or application can be accessed and operated using only a keyboard, without requiring a mouse or touch input.

This is essential for users with motor disabilities, vision impairments, or those who rely on assistive technologies that emulate keyboard input.

Standard keyboard controls

- **Tab** key to move forward through interactive elements
- **Shift+Tab** to move backward
- **Enter/Space** to activate buttons and links
- **Arrow keys** for menus, radio buttons, and sliders
- **Escape** for closing dialogs or canceling actions

Considerations

- Focus visibility
- Tab order
- Keyboard traps
- Skip links
- Access to all functionality
- Custom widgets
- Modal dialogs
- Hidden content
- Focus management

Demo

Screen Reader Testing

Screen readers are assistive technology software that converts digital content into synthesized speech or braille output for people who are blind or have low vision.

Popular screen readers:

- JAWS and NVDA (Windows)
- VoiceOver (Mac/iOS)
- TalkBack (Android)
- Orca (Linux)

Most common screen reader and browser combinations

VoiceOver

- Apple's built-in screen reader
- Works best with Safari
- Settings > Accessibility > VoiceOver Utility
- Start with: `Command + F5`
- Cheatsheets:
  - deque Quick Reference Guide: macOS Keyboard Commands
  - Apple VoiceOver Command Charts

Demo

STEP 4

Reporting

Compile accessibility testing results into a report that **summarizes findings, details violations, and suggests remediation strategies** with the goal of **making remediation as easy as possible.**

The final deliverable should be thorough but easy to understand, and should help **inform the prioritization** of any remediation suggestions.

Suggested Report Format

I. Table of Contents

II. Introduction/Executive Summary

III. Violations snapshot (optional)

  A. Violations by severity

  B. Violations by conformance level

  C. Defect clusters

IV. Background

  A. Pages tested

  B. Platforms and tools

  C. Conformance level(s) tested

  D. Exceptions

V. Violations

Violation form template

ID	Unique identifier for referencing this violation (e.g., V1, V2, V3)
Violation	General description of the violation
Appears on	Pages / templates on which the violation is experienced
WCAG criteria & level	The relevant Web Content Accessibility Guidelines success criteria and conformance level
Description	Details about the specific violation
Severity level	Degree of impact
Impact	How this violation can affect users with disabilities
Remediation	Suggestions for resolving the violation

Degree of Impact

CRITICAL The issue results in blocked content for individuals with disabilities. Until a solution is implemented, content will be completely inaccessible, making the organization highly vulnerable to legal action. Remediation should be top priority.

SERIOUS The issue results in serious barriers for people with disabilities. Until a solution is implemented, some content will be inaccessible, making the organization vulnerable to legal action. Users relying on AT will experience a significant frustration when attempting to access content. Remediation should be a priority.

MODERATE This issue results in some barriers for individuals with disabilities but would not prevent them from accessing fundamental elements or content. This might make the organization vulnerable to legal action. The violation must be resolved before a page can be considered fully compliant.

MINOR This is considered an issue that yields less impact for users than a moderate issue. For a page to be considered fully compliant this issue must be resolved but can be dealt with last.

Tips

- Document violations in a form as you go. Include screenshots. Very brief videos can help demonstrate violations that involve interaction(s).
- Violations should be grouped so that each violation should only be reported once, along with notes on where it appears.
- If there are multiple ways to remediate a violation, suggest the less disruptive one - that is, the solution that is less likely to cause regressions.
- Consider taking a course through Deque University to fill in any knowledge gaps and refine your skills.

Accessibility Audit Reporting Software

- Be Inclusive - beinclusive.app
- CAAT - caat.report

☑ Requirements Gathering

☑ Automated Testing

☑ Manual Testing

☑ Reporting

□ ✨ User Testing ✨

Thank you!

Slides: https://github.com/kaitlinbolling/accessibility-testing