Proceedings of the 2019 ICT Accessibility Testing Symposium: Perfecting Traditional Methods, Tackling Emerging Interfaces, and Beyond

Washington DC

October 1 & 2, 2019

Partnership for Public Service Conference Center,

1100 New York Avenue NW, Washington, DC, 20005, USA

Organized by Accessibility Track Consulting, LLC

Accessibility Track™

www.ictaccessibilitytesting.org

Contents

Introduction from the Chair

Dr. Chris M. Law

Chair, The 2019 ICT Accessibility Testing Symposium

The Information and Communications Technology (ICT) Accessibility Testing Symposium continues to grow each year. For the main two days of the symposium we have a packed and varied program. For our first theme, 'Perfecting Traditional Methods', many submissions relate to the management and implementation of testing programs in real-world settings. For our second theme, 'Tackling Emerging Interfaces', we have papers on augmented, virtual and mixed reality, as well as artificial intelligence and advanced text-to-speech development. Both themes bring to the forefront of our minds the long-standing quandary of how we can address both technical standards conformance and usability. Our keynote speaker has pioneered guidance and practice at the intersection of accessibility and usability.

Last year we offered a bootcamp course for beginners to learn accessibility testing. This year we're offering a super-sized version: a two-day bootcamp course on testing that is preceded by the symposium and a full day workshop, along with mentoring and coaching sessions throughout the week with the instructor and members of the symposium committee.

We have a diverse range of full- and half-day workshops prior to the symposium newcomers catering to beginner and intermediate (and above) levels. We are also offering a post-conference bootcamp course on managing risk in accessibility programs, led by yours truly and Pina D'Intino. This course, tailored to accessibility testing management professionals, is a modified version from that taught for the first time ever at this year's Digital Accessibility Legal Summit, which took place in March 2019.

On behalf of the Committee, we invite you to join like-minded colleagues in DC this fall for our fourth annual symposium.

Sincerely,
Chris M. Law, Chair, 2019 Symposium Committee

Keynote: "The Missing Link: Accessibility and Usability Working Together"

Shawn Lawton Henry

Accessibility is not about meeting standards. However, accessibility is often approached as a standards checklist. Accessibility is fundamentally about people— designing products so that people with disabilities can use them effectively. When you shift the focus of accessibility to people, you reap all sorts of benefits, such as more efficient development and evaluation. Traditional usability practices work well for accessibility. And general usability benefits from addressing accessibility. In this keynote, you'll learn how to get your boss and colleagues on board, how to maximize the benefits of accessibility and usability working together, and about resources to support your efforts.

Shawn Henry leads worldwide education and outreach promoting web accessibility for people with disabilities at the World Wide Web Consortium (W3C). Before joining the W3C Web Accessibility Initiative (WAI), she developed and implemented strategies to optimize user interface design for usability and accessibility with Fortune 500 companies, nonprofit organizations, education providers, and research centers. Shawn focuses her personal passion for accessibility on bringing together the needs of individuals and the goals of organizations in designing human-computer interfaces. Her book *Just Ask: Integrating Accessibility Throughout Design* offers an approach for developing products that are more usable for everyone.

Workshop: Introduction to Trusted Tester Methods

Andrew Nielson

New Editions Consulting, Inc.
anielson@neweditions.net

Kristen Smith-O'Connor

New Editions Consulting, Inc.
ksmith@neweditions.net

Ann Marie Davis

New Editions Consulting, Inc.
amcdavis@verizon.net

Abstract

This workshop introduces the DHS Trusted Tester 5.0 process and presents some primary methodologies for performing the manual testing process. The workshop also provides hands-on practice using the Accessible Name and Description Inspector (ANDI) to perform key parts of the Trusted Tester test process. Participants will receive information with a direct application to the Trusted Tester training curriculum and certification exam.

Trusted Tester 5.0 Background

The goal of the Trusted Tester program is to create and maintain a standardized approach for validating conformance against the Section 508 standards. A Working Group from the U.S. Federal Chief Information Officers' Council (CIOC), Accessibility Community of Practice (ACOP) revised the Trusted Tester process, to align with the Web Content Accessibility Guidelines (WCAG) 2.0 and the Revised Section 508 Standards. Trusted Tester 5.0 also improves the overall format, flow, and construction of the process and its test conditions to improve readability, coherence, and the overall effectiveness of the test process.

The Trusted Tester process is recognized as a leading practice among federal agencies in manual accessibility testing of information and communication technologies (ICT). The current version of Trusted Tester for Web and the Harmonized Process for Section 508 Testing are also both pending formal adoption by the ACOP.

Workshop Overview

The Trusted Tester 5.0 Basics workshop provides an opportunity for participants to jumpstart their education in manual accessibility testing for web content. The workshop is not intended as a replacement for the full Trusted Tester training curriculum. The workshop complements the Trusted Tester training curriculum by providing participants with an in-depth understanding of some of the core principles of the Trusted Tester test process along with hands-on application of the test process with real-life examples straight from existing websites.

The one-day workshop agenda will include:
Morning Session:

- Introduction to the Trusted Tester process – a primer on the format and organization of the Test Conditions and Evaluation Criteria
- Overview of ANDI
- Understanding core principles of Trusted Tester: Keyboard Access and Focus; Content Structure and Navigation
- Hands-on testing of websites
    - Review and discussion of test results/findings
    - Tips and tricks for overcoming common testing pitfalls

Afternoon Session:

- Understanding core principles of Trusted Tester: Forms; Links and Buttons; Images
- Hands-on testing
- General Q&A on testing principles
- Wrap-up of lessons learned and takeaways
- Wrap-up of lessons learned and takeaways for further use in the Trusted Tester certification training and exam

Facilitators' Background

Kristen, Andrew, and Ann Marie are three of the primary authors of the Trusted Tester 5.0 test process and the Trusted Tester 5.0 training curriculum. All have extensive experience in ICT Accessibility Testing and a deep understanding of the Trusted Tester process.

Copyright Notice

Workshop: Assessing and Authoring Image Descriptions for Digital Learning Products

Todd Marquis Boutin

SPi Global
Lewiston, ME, USA
todd.marquis.boutin@spi-global.com

Pina D’Intino

Aequum Global Access, Inc.
Pickering, Ontario, Canada
pina@aequumaccess.com

Abstract

Digital learning products rely on images to communicate concepts and support activities. The listening student must get the same benefit from the text alternatives (alt text) for these images. When assessing alt text, testers must evaluate the description based on the relationship between the image and associated lesson material (context) and the intended audience. To do so, they must understand the concepts and methodology that underpin alt text authoring. During this workshop, we introduce attendees to the theory and practice of alt text authoring. The workshop focusses on the following topics: (a) image context, audience, and purpose; (b) image type; (c) user preference and perspective; and (d) product- and process-related factors. Drawing on our experience with publishing and screen readers, we use real world examples to illustrate key concepts. Group discussion and mini-exercises support each topic-specific lesson, and attendees apply what they have learned by writing and reviewing descriptions for different images and contexts.

Workshop Overview

This full-day workshop consists of four topic-specific lessons followed by hands-on activities. Each topic-specific lesson consists of a presentation, group discussion, and mini-exercises. Hands-on activities include individual and small group practice in image evaluation, alt text authoring, and description review.

Topic-Specific Lessons

- **Image context, audience, and purpose**: We discuss how image purpose and description design change based on context and audience. Attendees practice evaluating images.

- **Image type**: We explore how images can be categorized into different types based on how they convey information. We then consider common type-based description templates.
- **User preference and perspective**: We review user data on alt text. Pina D'Intino shares her perspective as a screen reader user. Attendees listen to image descriptions via screen readers, compare their impressions, and practice revising those descriptions based on their findings.
- **Product- and process-related factors**: We outline the challenges facing large-scale alt text authoring and assessment, as well as possible solutions. Attendees share insights based on personal and professional experience.

Hands-On Activities

- **Image evaluation**: Attendees individually evaluate the context, audience, and purpose of images from varied learning products. They compare their findings in a group discussion.
- **Alt text assessment**: Attendees evaluate previously authored alt text for contextual appropriateness. They compare their findings in a group discussion.
- **Alt text authoring**: Attendees write descriptions for images from instructional and assessment material. We provide "walk-around" support as needed.
- **Peer workshopping**: Attendees review, compare, and discuss the descriptions written during the alt text authoring activity.

Learning Objectives

- Identify image purpose based on context, audience, image type, and other factors.
- Determine relevant image content and description design based on image purpose.
- Assess the contextual appropriateness of image descriptions.
- Practice authoring contextually appropriate alt text.

Copyright Notice

Workshop: Mobile Testing Workshop

Gian Wild
AccessibilityOz
Melbourne, Victoria, Australia
gian@accessibilityoz.com

Jennifer Chadwick

Siteimprove
Toronto, Ontario, Canada
jcha@siteimprove.com

Peter McNally
Bentley University User Experience Center
Waltham, Massachusetts, USA
pmcnally@bentley.edu

Members of the Mobile Testing Sub-Committee

Abstract

- Testing on mobile devices will be demonstrated, with opportunities for hands-on activities.
- Both mobile web and native apps will be covered, with iOS and Android operating systems.

Step 1: Identify what needs to be tested

Identify which devices, operating systems to cover and browsers to use. Recommended devices: **Apple iOS**: iPhone, iPad, Safari; **Android iOS**: Android phone, Android tablet, Chrome

Mobile product types

There are four types of mobile products and each type has different mobile testing requirements:

- **Desktop web sites:** have only one display, whether viewed on desktop, mobile or tablet.
- **Responsive web sites:** the display changes depending on the screen size or other features.
- **m.dot sites:** have a particular display for mobile and tablet sites. The m.dot site must also be tested against the entirety of WCAG 2.1, in addition to the web version.
- **Native Apps:** self-contained mobile applications that have been built usually oriented to completing a task (e.g., conducting banking transactions, ordering a product or service).

Step 2: Define essential functionality

For mobile sites and native apps, it is important to define the scope for testing and include the pages, features and functionality that allow the user to complete the tasks set out by the site or app, and if these were not tested or did not operate, would cause significant barriers to all users. This is especially important for native apps which are designed to be self-contained for a specific purpose (e.g. completing banking transactions).

Step 3: Conduct specific mobile tests

In addition to the familiar criteria that are tested on desktop – such as alternatives to images, and coding headings and tables – there are five more types of mobile testing errors:

a. **Critical mobile-specific interoperability**: hover trap, touch trap, zoom trap, etc.
b. **Mobile-specific interaction:** orientation, motion actuation, geolocation, etc.
c. **Mobile assistive technology support**: screen reader support, keyboard behavior, magnification / zoom behavior, inverse colors / grayscale behavior, etc.
d. **Mobile and desktop relationship errors:** consistency, restriction of content, etc.
e. Non-specific mobile issues common on mobile: alternatives for items only displayed in mobile (e.g. hamburger menus), underlined links, reference to attributes, etc.

Additional considerations for native apps

It's good to pay attention to the following elements, because they are the source of many defects:

- Focus placement on page load
- Focus for modal windows
- Custom-built elements (vs default components native to the app), including dropdowns, tooltips, etc. to ensure compatibility with accessibility settings
- Use of semantic elements (e.g. buttons, links, headings)

Copyright Notice

Workshop: Testing PDFs for Accessibility and Standards Conformance

Paul Rayius

CommonLook
1600 Wilson Blvd. Suite 1010, Arlington, VA 22209 USA
prayius@commonlook.com

Abstract

The goal for ITC testers who attend this hands-on workshop is an understanding of the technical requirements to ensure PDF documents meet the WCAG 2.0, WCAG 2.1, and PDF/UA standards, and an awareness of the tools available to test and ensure compliance (some of which are free). Attendees are taught how to read, interpret, and use the reports created by these programs.

Preparing for the Workshop

For this half-day workshop, we encourage attendees to bring their laptops with the following:

- Adobe Acrobat Professional (download a trial of Acrobat here) (https://acrobat.adobe.com/us/en/free-trial-download.html)
- PAC 3 (download PAC 3) (https://www.access-for-all.ch/en/pdf-lab/pdf-accessibility-checker-pac/downloading-pac.html)
- Commonlook Validator (download CommonLook Validator) (https://commonlook.com/pdf-validator/)
- Sample PDFs provided (sample files provided before the workshop)

Overview of the Presentation

This session explores the struggle to make PDFs accessible and how to test them to make sure that they comply with accessibility standards.

In Part 1, we'll very briefly touch on accessibility laws like Section 508 and the ADA, examine accessibility standards including WCAG 2.0, WCAG 2.1, and PDF/UA and consider how these laws and standards apply to PDFs. Using WCAG 2.1 as our standard, we'll test and review some sample PDFs with common accessibility problems like lists and tables. We'll answer the question: "What is required for these files to be compliant on a checkpoint by checkpoint basis?"

Also, we'll compare and contrast different versions of the same documents that are correctly and improperly tagged to highlight the differences.

In Part 2, we'll take a much closer look at testing PDFs for accessibility and standards compliance. Specifically, we'll look at what to consider when testing PDFs and how to know if your files are standards-compliant. Next, there is a detailed explanation of why some of the most commonly used PDF accessibility checkers fall short of the mark and fail to ensure 100% compliance with accessibility standards: an often-overlooked mistake of ICT testers. Using some of the sample documents from the first portion of the workshop, CommonLook demonstrates how documents can pass the Adobe Accessibility Checker, for example, but still have issues that must be resolved to pass WCAG 2.0 AA, WCAG 2.1 AA, or PDF/UA. Since many organizations are required to meet at least WCAG 2.0 AA (Revised Section 508), we show that (and why) using Adobe Acrobat Professional or DC alone is insufficient for testing and certifying PDF document compliance.

Next, we'll compare third-party PDF accessibility checkers including Adobe Acrobat DC, PAC-3, and the CommonLook Validator. We'll explain (and show) why these tools need to be in every ICT tester's toolkit. There is a discussion of tools and techniques to test PDF files in batches using popular automated HTML testing tools and dedicated tools for PDF accessibility testing such as CommonLook Clarity. A discussion of these tools and techniques demonstrates the benefits and drawbacks of each option and reveals what can and can't be done with automated tools. Even when using batch testing methods, automated testing methods, or both, manual verification is needed. We discuss techniques for testing documents for color/contrast and other accessibility issues that are not checked using automated testing and reporting tools. Finally, we review the reports from these tools and how ICT testers use the tools to review and audit websites for document accessibility compliance

ITC testers who attend this hands-on workshop will gain an understanding of the technical requirements to ensure PDF documents meet the WCAG 2.0, WCAG 2.1, and PDF/UA standards, and an awareness of the tools available to test and ensure compliance (some of which are free). Attendees will know how to read, interpret, and use the reports created by these programs.

Resource URLs

- WCAG 2.0 (https://www.w3.org/TR/WCAG20/)
- WCAG 2.1 (https://www.w3.org/TR/WCAG21/)
- PDF/UA (https://www.iso.org/obp/ui/#iso:std:iso:14289:-1:ed-2:v1:en)
- Mapping PDF/UA to WCAG 2.0 (https://www.aiim.org/Global/AIIM_Widgets/Community_Widgets/Achieving_WCAG)

Copyright Notice

Workshop: Finding Ways to Make Accessibility Resonate with Your Team: "Inclusive Design is not Rocket Science"

Kristina England

UMass Office of the President
333 South Street, Shrewsbury, MA 01545
kengland@umassp.edu

Tracy Axelson

UMass Office of the President
333 South Street, Shrewsbury, MA 01545
taxelson@umassp.edu

Kelsey Hall

Assistive Technology for Education, Accessibility Consultant
khall.consultant@gmail.com

Abstract

We often hear people groan when we mention making code or design accessible. Many designers and developers have preconceived notations such as "it's expensive" or "my creativity will be stifled." However, inclusive design is both less expensive over time and generates some amazing everyday technology, such as audio books, closed captions, and high contrast on mobile devices.

During this half-day workshop, you can anticipate learning the following information:

- Why inclusive design matters and how to make it resonate in your organization.
- What the Microsoft Inclusive Design Toolkit is and how to use it to build awareness.
- The key areas to focus on when designing an inclusive product or digital experience.
- The importance of manual testing over semi-automated testing, even when you have a good design system in place.

Full Description

We often hear people groan when we mention making code or design accessible. Many designers and developers have preconceived notations such as "it's expensive" or "my creativity will be stifled." However, inclusive design is both less expensive over time and generates some amazing everyday technology, such as audio books, closed captions, and high contrast on mobile devices.

During this session, we'll review key areas to focus on when building inclusive design into your design framework, including bypass blocks; color contrast and focus indicators; forms and error identification; collapsed content and modals; resizing text; descriptive links and images; and multimedia.

Leveraging Microsoft's Inclusive Design Toolkit

We'll also go over the Microsoft Inclusive Design Toolkit in detail. This toolkit provides a persona spectrum that will help you come up with stories that resonate for your audience. As part of this portion of the workshop, we'll explore products where inclusive design was "built in, not bolted on" and how that makes a huge difference for everyone. The goal of this session will be to provide you with enough materials and stories to bring back to your staff, whether your developers or your procurement office, to help foster awareness and adoption throughout your organization.

So how do you maintain an inclusive design structure in a small or large organization?

We'll give you some examples of great design system structures to consider internally when standing up your inclusive design procedures. Design systems allow you to define what developers and designs can use for button colors, focus indicators, hyperlinks, and more.

While design systems are a great way to provide structure, they should not be relied on solely for validation of inclusive products and content. We'll review the importance of manual testing throughout this session. While we won't go into detail on how to manually test, we'll talk about each persona and what tests are important to ensure the inclusiveness of your designs.

What You'll Gain from this Session

During this half-day workshop, you can anticipate learning the following information:

- Why inclusive design matters and how to make it resonate in your organization.
- What the Microsoft Inclusive Design Toolkit is and how to use it to build awareness.
- The key areas to focus on when designing an inclusive product or digital experience.
- The importance of manual testing over semi-automated testing, even when you have a good design system in place.

Copyright Notice

Bootcamp Course: Tools for Assessing Organizational Risk for Managers of Accessibility Testing Programs

Dr. Chris M. Law

Accessibility Track
Arlington, VA, USA
chrismlaw@accessibilitytrack.com

Pina D'Intino

Aequum Global Access
Pickering, ON, Canada
pina@aequumaccess.com

Abstract

Accessibility professionals are being called upon to assess levels of potential risk to organizations around digital accessibility. In recent years, more and more lawsuits have been filed against organizations that have failed to implement accessibility in their websites and electronic documents. The course provides actionable guidance for accessibility professionals which has been drawn from related fields of study (e.g., industrial safety) as well as consulting practice experience. The course is in four parts, covering (1) Sources of risk; (2) Measurement; (3) Operational issues; and (4) Implementation.

Background

In early 2018, members of the International Association of Accessibility Professionals (IAAP) Organizational Development Committee surveyed 205 people working in accessibility programs (Law et. al, 2018). Of the 115 respondents who worked in 'large' organizations, 81% cited 'Legal Compliance / Risk Management' as one of the top three drivers of their accessibility programs. This the number one cited factor, with 'Inclusion' coming in a distant second. This focus on legal risk in accessibility programs is understandable, given the steady increases in the number of lawsuits relating to the Americans with Disabilities Act in recent years (Vu et al, 2018).

This course was developed in response to the needs of accessibility (and legal) practitioners for guidance and tools for handling organizational risk as it relates to digital accessibility. The approach used in creating this course was to apply lessons from other fields (such as management, industrial safety, legal, and quality control) to the everyday aspects of how digital accessibility is handled within organizations. For example, a highly useful resources is 'The Legal Risk Management Handbook', Whalley & Guzelian (2017) which mostly deals with ethical and operational issues around legal risk. The content of Whalley & Guzelian's book is not directly related to web accessibility, so the concepts are interpreted and then applied to the topic of digital accessibility. We aim to expand the skills of those working in accessibility program management, in applying what they already know—for example, about accessibility testing of digital content—to new concepts such as quantifying risk, and assessing related organizational elements. In this one day boot camp style course we will provide practical exercises and guidance to the level where participants are able to begin systematically addressing risk when they return to their organization. Recognizing that we cannot cover all aspects of risk mitigation in just one day, we provide resource handouts summarizing and linking to further sources of guidance from the accessibility field, and from those other fields that have been embedding risk mitigation activities in their operations for many decades.

Brief course synopsis

Part 1 – Introduction to Risk, and Sources of Risk. We introduce risk concepts, and look at how and why risk assessment and mitigation has been studied and applied in other fields. We address the Precautionary Principle, as well as how human behavior and risk is often not a straightforward cause-and-effect relationship. We examine how to systematically identify sources of risk, and then how to articulate those risk sources in business and accessibility program terms.

Part 2 – Measuring Risk. Measurement of risk covers three main categories: (1) post-incident evaluation of what *did happen*, including seeking out root causes; (2) examining what, within reason, *could happen*; and (3) assessing *where you are now* in your organizational responses to the risks we identified earlier in Part 1. In the course we cover all three, but focus our time on the third category, since (1) most attendees will not have had an incident from which to draw from; and (2) knowing what could happen is only really achievable once you have a good grasp of the third category of where you are now. For this third category we conduct exercises with participants on how and when to gather interview information and other data from other parts of their organization in order to better understand where their risk points are, as well as impacts of those points on usability, accessibility, inclusion and overall user experience.

Part 3 – Legal Risk is Operational Risk. Once you know sources of risk in an organization, you need to begin thinking about how to address issues systematically. We examine proactive versus reactive approaches, the Law of Safety Progress, and look at why risk mitigation does not equal risk elimination. We use exercises to address common risk mitigation techniques.

Part 4 – Implementation: Planning for Change. People do not resist change; instead they resist ambiguity. When teams get together to decide on the current and the desired state, agreement is the easy part. Aspirations usually break down in the implementation, where fuzziness in task goals and divergences in self-interest opinions of staff are allowed to occur. Therefore, we wrap up the course with a look at how to systematically and preemptively handle resistance as groundwork for effective implementation, and how accessibility can be sustained through a maturity model and process.

References

Law, D'Intino, Romanowski & Sinclair (2018) **Priorities for the field: Management and Implementation of Testing within Accessibility Programs.** In proceedings of the 3rd Annual ICT Accessibility Testing Symposium, Arlington, VA. pp.141-153

Vu, Launey, & Ryan (2018). **Number Of Federal Website Accessibility Lawsuits Nearly Triple, Exceeding 2250 In 2018**. ADA Title III blog., January 31, 2019. Available: https://www.adatitleiii.com/2019/01/number-of-federal-website-accessibility-lawsuits-nearly-triple-exceeding-2250-in-2018/

Whalley & Guzelian (2017). **The Legal Risk Management Handbook.** Kogan Page, New York, NY.

Copyright Notice

Bootcamp Course: Accessibility Testing for Beginners

Gian Wild

AccessibilityOz
Melbourne, Victoria, Australia
gian@accessibilityoz.com

Abstract

This two-day course is being offered as part of a five-day package. The instructional course days takes place after participants have attended the main two-day symposium, and one of the one-day pre-conference workshops. The primary goal is to teach participants to test websites for conformance with industry standard requirements. The secondary goal is to provide participants with and an introduction to the 'World of Accessibility Testing' through their attendance of the main symposium.

Course Day 1

Morning: Accessibility testing methods and tools

The morning will be spent examining and using a variety of testing tools and techniques – with hands-on demonstrations including manual inspection methods, and spidering testing tools.

Afternoon: Everything you need to know about accessibility testing, Part A

The session will cover a whole range of testing requirements: starting with an overview of testing – who, when, what and how. Each topic will include an in-depth discussion and hands-on exercises for participants.

Part A topics will include:

- Choosing who will do the testing
- Choosing when to test – building a new site or testing an existing site
- Choosing what to test

Course Day 2

Morning: Mobile accessibility testing

Testing on mobile devices (iOS, Android) will be demonstrated with opportunities for hands-on activities. Mobile versions of websites will also be covered, some of which are also testable using PCs (laptops). Participants will learn what can be tested on a laptop and what needs to be tested on a device. This session will cover the accessibility of mobile web sites—responsive and m dot sites. Participants will receive a copy of the test process.

Afternoon: Everything you need to know about accessibility testing, Part B

This continues the previous day's session. Part B topics will include:

- Choosing how to test
- Choosing how to present findings
- Developing a scope analysis for a site (based on the W3C Evaluation Methodology)
- The day will end with a final debrief from the week, along with discussing arrangements for the follow-on activities with the instructor in the month that follows the event.

Follow-on activities

In the month following, the instructor will hold weekly catch-up meetings for participants to ask any questions arising from implementing these testing practices.

Participants will also receive a six-month subscription to AccessibilityOz' OzWiki: a database of accessibility errors, screenshots and solutions, and one month's access to AccessibilityOz' testing tool OzART.

Copyright Notice

W3C Accessibility Conformance Testing (ACT) Rules for WCAG

Shadi Abou-Zahra

W3C Web Accessibility Initiative (WAI)
2004, Route des Lucioles BP94, 06902 Sophia-Antipolis, France
shadi@w3.org

Abstract

The W3C Web Accessibility Initiative (WAI) is currently in the process of formally publishing a first set of Accessibility Conformance Testing (ACT) Rules for the Web Content Accessibility Guidelines (WCAG) 2. These ACT Rules are publicly documented checks for web accessibility, to help improve consistency and transparency among automated testing tools and manual testing methodologies. This paper introduces this work of the W3C Web Accessibility Initiative (WAI) on internationally harmonized ACT Rules.

Structure of ACT Rules

The structure of an ACT Rule is defined by the ACT Rules Format 1.0 specification (Fiers et. al., 2019). ACT Rules consist of the following parts:

- **Descriptive Title** – title for the ACT Rule, such as “Buttons have an accessible name”
- **Rule Identifier** – identifier for the ACT Rule (within a ruleset), such as HTML/Button-1
- **Rule Type** – there are two basic types of ACT Rules, depending on what is being tested:
    - **Atomic Rule** – test one specific situation, which may be part of a composite rule
    - **Composite Rule** – combine outcome from multiple atomic rules to one outcome
- **Accessibility Requirements Mapping** – maps the ACT Rule to particular accessibility requirements, such as Web Content Accessibility Guidelines (WCAG) 2 Success Criteria
- **Rule Input** – describes the scope of input into ACT Rules, which is one of the following:
    - **Input Aspects** – input into atomic rules, such as DOM Tree and CSS Styling etc.
    - **Input Rules** – input into the composite rules, which are the atomic rules in scope
- **Applicability** – description of the specific parts of the content, for which the rule applies
- **Expectations** – description of the expected characteristics of the applicable rule content
- **Assumptions** – assumptions made, such as specific interpretations of the requirements
- **Accessibility Support** – known limitations regarding browsers and assistive technology
- **Test Cases** – sample code demonstrating passed, failed, and inapplicable rule conditions
- **Change Log** – history of changes for the ACT Rules, to support backward compatibility
- **Glossary** – list of key terms defined by the ACT Rule or used by the specific ACT Rule
- **Issues List (Optional)** – list of known issues or bugs for the particular ACT Rule, if any
- **Background (Optional)** – relevant background, such as additional documentation, if any
- **Acknowledgements (Optional)** – such as rule writers, reviewers, and other contributors

Descriptive Title

Every rule needs to have a descriptive title. The specification does constrain the style and format of such titles because different organizations have varying practices. It is, however, a consensus within the W3C ACT Rules Community Group to use 'positive formulation'. For example, titles should read:

> Buttons have an accessible name ✓

Rather than:

> Buttons do not have an accessible name ✗

Another convention for writing titles for ACT Rules is to avoid mentioning specific technologies except if the rule is actually confined to particular technologies. For example, the rule above may apply to both HTML and WAI-ARIA, which is why preferably no technology is mentioned.

Rule Identifier

To simplify automated processing of ACT Rules, and especially of results from testing, every rules must have an identifier. An identifier is a short handle. It could be a database reference, a web address (URI), or any such identifier for a rule in a particular context.

The specification does not further constrain the format of identifiers because there can be many different contexts within organizations and testing tools. Yet one requirement is that identifiers must be unique within a set (logical collection) of ACT Rules. For example, if an organization publishes a ruleset, each rule within that set must have a uniquely distinct identifier.

Rule Type

Essentially, there are two types of ACT Rules:

- **Atomic Rule** – test one specific situation, which may be part of a composite rule
- **Composite Rule** – combine outcome from multiple atomic rules to one outcome

The underlying intention is to encourage rule authors to break down complex checks into smaller sub-checks (atomic rules), which are combined together in a transparent way (composite rules).

Atomic Rules

Atomic rules test for one specific situation, such as if page components identified as buttons have an accessible name. Many accessibility checks can be formulated using atomic rules. However, some accessibility requirements have 'or'-conditions or exceptions, and cannot be formulated in simple atomic rules. For example, WCAG Success Criterion 2.1.2 requires that navigation on a web page can be carried out with standard keys or that the user is informed about the keys they need to use for navigation, to avoid 'keyboard traps'. This requires that each condition, 'standard keys' and 'user informed' to be formulated in separate atomic rules, and to formulate the overall check in a third composite rule that utilizes these two atomic rules.

Composite Rule

Composite rules do not reflect actual checks as atomic rules do. Instead, they combine the results from a set of atomic rules in a particular logic to reflect a higher-level accessibility check. These combinations could be more complex than the previously described check for 'keyboard traps'. For example, a rule that checks if pre-recorded video has accessible alternatives (WCAG Success Criterion 1.2.1) checks several aspects using atomic rules, of which at least one needs to be true:

- The video is not an alternative for other content
- The video has an audio description track
- The video has a transcript accompanying it
- The video is accompanied by another alternative

The specification does not allow the cascading of rules in further levels, such as composite rules within other rules. Such cascades would potentially start reflecting tool-specific architectures for checking rather than to remain agnostic to specific tools and methodologies.

Accessibility Requirements Mapping

Each rule must describe how it relates to specific accessibility requirements, such as to WCAG. The ACT Rules Format 1.0 specification has been designed to address WCAG testing but is not limited to WCAG. For example, the specification can be used to write rules for other standards, such as company-internal guidelines, contractual obligations, and local policies that may define particular accessibility requirements. This also allows the specification to address future versions of WCAG being explored in the W3C Accessibility Guidelines Working Group (AGWG).

In some cases, when atomic rules are designed for used within composite rules, they may not be checking for any particular accessibility requirement. For example, checking for 'standard keys' and 'user informed' (as described above) are in themselves no accessibility checks. Only when these are combined to check for 'keyboard traps', does the composite rule check accessibility.

When an ACT Rule maps to an accessibility requirement and the outcome of running that rule on a piece of content is 'failed', then it **must** mean that this content does not meet the accessibility requirement. However, for the outcome is 'pass' the ACT Rule must define if that means that the content meets the accessibility requirement or whether further testing is needed to determine that.

Rule Input

Rule input describes what the rule consumes to calculate an outcome. This is different for atomic rules that operate directly on the content and for composite rules that operate on other rules.

Input Aspects

For atomic rules, the 'input' could by roughly described as the type of content that the rule is run on. For example, this could be the Hyper-Text Markup Language (HTML), the Cascading Style Sheets (CSS), the Document Object Model (DOM), or combinations of these and other content technologies. Rules must only process content from the specified input aspects.

Input Rules

For composite rules, the 'input' are the atomic rules that are used to calculate an outcome. More specifically, the 'input' is actually the 'outcome' from running every specified input on the same 'test subject'. These outcome results are then combined according to a logic specified in the rule, to calculate an overall 'outcome' for the composite rule itself.

Applicability

The applicability is, together with the expectations, the heart piece of ACT Rules. Applicability defines the exact parts of the content that apply to the rule. For example, for the above mentioned ACT Rule "Buttons have an accessible name", the term 'button' is ambiguous as it is defined by both HTML and CSS. To address this issue, the applicability section for this rule reads:

> The rule applies to elements that are included in the accessibility tree with the semantic role of `button`, except for `input` elements of `type="image"`.

In this, exact definitions for 'included in the accessibility tree' and 'sematic role' are linked from the description to avoid ambiguity. Such a specific definition for the applicability allows readers to better understand what the rule is specifically testing and what it is not, to ensure transparency.

Expectations

Similar to applicability, also expectations can be ambiguous if they are not described in more detail. Continuing with the above-mentioned ACT Rule "Buttons have an accessible name", also the phrase 'have an accessible name' is ambiguous if it is not described more specifically.

The expectations section for this rule therefore reads:

> Each target element has an accessible name that is not only whitespace.

The term 'target element' refers to the selection defined by the applicability section. Definitions are provided for the terms 'accessible name' and 'whitespaces', to avoid ambiguity of the phrase.

When the expectations is met for every target element, the outcome of the rule is 'passed'. If the expectations is not met for any target element, the outcome of the rule is 'failed'. If there are no target elements, then the outcome of the rule is 'not applicable'.

Assumptions

In some cases, assumptions need to be made, in which case they must be adequately documented for transparency. For example, for the above mentioned ACT Rule "Buttons have an accessible name", an assumption is made that 'buttons' are considered 'user interface components':

> The rule assumes that all buttons are user interface components as defined by WCAG 2.

This is because WCAG does not actually state that buttons must have accessible names. It does, however, state that all user interface components must have a defined 'role', 'name', and 'value'. Only under the assumption that buttons are user interface components would they then need to have a ‘name’. In some cases, this may seem as stating the obvious but many differences in web accessibility testing occur from different assumptions being made without proper documentation.

Accessibility Support

Also, part of proper documentation is to record any known accessibility support aspects related to the rule. For example, technically using the ‘title’ attribute is one way of providing an accessible name to a user interface component, such as to a form field. Given how the ACT Rule is written, the calculation for accessible name will pass content using this technique to identify form fields. However, some assistive technologies do not (correctly) process the ‘title’ attribute so that this technique is actually not suitable for them. This section is to record such caveats of a rule.

Test Cases

One of the most practical sections of an ACT Rule is its test cases. These include examples of:

- Code that is expected to produce ‘passed’ outcome
- Code that is expected to produce ‘failed’ outcome
- Code that is expected to produce ‘not applicable’ outcome

This has three primary purposes:

1. Help the rule author write accurate rules (test-driven authoring)
2. Help the reader understand what the rule does in practical terms
3. Help implementers of the rule to validate their implementations

While the ACT Rules Format 1.0 specification does not define any specific requirements for how many test case a rule must have, it is generally understood that the more (unique) test cases there are, the better it will be for quality assurance of the rule itself as well as for any implementations.

The ACT Rules Community Group provides a test runner that outputs a JSON file with all test cases, to help implementers run their tools and methodologies against these test cases.

Change Log

Another important aspect of quality assurance is versioning each rule, and documenting changes from one version to another. The changelog section of an ACT Rule captures these changes that were made during the lifetime of the rule.

Glossary

All key terms and phrases used in an ACT Rule must be defined in the glossary. Sometimes they are specific to the particular rule, and sometimes they are used across several rules. For example, the definition for ‘whitespace’ is used multiple times across different rules. Where possible, ACT Rules should refer to existing definitions, such as for ‘user interface components’ that is defined by WCAG or for ‘semantic role’ that is defined by WAI-ARIA.

Issues List (Optional)

Despite best efforts, issues can always occur. These could be minor typos, broken links, or more substantive issues. Such identified issues can be logged in the issues list, for example while the rule is being revised to address issues. That is, ideally all issues will eventually be moved to the changelog section, assuming that the rule is being actively maintained.

Background (Optional)

Often rules relate to existing documentation, such as entries in the Techniques for WCAG 2 or in the Understanding WCAG 2 documentation. They could also relate to references on accessibility support or other materials to help readers and implementers better understand the background of the rule, the intent, and rationale for taking certain decisions.

Acknowledgements (Optional)

Finally, ACT Rules also have a section to credit relevant contributors. This could be a funding entity, such as the European Commission (EC) that has massively contributed to this first set of ACT Rules, relevant organizations and donors, and individuals if needed. This section could also be used to define certain licensing or terms of use for ACT Rules entities own or contribute.

Creation of ACT Rules

Based on ACT Rules Format 1.0 specification (Fiers et. al., 2019) different entities can document ACT Rules. The W3C consensus process is used to review rules proposed from different sources and gather broader support and acceptance of the proposed rules within the community. Formally published ACT Rules are then implemented into the different automated testing tools and manual testing methodologies. Ultimately, this leads to consistent test results and transparent evaluation.

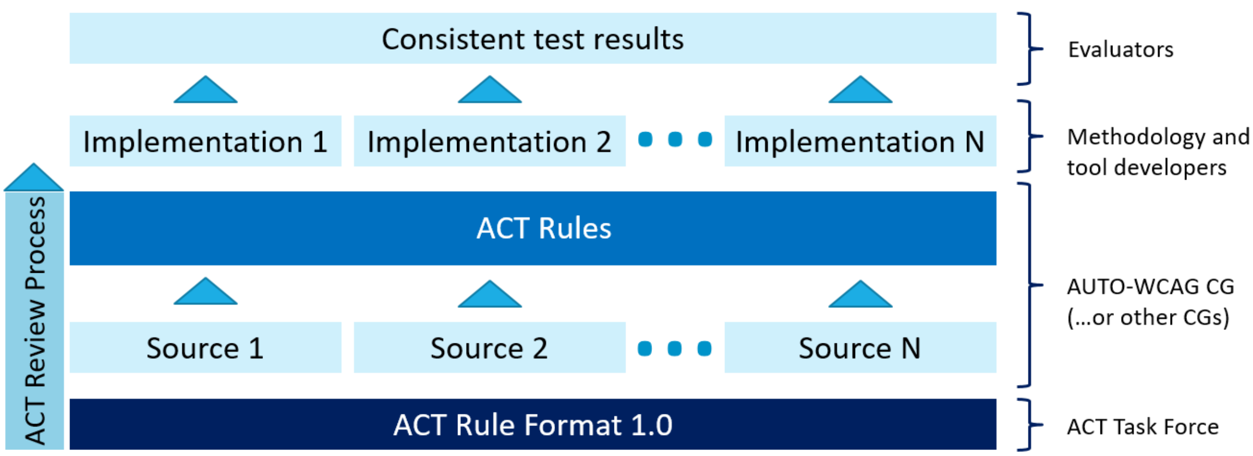

Figure 1: Components of Accessibility Conformance Testing (ACT)

Status of ACT Rules

Currently, ACT Rules are mainly developed by the W3C ACT Rules Community Group (W3C ACT-R CG, 2019). The work is carried out openly on the GitHub development platform:

https://act-rules.github.io/

Anyone can participate in and contribute to this effort. Currently, 56 participants from different organizations and businesses are involved in this work. New participants are joining this effort as we transition from the formerly Auto-WCAG Community Group to this newly branded group.

To date, this group published 47 rules with several more in the development pipeline. These 47 rules were each reviewed by at least three independent reviewers, implemented in at least two automated testing tools or manual testing methodologies, and validated on a set of real websites. That is, they are considered fairly mature and stable, and ready to be submitted to the W3C/WAI Accessibility Guidelines Working Group (W3C AGWG, 2019), which is the group that publishes WCAG and is the authoritative body to interpret WCAG. ACT Rules approved and published by AGWG are expected to become authoritative references for developers of testing tools and methodologies, similarly to how Techniques for WCAG 2 and Understanding WCAG 2 support developers of web content in understanding and implementing the accessibility requirements.

The expectation of the ACT Rules Community Group is to publish a set of 55 rules by October 31st, 2019, and to submit a first subset of rules to AGWG for formal W3C publication. Currently the ACT Rules Community Group has active participation from at least 5 automated testing tool developers, at least 4 manual methodology developers, and at least 3 corporations with internal design and testing guidelines. As the group continues to publish rules and succeeds in achieving formally published ACT Rules by AGWG, we expect this participation to continue to increase.

Conclusion

The W3C work on Accessibility Conformance Testing (ACT) was launched in 2016. It is based on prior work dating back to 2001. However, this recent work includes active involvement from commercial tool vendors and from public bodies who were not as involved in prior work. The work can be considered in three complementary aspects:

1. W3C standard to define how ACT Rules are written, to allow different organizations to share their own rules. This standard is expected to be completed in October 2019.
2. W3C community group that is open to anyone, regardless of W3C membership status, to support organization in sharing, reviewing, and agreeing on a common set of rules.
3. A controlled process through which this community group and other organizations can submit candidate rules to the W3C working group for formal recognition and publication.

This work is progressing well. The standard has been maturing according to plan and is now in the final stages of the W3C standardization process. Also, an increasing set of candidate rules has been developed by the ACT Rules Community Group, and an initial subset of these rules is currently being submitted to W3C to test-drive and to refine the publication process. Part of this process is to demonstrate a minimal set of at least three independent implementations of the rules before they are published. That is, these rules are not merely theory but being actually adopted by different tool vendors. We continue to expect that the ACT Rules Format 1.0 standard and an initial set of rules, including the corresponding implementations, by October 2019.

Acknowledgements

This work on ACT and this presentation are supported by the WAI-Tools Project, co-funded by the European Commission (EC) under the Horizon 2020 Program (Grant Number GA-780057).

References

Fiers, W., Kraft, M., Mueller, M.J., Abou-Zahra, S. (2019). Accessibility Conformance Testing (ACT) Rules Format 1.0. W3C Proposed Recommendation, 30 July 2019. Retrieved on 30 August 2019. https://www.w3.org/TR/act-rules-format/

W3C ACT-R CG (2019). Accessibility Conformance Testing (ACT) Rules Community Group (ACT-R CG). Community Group of the World Wide Web Consortium (W3C). Retrieved on 30 August 2019. https://www.w3.org/community/act-r/

W3C AGWG (2019). Accessibility Guidelines Working Group (AGWG). Web Accessibility Initiative (WAI) Working Group of the World Wide Web Consortium (W3C). Retrieved on 30 August 2019. https://www.w3.org/GL/

Copyright Notice

Accessibility Testing: Mileage May Vary

Karen Herr

Digital Accessibility Engineer
Optum
11000 Optum Cir, Eden Prairie, MN 55344, USA
Karen.herr@optum.com/Karen.herr.a11y@gmail.com

Sean Kelly

Digital Accessibility Engineer
Optum
11000 Optum Cir, Eden Prairie, MN 55344, USA
sean_kelly@optum.com/sean.kelly.a11y@gmail.com

Abstract

When accessibility audits of the same digital property result in differing feedback, our stakeholders have questions. How does this happen, what are we missing, and how can we assure our stakeholders of the validity of our findings?

Replication Difficulties

> "No man ever steps in the same river twice, for it's not the same river and he's not the same man."— *Heraclitus (544 BCE)*

It's very difficult to exactly replicate a testing situation. Was new code in the digital product introduced? Did Apple release a new iOS operating system version? The digital product version, operating system, browser version, screen reader version, other accessibility tools, and the tester would need to be exactly the same – and even then exact replication is not guaranteed.

Browser/Assistive Technology (BrAT) Combinations

With dozens of Browser + Assistive Technology combinations it's not financially feasible to test every browser and assistive technology combination. As such, we test a small subset of combinations and acknowledge there could be some gaps in results. Like anywhere else in accessibility, weighing risk versus available resources is a constant balancing act.

Limits to Programmatically Discoverable Issues

It's 2019, and the push to automate all testing is pervasive. However, only approximately 25% of accessibility errors can be found with automation tools (Groves, 2011). The rest are found through manual testing and testing with assistive technologies.

Even within those programmatically discoverable issues, one can easily find places where the browser's rendering engine has done different things to fix questionable HTML and that the same testing tool might give different results in different browsers starting with the same code.

Human Factors

Many of the WCAG checkpoints are subjective. As human beings testing software, we have different experience levels in different areas. Often, full-time screen reader users conducting testing will find issues that others missed, due to their depth of expertise with assistive technology.

Even if two different testers identify the same issue, their suggested approaches for resolution may vary greatly. Our own challenges as testers also tend to color our findings—one tends to prioritize issues that are personally difficult.

Accessibility Testing as Traditional Software Testing

Everyone wants defect-free software. While that is a noble goal, it's not entirely feasible. One of the seven principles of software testing is that "Exhaustive Testing is Impossible", except for trivial cases (Graham, Van Veenedaala, Evans, Butman, & Black, 2008). Constraints on time, budget, and resources are always present, so we attempt to prioritize our testing approaches.

The Pareto Principle (the law of the vital few) is better known as the 80/20 rule. Vilfredo Pareto's initial application of the principle indicated that 80% of wealth comes from 20% of the population.

Dr. Joseph Juran extended Pareto's principle to quality (Butman, 1997). In software testing, it is suggested that 80% of all bugs can be found in 20% of program modules. Inversely, 20% of all bugs can be found in 80% of program modules. The return on investment for testing all program modules is not typically deemed high enough to test everything.

Accessibility Testing as Usability Testing

Usability Testing

Rich accessibility testing includes visual and code inspections, keyboard testing, and testing with assistive technology. It also includes task-based scenarios, or usability testing. Tom Landauer and Jakob Nielson determined that the number of usability problems found in a usability test with n users is:

$$N = (1-(1-L)^{n})$$

where *N* is the total number of usability problems in the design and *L* is the proportion of usability problems discovered by a single tester. The typical value of *L* is 31%, averaged across a large number of projects we studied (Nielson, 2000).

Applying this formula, one accessibility tester performing task-based scenarios would find approximately 31% of the usability problems. Two accessibility testers would find 52%, and three would find 67% of the usability problems. The curve continues to show us that to find 100% of the usability problems, we would need 15 testers.

User Experience Issues Found While Accessibility Testing

While accessibility testing, we may encounter a "user experience" issue that is not mappable to WCAG, but nonetheless provides an experiential barrier for a user.

Unlike traditional software testing that has no formal body of "rules," we can be over-focused on the WCAG success criteria or ensuing checkpoints and explicitly called out examples of defects. We are frequently in the position of working with others who might be interested in a bare minimum "letter of the law" approach rather than understanding the need to create an experience that is truly usable by all and adequately takes into account what makes an experience work better.

Examples

A search result page with 1000 results that continue down the same page. Though not uncommon, this is not a particularly efficient approach. Visual users can scan the results and hope to find what they are looking for. Screen reader users would need to tab or swipe through each result. Users with other cognitive or perceptual issues would often have difficulty parsing such a wave of results. Nothing in WCAG explicitly requires pagination or filter options, yet we add them for user experience purposes.

Similarly, a "mega menu" with dozens of items creates a barrier for both assistive technology users and anyone else for whom the extra cognitive load is a problem, as well as users of zooming software.

Typographical issues such as text that is all caps, text that is justified, or text that is small can present issues for individuals with dyslexia or low vision.

Testing Levels

Meets the standard (minimum)

When we talk about something meeting the standard, we generally mean it meets the WCAG Success Criteria (World Wide Web Consortium (W3C), 2018). In a pass/fail situation, does it pass, or does it fail?

Best Practice

A standard that is considered a "best practice" would be one that is a step above. Not only does it meet the WCAG Success Criteria, but it offers a better user experience.

Best in Class

To be considered “best in class”, a solution would not only meet the WCAG Success Criteria but would provide the best user experience possible. Accessibility considerations are part of the conceptualization from the earliest design phases. Every user interaction includes the most holistically accessible approach from the beginning.

Examples of Testing Levels

Meets the Standard	**Best Practice**	**Best in Class**
A form has a disabled Submit button that becomes enabled when the user enters required fields.	Disabled Submit buttons don’t offer a great user experience because they are skipped in the tab order. A workaround for this is to a) use CSS to make the button appear as disabled, b) use aria-disabled=”true” instead of the “disabled” attribute, so it is announced as disabled to assistive technology, and c) JavaScript in place to ensure the button doesn’t submit when “disabled”	Don’t use disabled Submit buttons. Disabled Submit buttons force your users to figure out what magical combination of fields must be completed in order to Submit. Instead, use an active Submit button and tell the users with error messages which fields need to be provided or fixed.
Pre 2.1: No recommendation for color contrast of controls	3.0:1 contrast of controls to background.	3.0:1 contrast of controls in hover or focus state against the natural state. Think of a menu where
A page is organized using descriptive headings	Each page has one top-level heading.	The <title> of the page is congruent to the top-level heading.
AA - No requirements for touch target size	AAA - Minimum target size	Minimum spacing between targets.
A page is organized using descriptive headings	The heading levels are strictly hierarchical	

Meets the Standard	Best Practice	Best in Class
SC 2.4.4 Multiple “Read More” links are sufficient in and of themselves if the surrounding text is descriptive.	The “Read More” link uses aria-describedby to add more detail to the “Read More” text. For example “Read More about Accessibility”.	Multiple “save” buttons are provided for changing different information of a user profile. The button text is differentiated by “save name” “save address” (SC 2.4.4 is specifically written for links.)

Categorization of Issues

Under which performance criteria do you log issues? Depending on the situation and the person doing the assessment, the feedback may vary.

Example 1: A button with an image inside.

```
<button id=”myButton”><img alt=””src=”picture.gif”></button>
```

Error: Buttons must have discernible text.

- This could be under 1.1.1 Non-text content.

  Solution: `<button id=”myButton”><img alt=”Description of image” src=”picture.gif”></button>`

- It could also be under 4.1.2 Name, Role, Value because the button does not have a name.

  Solution: `<button id=”myButton” aria-label=”Description of image”><img alt=”” src=”picture.gif”></button>`

Example 2: An image with text that is used as a button. The image has “GO” as the text, but “Search” as the image’s alt text.

```
<input name=”btnG” type=”image” src=”images/go.gif”
alt=”Search” border=”0”>
```

Error: None found with automation

- This could be under 1.1.1 Non-text content

  Solution: `<input name=”btnG” type=”image” src=”images/go.gif” alt=”Go” border=”0”>`

- Or it could be under 1.4.5 Images of Text

Solution: <button border=”0”>GO</button>

- It could also be under 2.5.3 Label in Name because the visual text does not match the accessible name.

  Solution: <input name=”btnG” type=”image” src=”images/go.gif” alt=”Go” border=”0”> or <button border=”0”>GO</button>

Accessibility as a “Wicked Problem”

Looking at managing these aspects of accessibility brings to mind Horst Rittel’s concept of a “Wicked problem” (Rittel & Webber, 1973).

> “A wicked problem is a social or cultural problem that is difficult or impossible to solve for as many as four reasons: incomplete or contradictory knowledge, the number of people and opinions involved, the large economic burden, and the interconnected nature of these problems with other problems. Poverty is linked with education, nutrition with poverty, the economy with nutrition, and so on.”

Difficulties with achieving “perfect” accessibility satisfy several of the criteria that Rittel and others describe (Kolko, 2012). For example:

- Solutions to wicked problems can be only good or bad, not true or false. There is no idealized end state to arrive at, and so approaches to wicked problems should be tractable ways to improve a situation rather than solve it.
  - Accessibility is a journey not a destination.
- Every wicked problem is unique.
  - While many Accessibility problems follow particular patterns, there are usually unique aspects to the combinations of issues
- Every wicked problem is a symptom of another problem. The interconnected quality of socio-economic political systems illustrates how, for example, a change in education will cause new behavior in nutrition.
  - Accessibility issues can be symptoms of a lack of accessibility education, budget, and release schedules.

Mitigation Techniques

Unified descriptions and examples

Describe problems and resolutions with common language and specificity

Describe the issues and resolutions using commonly agreed upon descriptions and nomenclature from a common shared source.

Knowledge Database/Example Library

The shared source where issue and resolution descriptions reside with corresponding examples. All the descriptions are available as pre-formed text to use in evaluation documents. Design and code examples are available where relevant and useful.

We know that there are many solutions and don't want to limit solutions to a single option.

The goal is not necessarily to cut and paste solutions but to have flexible examples with wide applicability to cover a number of nuanced options.

Agreed-Upon Level of Acceptance (minimum, best practice, best in class)

As mentioned before, meeting the success criteria doesn't guarantee a good user experience. Your team members should be aware of the minimum criteria as well as the standard level of assessment. At Optum, we have several items that are considered "best practice," are above the minimum success criteria, and are expected in our ICT. It's not unusual to recommend remediation of an AAA success criteria, such as poorly implemented abbreviations. If we recommend remediation of items outside our baseline, we're sure to state that "we recommend for best user experience" a solution, but don't hold the teams to that recommendation.

Peer Collaboration

Pair Testing

Pair testing is two people testing a scenario together: sharing insights, asking questions, making suggestions, and reaching consensus (Azeri, 2018). With pair testing, more bugs and more consistent results are found. Improved communication and relationships are built within the team, and learning opportunities are provided. While assigning two engineers to perform testing costs more, the improvement of results can justify the expense.

At Optum, we take advantage of a natural opportunity for pair testing when we already have an engineer working as a visual assistant for another engineer that has a visual disability. We've found that this collaboration is especially beneficial when pairing an experienced engineer with a newer engineer.

Peer Review

On one of our high-profile projects, we have an "Accessibility Round Table." The members of the round table meet regularly to peer review each other's findings. This provides opportunities to share new ideas, methods of testing, and to verify the consistency and accuracy of results. This process is similar to a code review found in software development.

Training and Certifications

Our personal experiences shape our accessibility testing; if you come from test or development background, your approach and solutioning may vary. As such, accessibility testers should have a baseline of knowledge. Ideal ways to ensure your testers are operating from the same baseline is to encourage certifications. The 508 Trusted Tester certification is a solid testing methodology and also ensures your testers are well-versed in the WCAG 2.0 success criteria. Alternatively, the IAAP WAS certification provides a good measure of your tester's knowledge of WCAG, ARIA, HTML, and CSS.

Summary

A wide range of reasons for variation in testing results exists. Practitioners in the development and testing processes are more effective if they understand, anticipate, and can articulate why test results differ. Applying techniques for mitigation can alleviate many of these differences, but still, differences will exist. In order to recognize and plan for these inevitable discrepancies, expectations should be managed with all stakeholders.

References

Azeri, I. (2018, 06 20). *The Benefits and Challenges of Pair Programming and Pair Testing.* Retrieved 08 27, 2019, from mabl: https://www.mabl.com/blog/benefits-challenges-pair-programming-and-testing

Butman, J. (1997). *Juran: A Lifetime of Influence*. Hoboken: John Wiley & Sons, Inc.

Graham, D., Van Veenedaala, E., Evans, E., Butman, J., & Black, R. (2008). *Foundations of Software Testing: ISTQB Certification*. London: Cengage Learning EMEA.

Groves, K. (2011, 06 12). *The Problem with Automated Website Accessibility Testing Tools*. Retrieved 08 27, 2019, from Karl Groves: https://karlgroves.com/2011/06/12/the-problem-with-automated-testing-tools

Kolko, J. (2012, 03 1). *Wicked Problems: Problems Worth Solving.* Retrieved 08 27, 2019, from Austin Center for Design: https://www.wickedproblems.com/

Nielson, J. (2000, 03 18). *Why You Only Need to Test with 5 Users*. Retrieved 08 27, 2019, from NN/g Nielsen Norman Group: https://www.nngroup.com/articles/why-you-only-need-to-test-with-5-users/

Rittel, H., & Webber, M. (1973). Integrating Knowledge and Practice to Advance Human Dignity. *Policy Sciences,* 155. Retrieved from https://doi.org/10.1007/BF01405730

World Wide Web Consortium (W3C). (2018, 06 5). *Web Content Accessibility Guidelines (WCAG) 2.1*. Retrieved 08 27, 2019, from w3c: https://www.w3.org/TR/WCAG21/

Copyright Notice

Creating an Accessibility Testing Program in an Academic Research Setting

Jocelyn Bojorquez
jocelynbojorquez@suu.edu
Karrie Eliason
karrieannsavage@suu.edu
Erin Flores
erinweekes@suu.edu
Teigen Judd
juddteigen@suu.edu
Ashley Tyler
ashleystubbs@suu.edu
James Pollard
Jamespollard1@suu.edu
Southern Utah University
351 W University Blvd, Cedar City, UT 84720, USA

Gretchen Andrew
University of Nevada, Las Vegas
4505 S Maryland Pkwy, Las Vegas, NV 89154, USA
Gretchen.andrew@unlv.edu

Stacey Martinez
Forest Service- Chief Information Office
1617 Cole Blvd., Bldg. 17, Lakewood, CO 80401, USA
stacey.a.martinez@usda.gov

Abstract

As an academic institution engaged in a partnership agreement with the Forest Inventory and Analysis (FIA) program and the University of Nevada, Las Vegas (UNLV), Southern Utah University (SUU) was tasked with making a government computer application Section 508 compliant. The challenges that SUU experienced were similar to those of other organizations, such as the limited awareness of accessibility issues, lack of an established testing framework, ambiguity of guidelines, and lack of Section 508 training and education. While attempting to overcome these challenges and fulfill their mandate, SUU created an effective Section 508 team and reflected on the process they used to create an accessibility testing program. The process reported herein demonstrates that even with a lack of expert Section 508 knowledge, or a group of developers and testers who have been specifically trained in Section 508 issues, an effective Section 508 testing plan can be created.

Introduction

Accessibility allows people with disabilities to have equal access to products, locations, resources, and technology. As research into accessible technology has progressed, the definition of disability has been refined. Currently, there are five types of disabilities recognized by the United Nations Convention and other accessibility focus groups (Foley & Regan, 2002; Hasan et al., 2017):

1. Visual impairments
2. Hearing impairments
3. Physical and motor impairments
4. Cognitive impairments
5. Photo-epilepsy sensitivity

Accessibility is important for social, economic, and legal reasons. Socially, people with disabilities are a growing minority; since 1997, the U.S. population of people who have some form of disability has increased from 51.8 million to 56.7 million people (Foley & Regan, 2002; Jaeger and Matteson, 2009; Onyeabor, 2016). Economically, while many people with disabilities are financially contributing-members of society, they are still greatly underrepresented in the Science, Technology, Engineering, and Mathematic (STEM) fields (Burgstahler & Doyle, 2005). Estimates of the disposable income of people with disabilities have exceeded 175 billion dollars (Loiacono, Djamasbi, & Kiryazov, 2013).

As the disabled population increases over time, the need to provide technological solutions to them has also been increasingly recognized in legislation. For example, Section 508 of the Rehabilitation Act of 1973 (29 U.S.C. § 794d) required that all electronic and information technology (EITs) created with federal funds must be accessible for users with disabilities. This law provides specific criteria that a federally funded technology needs to abide by in order to be compliant. Along with the legislative mandates that have emerged, there have also been a concomitant rise in accessibility related lawsuits (Launey, Vu & Ryan, 2018). These types of lawsuits have affected private companies, academic institutions, and government entities. For example, private companies were sued by potential job applicants for having inaccessible online forms on their website (Russell, 2019; Smith, 2018). This form of legislation activity has prompted many entities to prioritize conforming to accessibility standards in their settings.

The research presented in this paper originated from a government funded project designed to provide a variety of users with an analysis tool for national forest data. The Forest Inventory and Analysis (FIA) program entered a partnership with Southern Utah University (SUU) and the University of Nevada, Las Vegas (UNLV). This team was tasked with testing and demonstrating Section 508 compliance for the Design and Analysis Toolkit for Inventory and Monitoring (DATIM) application. There were many challenges that were identified during DATIM testing, which will be described below along with insights which may serve as advisory techniques for other accessibility teams.

Literature Review

Limited Awareness

Studies for accessibility and Section 508 compliance have been conducted by a variety of organizations including universities and government entities to assess the awareness and investment levels for accessibility solutions in their fields. For example, the Employment and

Disability Institute at Cornell University conducted a study of 30 community college websites for Section 508 compliance via automated testing, manual testing, simulations, and target group usability testing. All of the websites evaluated for Section 508 compliance "performed poorly on accessibility evaluations and usability testing" (Bruyère, 2008). Bruyère (2008) also found that the contributing barriers to creating accessible web sites consisted mainly of 1) lack of awareness about the need for web accessibility, 2) the cost and time involved, and 3) the lack of knowledge about what is required to make websites accessible. It is difficult to enter a field of work without prior awareness. A government study found that the key obstacle to creating accessible products was finding people who understood and could identify the barriers to accessibility in design, while also being able to understand the legal implications and how to implement the designs properly (Jaeger, 2006).

Missing Framework and Ambiguity of which Guidelines to Follow

Numerous entities have found it difficult to implement an effective Section 508 testing plan due to the lack of access to an established testing plan framework (Jaeger & Matteson, 2009; Michalska, You, Nicolini, Ippolito, & Fink, 2014). This resulted in each entity deciphering the standards with their own interpretations and implementing the guidelines individually by making their own testing plan framework. The ambiguity of which accessibility guidelines to use or how to use them, contributed to the difficulty in developing testing plans. According to Olalere and Lazar (2011), testing groups often reported not knowing which accessibility standards or guidelines to follow so they would end up doing several different guideline tests for the same sites.

Lack of Enforcement, Training, and Education

Lack of governmental enforcement of accessibility compliance, difficulty finding technical staff with accessibility knowledge, and lack of accessibility training in education create a challenge to the accessibility testing process. The lack of enforcement was demonstrated by a 7-year gap, between 2001 and 2008, where the Department of Justice did not provide any compliance checks, or at least did not present the compliance reports to Congress regarding accessibility efforts (Olalere & Lazar, 2011). This lack of enforcement has tended to make accessibility a lower priority for many projects (Michalska et al., 2014). This challenge is diminishing as the U.S. government is working towards more legislation to encourage accessibility solutions. For example, in 2018, the U.S. government passed a refresh for the Section 508 law to incorporate the WCAG 2.0 guidelines (36 CFR Parts 1193 and 1194, 2017).

There were also the challenges of training staff for Section 508 while also keeping up with the rapid changes in technology and the rapid turnover rate for web-page designers (Michalska et al., 2014). The rapid turnover rate of developers has created a lack of accessibility-oriented design mindsets in the workforce. In addition, it has been difficult to educate application developers on how to make technology accessible when the technology is rapidly changing. As expected, functionality of a website usually takes priority over technological accessibility.

Since training can be a steep learning curve and accessibility training typically hasn't emphasized in a classroom setting, one alternative to the very time-consuming manual testing process has been the development of automated testing methods. While advances in automated testing have been made, some researchers found that it is not a one-stop replacement for all accessibility related evaluation; humans completing some aspects of testing is still needed (Jaeger, 2006). Automated testing has not yet addressed compatibility issues with assistive technologies. Other accessibility testing methods encouraged by Jaeger include using expert testers. While having expert testers on staff can be beneficial, testing is often performed by existing staff, regardless of their Section 508 background. Therefore, it often defaults to existing staff to learn Section 508 testing processes and find the resources to perform the testing (Rowland and Whiting, 2016). In the current study, use of student and full-time entry-level researchers was a key component in development of a functional 508 testing team.

Methods

The following methods describe the development and implementation of a testing plan framework to assess the DATIM application's compliance with Section 508 standards. The process of developing a testing plan framework began in November 2016 by assigning 3 inexperienced students to research existing techniques to demonstrate a web application's Section 508 compliance. The result of the student's research was that the compliance results were very dependent on the individual investigator interpretations of the Section 508 criteria. In order to provide consistent and reproducible testing processes for DATIM, an in-house manual testing method process modeled on the written criteria in Section 508, was developed. The first step in this process was the creation of an accessibility focused team tasked with the development of manual testing process outlining the accessibility requirements included in the Section 508 legislation.

Creating an Effective Accessibility Team

Identify a Project Leader

To create an effective Section 508 team, a project leader was identified to coordinate criteria research, suggest criteria testing solutions, and lead Section 508 training sessions for students and full-time research staff. The leader position required effective management of testers and developers, up front evaluation of testing needs, collaboration with other staff responsible for documentation and reporting, and continuous research to ensure proper methods were being employed. This approach was followed with the idea that it is better to share the knowledge with everyone on the team so that there is an overall increase of active awareness of Section 508 requirements, which will ultimately lead to a more accessible product.

Conducting Research

Due to the rapidly growing nature of the Section 508 field, it was imperative to conduct continuous methods research in order to stay updated on the newest and most effective techniques and standards for testing. It was discovered that a variety of tools that were available to help streamline the general process of manual accessibility testing. Conducting extensive research also helped the team identify the appropriate interpretation of standard Section 508 criteria and definitions for DATIM's needs. Focusing on the accessibility com777pliance of the application (DATIM) helped to mediate some of the ambiguity of standards which are written in a very broad-based manner.

Trainings

The research and communication organized by the project leader eventually evolved into in-house Section 508 trainings for basic Section 508 education and manual testing. In this team, the project leader would conduct Section 508 trainings for new hires and updated trainings for previous testers.

The DATIM manual testing method was developed through research into the various criteria that needed to be addressed for Section 508 compliance and filtered through the more specific needs of the DATIM application. Once the research was completed, the project leader created a step-by-step document specifying the applicable criteria that needed to be tested on DATIM. Time after this document was created, SUU learned of another instructional document that the Forest Service (FS) Chief Information Officer's (CIO) office had made for Section 508 compliance (S. Martinez, personal communication, 2019). However, there was still a need to pursue the creation of an instructional document at SUU because the file the CIO's office published could not be accessed by SUU (since no one had access to the CIO's internal share site). With access to limited resources, each training was adjusted based on the tester's feedback for improvement. When the team transitioned to an automated form of testing, the trainings were then redirected to the automated method and was taught to all current employees and new hires.

Finding Tools and Resources

Research assisted the SUU team in finding existing tools and resources for to include in the testing process. A manual testing guide was first created which was filled with links to websites and reading material that helped testers understand what and why they were testing. Some web-based tools used included the Paciello's Web Accessibility Tool ("Web Accessibility Toolbar (WAT)", 2012) for use in Internet Explorer and the Web Developer Tool and Web AIM eyedropper as Google Chrome extensions.

Manual Testing Methods

The manual testing method was matured over time; allowing each standard to be tested step-by-step and interpreted the same way (or as consistently as possible) by various testers. The process for establishing one set of interpretations was the following:

1. The development of initial interpretations of the standards were set by the project leader.
2. Then the project leader would review of the initial interpretations with the testers.
3. The interpretations were then tested against DATIM to see whether the testers would come to the same conclusion about the interpretations and feasibility of the method.
4. The interpretations were either committed as testing instructions for use or discarded if the interpretations and instructions failed. This would cause the project leader to search for new and more viable methods.
5. Once the testing methods and interpretations were established, the project leader conducted trainings to refresh each tester on the interpretations.

Manual testing of DATIM – which had about 50 pages – continued for approximately 12 months (Table 1) during which time it was noted that manual testing could be a very time-consuming process. On average, testing took about 40 hours for 1 page in DATIM as a new tester and down to 2-6 hours for 1 page in DATIM as an experienced tester. After analyzing the amount of time required for manual testing, the team opted to find an automated testing method to see if the testing was comparable in terms of quality yet would reduce the amount of time spent testing. Discussions with the CIO resulted in SUU gaining access to the Accessibility Management Platform (AMP) software (AMP, 2019).

Automated Testing Methods

AMP is a paid-subscription, automated testing tool that tests web-based applications and websites for accessibility compliance. It was used because the Forest Service already had a subscription to this tool. A software engineer at the FS CIO team discovered and recommended the AMP software because it suited developers' needs (D. Hamilton, personal communication, 2019). Although this tool is not used throughout the whole federal government and should not be relied upon as the sole method of testing, the FS-CIO encourages the use of AMP (S. Martinez, personal communication, 2019). Also, while AMP requires a paid subscription and is the tool SUU used for testing, there are many free automated testing tools, such as Axe or WAVE.

SUU tested for compliance on the DATIM application for 4 live versions on the Production server and one version on a frozen, nonpublic server (Table 1). Once AMP was set up, the team performed the next cycle of full DATIM testing (version 8) using both AMP and manual testing to compare results and determine which method would be most efficient for DATIM. In order to assess the similarities and differences between automated and manual testing, full DATIM testing was performed with each method over a span of 6 months on the frozen server.

Table 1. Testing Schedule for Data Collection

Server Type	Version	Testing Dates	Testing Method	Number of pages tested
Live	7.0.1.17045	January - March 2017	Only Manual	3
Live	7.0.3.17090	April - June 2017	Only Manual	8
Live	7.0.6.17150	July - September 2017	Only Manual	17
Live	7.0.7.17214	October - December 2017	Only Manual	3
Frozen	8.0.1.17342	January 2018 - June 2018	Automated and Manual	50

The last version tested for data collection purposes sat on a frozen server from January 2018-June 2018. This was the only version that was tested fully, for both manual and AMP testing.

Findings

Advisory Techniques

Five advisory techniques were discovered which were helpful in creating an effective Section 508 testing process for the DATIM application. These techniques included: using the appropriate testing guidelines, training staff in a variety of Section 508 issues, Extensive documentation of all aspects of the testing process, communication through social media, and automation of the testing process whenever possible. Additional details are presented below for these advisory techniques.

Testing Guidelines

Dependent on sector classification, federal applications, such as DATIM, should follow the standards outlined by the Section 508 law (Chief Information Office, 2016). With the WCAG 2.0 implementation in the Section 508 refresh (36 CFR Parts 1193 and 1194, 2017), testing criteria became more consistent between federal and international accessibility guidelines. State, local, and private companies who do not receive any governmental funding for product development or as subsidiaries, should use the ADA guidelines (28 CFR Part 36, 2017-revised).

Staff Training Leading to Positive Learning Outcomes

The SUU team trained a total of 14 students on Section 508 from 2017 to 2019. Students were used to conduct extensive testing and research, which is similar to responsibilities interns have when working for private companies. Training students to be experts exposed them to this very dynamic field of work and provided real-world experience.

Importance of Documentation

Documentation was a crucial aspect of the Section 508 testing process. By creating and utilizing a reporting template and a step-by-step document that interpreted each of the Section 508 standards for the DATIM application, testing and communication were enhanced and more consistent. Documentation assisted the project leader with reporting compliance estimates for DATIM. It also clarified areas lacking in compliance to be reported back to the developers which enhanced the developers understanding of 508 requirements and allowed for more efficient time allocation.

Communication through Social Media

With 75% of the United States' population participating in social media platforms, more people are using social media to communicate about their projects (Fichtner, 2015). GitHub, a social media platform used by the SUU and UNLV team, allowed the team to work virtually. This tool provided repositories where the UNLV and SUU staff could report issues and track fixes ("Build software better, together", 2019).

There are many social media platforms that exist can serve the needs of organizations. Other forms of social media use in this type of environment may include Kanban flow, One Desk, and Orange Scrum, among others. These forms of communication were easily utilized by a variety of testers and developers.

Automated testing efficiency

Initially, the manual testing effort involved a large amount of time to complete a small number of application pages. For example, manual testing of just one page required from 6-40 hours depending on the experience of the tester, the page complexity, and extent of non-compliance of the page. Using AMP, testers were able to fully test approximately 50 pages in DATIM.

Although automated testing does not test everything, such as the compatibility of assistive technology with an application, it reduced testing time dramatically and increasing the reproducibility of violations found. It was also found that AMP could potentially test 80 % of the standards, however this was not the end of the process. The AMP tool still provided listings for issues identified as "Needs Review" where a human user, falling back on the manual testing methods, needed to confirm whether it was a real violation or not.

Discussion & Conclusion

The findings presented above demonstrate that, despite the challenges of not having a testing plan framework, having limited access to resources, or testing experts, it is ultimately possible to reach Section 508 compliance effectively through the development of a customized testing plan with methods similar to the ones presented. It was clear that the lack of an established professional testing plan framework for DATIM greatly increased the initial learning curve for staff and it required extensive staff training and research into testing techniques.

During the process of developing a testing plan for DATIM's Section 508 compliance, coherence of test results improved via the use of automated resources, documentation templates, and through resolution of the testing guideline ambiguity. AMP served as a testing platform that both testers and developers could rely on for descriptions of each WCAG 2.0 guideline and older Section 508 standards, examples of compliant vs noncompliant code, and a summary of techniques that could help them apply fixes.

There was an apparent improvement in the team's Section 508 efficiency throughout the project. The time and cost associated with testing decreased once research on testing methods were completed and the staff were trained in these methods. In addition, there was also an increase in efficiency when the automated methods were instated. Furthermore, the DATIM application was having Section 508 issues fixed during its development stage rather than having it done at the end of the development project, thus saving time and money by alleviating the need to come back to it at the end, when the application should primarily be in maintenance mode.

One benefit for students that was realized during this project stemmed from the increased awareness of accessibility issues that the training and testing imparted to the students which was similar to the findings of Smith (2018) and Rowland and Whiting (2016). The awareness the students gained has served as a foundation for some of their future work; for example, some of the students involved in the testing were pursuing careers where they will be directly helping people. Therefore, technological accessibility awareness will allow them to be more conscious of the products and technologies they use for their future clients and patients.

References

28 CFR Part 36. (2017 –revised). Nondiscrimination on the Basis of Disability by Public Accommodations and in Commercial Facilities. Retrieved from https://www.ada.gov/regs2010/titleIII_2010/titleIII_2010_regulations.htm.

29 U.S.C. § 794d (1998). Section 508 of the Rehabilitation Act of 1973 (amended).

36 CFR Parts 1193 and 1194. (2017). Information and Communication Technology (ICT) Standards and Guidelines. Retrieved from https://www.federalregister.gov/d/2017-00395

Accessibility Management Platform (AMP) - Level Access. (2019). Retrieved from https://www.levelaccess.com/solutions/software/amp/

Bruyère, S. (2008). Making college web sites accessible to students with disabilities. *Community College Journal,* 36-39.

Burgstahler, S., & Doyle, A. (2005). Gender differences in computer-mediated communication among adolescents with disabilities: A case study. *Disability Studies Quarterly, 25*(2), retrieved from http://dsq-sds.org/article/view/552/729.

Build software better, together. (2019). Retrieved from https://github.com/

Chief Information Office, Forest Service: Southern Region. (2016). Laws and Regulations Applying to Federal Agency Programs and Activities [Internal - Accessibility]. Retrieved from https://usdagcc.sharepoint.com/sites/fs-r08-rwhv/SitePages/Accessibility%20(National).aspx

Fichtner, C. (2015). How to successfully use social media on your projects. Paper presented at PMI® Global Congress 2015—EMEA, London, England. Newtown Square, PA: Project Management Institute. Retrieved from https://www.pmi.org/learning/library/use-social-media-on-project-successfully-9920

Foley, A. & Regan, B. (2002). Web design for accessibility: Policies and practice. *AACE Journal, 10*(1) 62-80.

Hasan, N., Ashraf, M., Ahmed, E., Hasan, M. R., & Bhattacharjee, V. (2017). The impact of information and communication technology (ICT) on the lives of disabilities: A case in Bangladesh. *International Journal on Disability and Human Development*, *16*(3), 301–310. Retrieved from https://doi.org/10.1515/ijdhd-2016-0027.

Jaeger, P. T. (2006). Assessing Section 508 compliance on federal e-government web sites: A multi-method, user-centered evaluation of accessibility for persons with disabilities. *Government Information Quarterly, 23*(2), 169–190. Retrieved from https://doi.org/10.1016/j.giq.2006.03.002

Jaeger, P., & Matteson, M. L. (2009). E-government and technology acceptance: The case of the implementation of Section 508 guidelines for websites. *Electronic Journal of E-Government 7*(1) 87-98.

Michalska, A. M., You, C. X., Nicolini, A. M., Ippolito, V. J., & Fink, W. (2014). Accessible web page design for the visually impaired: A case study. *International Journal of Human-Computer Interaction*, *30*(12), 995–1002. Retrieved from https://doi.org/10.1080/10447318.2014.925771.

Olalere, A., & Lazar, J. (2011). Accessibility of U.S. federal government home pages: Section 508 compliance and site accessibility statements. *Government Information Quarterly 28*(3), 303-309.

Onyeabor, S. (2016). Addressing health disparities at the intersection of disability, race, and ethnicity: The need for culturally and linguistically appropriate training for healthcare professionals. *Journal of Racial and Ethnic Health Disparities*, *3*(3), 389–393.

Russell, A. (2019). Website accessibility: Make sure your website and job application portal is accessible | California Employment Law. Retrieved from https://californiaemploymentlaw.foxrothschild.com/2019/04/articles/discrimination/disability-discrimination-discrimination/website-accessibility-make-sure-your-website-and-job-application-portal-is-accessible/.

Rowland, C., & Whiting, J. (2016). How can nontechnical individuals assist in web accessibility testing? *2016 ICT Accessibility Testing Symposium*. Retrieved from https://2019ict.org/wp-content/uploads/2019/05/Proceedings-of-the-2016-ICT-Accessibility-Testing-Symposium.pdf.

Smith, A. (2018). Potential job applicants sue companies with difficult online forms. Retrieved from https://www.shrm.org/ResourcesAndTools/legal-and-compliance/employment-law/Pages/Job-Applicants-Sue-Companies-with-Difficult-Online-Forms.aspx.

Vu, M. N., Launey, K. M., & Ryan, S. (2019). Number of federal website accessibility lawsuits nearly triple, exceeding 2250 in 2018 | ADA Title III. Retrieved from https://www.adatitleiii.com/2019/01/number-of-federal-website-accessibility-lawsuits-nearly-triple-exceeding-2250-in-2018/.

Web Accessibility Toolbar (WAT). (2012). Retrieved from https://developer.paciellogroup.com/resources/wat/

Copyright Notice

Comparing Free Automated Accessibility Testing Tools

Kathryn Weber-Hottleman

University of Connecticut
25 Gampel Service Drive, Unit 1138 Storrs, CT 06269-1138
k.weberhottleman@uconn.edu

Abstract

Automated accessibility testing tools have the ability to cut down on testing time and offer quick insights into both visible and hidden page elements. However, it can be difficult to identify the testing tool that meets the user's needs, since each automated testing tool has distinct strengths. Free automated testing tools give users the opportunity to determine what style of automated testing tool will be most beneficial. This paper will provide a comparison of four popular free automated testing tools: WAVE by WebAIM, the ARC toolkit by the Paciello Group, axe by Deque, and the Accessible Name and Description Inspector (ANDI) by the Social Security Administration. After evaluating these testing tools, users will have a better sense of which will meet their needs and should feel confident exploring these tools.

Introduction

Automated accessibility testing tools are a great starting point for all accessibility testers. They have the ability to dramatically cut down on testing time and offer quick insights into both visible and hidden page elements. However, it can be difficult to identify the testing tool that best meets the user's needs. Different tools have different insights in their analyses, and they often have different price tags attached to them. Furthermore, some tools are geared towards new users, while others are best interpreted by experience accessibility testers. In order for the user to get a sense of which tools will be most beneficial, it is useful to practice using free accessibility testing tools before investing in an enterprise-level tool. This paper will provide a comparison of four popular free automated testing tools: WAVE by WebAIM, the ARC toolkit by the Paciello Group, axe by Deque, and the Accessible Name and Description Inspector (ANDI) by the Social Security Administration. After evaluating these testing tools, users will have a better sense of which tool will meet their needs and should feel confident exploring these tools.

WAVE by WebAIM

The first tool is WAVE by WebAIM. WAVE is compatible with Chrome and Firefox and has an additional free Chrome plugin. It checks one page of a site at a time, focusing on the currently opened page. WAVE is keyboard accessible. It is an excellent tool for getting a sense of the page's structure and content. It describes generally where errors could be present on the page, based on the elements it finds. Along with this, it offers thorough explanations for why an error could be present and how to fix the potential error.

WAVE is set up as a sidebar displayed alongside the page under review. The sidebar shows different categories, including a summary of issues, details, documentation, and an outline. It also provides options to toggle CSS styles and to see only color contrast issues. By choosing "Styles", which is the default view, users see the page rendered with CSS. "No Styles" strips out the CSS and allows the user to see if any valuable information is provided only through the CSS. The "Contrast" tool identifies areas of low color contrast and provides users with tools to assess the foreground and background colors' contrast. This can be done by entering a Hex code or by choosing a color from the color picker. Contrast then identifies the contrast ratio and evaluates if the ratio passes WCAG (Web Content Accessibility Guidelines) 2.0 levels AA and AAA for both normal and large text.

The summary shows what WAVE has detected, showing not only errors and alerts but also the page's elements, like features and structural elements. The initial summary view also displays a markup of the different elements and errors on the page under review. In the detail view, users have the option of filtering issues based on review criteria, such as WCAG 2.0 level A, WCAG 2.0 level AA, and Section 508. Regardless of how users filter issues, errors are presented at the top of the sidebar, followed by alerts and then features, which are page elements. Users can use checkboxes beside element icons or element groups to show or hide markup on page elements. When in the details view, clicking on an element icon shows the element's location on the page. Keep in mind that some issues are strictly code-based and may not appear visually on the page. In the documentation view, users can see more information about issues. There is an information icon next to each page element or issue in the details view, and selecting this icon routes users to documentation about that element or issue. Documentation provides information on why it is important for the element to be accessible, along with general information on how to remediate the issue. In the outline view, users can see all of the heading levels on a page. In addition to showing the user the page's structure, this is also a quick way to see if heading levels are appropriately nested and mimics what some screen readers are able to do.

Out of the WAVE sidebar, at the bottom of the webpage, there is a popup that displays the element's code in context. Users can therefore see exactly where the element or issue is in the code. This tool is especially useful when taken in conjunction with the details view, because when the user selects an element through the details view and then expands the code popup, the code will show the element's specific location.

There are a few cons to using WAVE. It is not possible to custom sort or export test results, so users are restricted to viewing the results only in the web browser, alongside the webpage. Also, WAVE does not evaluate if the page has a language attribute. Finally, if users select an element category on which to focus, clicking on the element category does not identify all of the element's instances at once. Users must click through each element individually to see where it is on the page.

WAVE is strong at identifying poor color contrast, images without alternative text, document language, empty and redundant links, and empty table headers. WAVE works best if the user knows how to read the site's code, because WAVE shows where in the code the error occurs. It is still up to the user, though, to be able to tell if a property or attribute is missing from the code, like an ALT tag or an iframe title. The biggest takeaway about WAVE is that the user needs to know what he or she is reading in WAVE's report, so the user can determine if an error is present or not. The steepest part of the WAVE learning curve is knowing how to interpret a site's code and find code-based issues.

ARC by The Paciello Group

The ARC toolkit is very similar to WAVE in terms of scanning and reporting. It is compatible with Chrome via a free plugin and is keyboard accessible. It examines the current page. After installing the plugin, it can be found in developer tools by right clicking, selecting "Inspect", and scrolling through the tools at the top of the inspect pane. ARC displays two sections, Test Groups and errors. Test groups are arranged in several categories, including media, structure, keyboard, ARIA, color, and IDs. These categories show which WCAG 2.0 criteria are not met by the webpage by marking visible errors, visible warnings, and hidden errors. ARC also shows the total number of visible and hidden page elements. ARC has options to show and track focus, check page reflow, and check text spacing. These last two options are not defaults, as they are found in WCAG 2.1. A cautionary note about showing and tracking focus is that it will interfere with manually checking visible focus on elements.

By selecting "Run Tests", users can see errors on the page broken down into errors or warnings and visible or hidden elements. While part of the pane is a table showing errors based on category, the other section of the pane provides a list of what are termed "visible assertions". Visible assertions tell the user what errors and/or warnings are found on the page and shows how the elements are grouped. Groups include headings, landmarks, pseudoContent brought in by CSS, and others. Clicking on a group brings up the code for where each error or warning can be found and visibly indicates on the webpage where each instance of the element is located. It also indicates which code blocks demonstrate errors or warnings. Groups can get granular, because each group breaks down into several element types, which the user can choose to display or hide. When hovering over a visible element's error or warning code, ARC highlights the element on the webpage. For example, choosing the group Headings highlights all headings on the webpage and displays the heading level. This is excellent for manual testing, and many free testing tools lack this functionality.

There are a few cons to using ARC. It is not possible to view all errors and warnings at the same time after the initial Run Tests is complete—errors and warnings must be viewed in the group context. The only time users can see all errors and warnings is when Run Tests is first selected. Also, ARC does not show properties like language in the page.

ARC is strong at identifying both visible and hidden errors, including errors that occur because of CSS. One of ARC's best features is that it only reports errors and warnings in its Visible Assertions pane. This saves the user from having to weed through to determine identified errors from identified elements. The biggest takeaway about ARC is that the user should be able to read code, because ARC's reporting structure is based on displaying the page's code.
The steepest part of the ARC learning curve is knowing how to interpret a site's code and find code-based issues.

Axe by Deque

Axe by Deque is compatible with Firefox, Chrome, and Android, and it reviews one page at a time. It is more robust than WAVE and offers options to check all elements that can be tested through automated testing. Axe is keyboard accessible. It has free browser extensions for Chrome and Firefox, and, following the extension installation, it can be found in developer tools by right clicking, selecting “Inspect”, and scrolling through the tools at the top of the inspect pane. When first viewing axe, there is a button labeled “Analyze”, which, when selected, displays a list of the page’s issues. Issues are sorted into violations, needs review, rejected, and best practices. Axe bases its issues on WCAG.

In the output following page analysis, axe provides a list of issues, with options to inspect the elements in question’s nodes to highlight the visible issues on the webpage. Inspecting the element’s node brings users to the precise code containing the issue. Showing highlight puts a visible border around the elements that are in question.

After selecting an issue to view, axe shows several different pieces of information regarding that issue. First, axe provides an issue description. This expands on the error name to briefly describe how the issue can be remediated. Along with this, axe offers an analysis of the impact the issue can have on the webpage and provides a link to learn more about the issue. This link routes to Deque University, which displays a detailed page regarding the issue, its impact on users with disabilities, how to resolve the issue, and additional resources. Axe also shows the element’s location, which is a code snip. The element source, which follows the element’s location, shows the code snip in context. Finally, axe offers suggestions for resolving the issue.

Axe is strong at identifying visible issues and some hidden issues, such as errors regarding landmarks. It also has the capacity to scan for video captions. One of Axe’s best features is that it is dedicated to preventing false positives in its reports. This is a pro because it stops users from investigating errors that might not exist. Similarly to ARC, the biggest takeaway about axe is that the user should be able to read code, because axe’s reporting structure is based on displaying the page’s code. The steepest part of the axe learning curve is knowing how to interpret a site’s code.

ANDI by the Social Security Administration

ANDI, the Accessible Name and Description Inspector, is compatible with all browsers, including Internet Explorer, Microsoft Edge, and Safari, and it reviews one page at a time. ANDI is keyboard accessible. It is more robust than WAVE, offering options to check all elements that can be tested through automated testing. ANDI is installed by making the tool’s link a favorite or bookmark. Navigating to a webpage and then clicking on ANDI from the favorites bar activates ANDI on the page. ANDI is a toolbar that docks at the top of the page.

ANDI displays different modules, including focusable elements, graphics/images, links/buttons, tables, structures, color contrast, hidden content, and iframes. Not all modules are available on every page; ANDI only displays certain modules, like tables, if that element is present on the page. This cuts down on the number of possible issues a user has to review.

ANDI divides the toolbar into two sections, information about a specific element and general information about the page. For example, the focusable elements module will show individual elements on which the user can focus. It also offers general information, indicating the number of focusable elements on the page and identifying accessibility alerts. These alerts are broken down into danger, warnings, and caution. Danger elements are errors on the page, while warnings indicate a possible error. Caution elements need to be further analyzed. Selecting an alert expands the alert to show the precise issue and highlights the element in question on the webpage. Finally, there are additional components that ANDI offers in the upper right portion of the toolbar. The first refreshes ANDI, causing ANDI to rescan the page. Second is advanced settings, which offers the options to toggle highlighting elements, making the page linear, and collapsing ANDI to a miniature version. ANDI also has hotkey combinations that can be found in this area.

The default module is focusable elements. ANDI visibly indicates all focusable elements on the page and highlights the currently focused element specifically. ANDI identifies the type of element, for example a link, determines accessibility components, and provides ANDI Output. Accessibility components include characteristics of the element, such as ARIA attributes and HTML tags. Accessibility components also show an element's inner text and CSS styles, such as ::before. ANDI Output shows what a screen reader user would hear when focusing on a given element. Within focusable elements, there are also options to view tab order, title attributes, and label tags. Graphics/images, ANDI's second module, shows images' alternative text and the image source. The Output typically reflects the image's alternative text. Within graphics/images, there are also options to hide inline images, to hide and find background images, and to show font icons. Links/buttons' accessibility components show characteristics like inner text and if the link has CSS attached to it. Within links/buttons, there are options to show only links, only buttons, and only ambiguous links. Tables identifies not only table cell data, but it also shows table headers, if headers are present. Tables offers the option to view the table's markup, so that it is clear which cells are header cells and which are body cells. Structures observes headings, lists, landmarks, and live regions. It has the option to view reading order, show page titles, and examine role and language attributes. Color contrast shows the contrast of foreground to background for many different page elements, including links, buttons, headings, and paragraphs. It identifies the type of element and shows the contrast ratio along with a pass or fail indicator. If the element's color contrast fails to meet the WCAG requirement of 4.5:1 for small text, ANDI will suggest an alternative color that meets the contrast ratio. ANDI also identifies where manual contrast testing will be necessary. This is the case for gradient colors. Hidden content examines elements that may be hidden through CSS. Users can choose to reveal all hidden content and to see different CSS hiding techniques like display:none, position:absolute, and overflow:hidden. Users can also see elements styled with ::before and ::after, showing visual changes. Users can also view title attributes. Iframes shows any iframes present on the page and determines their titles, if available. If titles are not available, ANDI provides a warning to the user.

ANDI is fairly intuitive to learn and is more like WAVE in terms of ease of understanding. It shows both errors and page elements, and it is easy to identify where the different elements are on the page. It is possible to toggle different features on and off, like tab order indicators, which keeps the output from being too overwhelming. It also links out to explanations of how to fix different errors, similar to WAVE. One of ANDI's most useful features is ANDI Output, an aspect of the tool that shows how a screen reader would likely interpret each element on a page.

Analysis

Each automated testing tool has distinct strengths. Users may choose to use a combination of tools for testing, based on the information they need from the tool. Experienced users may desire a more in-depth tool, where users new to the field may want to start with visual, summary-style tools. This section provides an analysis of the tools' strengths, based on the tools' depth, ease of use, help documentation, and overall performance.

Overall, the author recommends ANDI for free automated testing. ANDI is relatively intuitive to use, and its interpretation of errors, warnings, and page elements is straightforward. ANDI Output is helpful for both new and experienced users, saving users from having to use a screen reader in addition to a testing tool to understand how a screen reader would interpret different page elements. ANDI is a robust tool that evaluates every element it is possible to test through automated testing.

For an in-depth look at errors in a site's code, the author recommends both axe and ARC. Both identify code-based issues and the issues' locations in the page's code. They also show visible and hidden errors and warnings, highlighting visible elements when the user examines a given issue. Both also have the capability of showing the specific node where the element in question can be found, which is especially helpful for developers and designers.

Perhaps the easiest tool to use is WAVE. WAVE's summary and details views are intuitive to understand, thanks to color-coding and different icons representing errors, warnings, and elements. At a glance, WAVE makes it easy to sort out issues from elements. WAVE shows all page elements and has the option to show or hide code. This is helpful for those who are just learning how web pages are composed and more experienced users, including developers and designers. WAVE also includes strong built-in help documentation, so the user can evaluate errors and solutions in the same sidebar.

For robust help documentation, users should look to axe. Axe provides remediation recommendations within the tool itself, and it also has the option for users to explore Deque University for in-depth information about errors, warnings, and remediation. This is helpful for new and experienced users alike. The help documentation in Deque University provides a rationale of why an issue is important to remediate, who the issue impacts, examples of how to remediate the issue, and additional resources. Whether users are just starting to learn about accessibility or are fine-tuning their testing approach, axe provides strong support.

Conclusion

Automated testing tools have the ability to cut down on testing time and to offer a quick evaluation of page elements. Free automated testing tools give users the opportunity to determine what style of automated testing tool best suits their experience and need, before investing in an enterprise-level product. While these tools offer different depths and breadths of analysis, all can provide users with insights into a webpage's accessibility. This tool analysis demonstrates what each tool offers so users can feel comfortable engaging with each tool and evaluating its efficacy for meeting their needs.

References

Deque. (n.d.). axe for Web: Open Source Rules Library for Accessibility Testing. Retrieved from https://www.deque.com/axe/axe-for-web/

Deque. (n.d.). List of axe 3.3 Rules. Retrieved from https://dequeuniversity.com/rules/axe/3.3

Social Security Administration. (n.d.). ANDI Guide. Retrieved from https://www.ssa.gov/accessibility/andi/help/howtouse.html

Social Security Administration. (n.d.). ANDI Modules. Retrieved from https://www.ssa.gov/accessibility/andi/help/modules.html

The Paciello Group. (2019, June 5). ARC Toolkit. Retrieved from https://www.paciellogroup.com/toolkit/

WebAIM. (n.d.). WAVE Web Accessibility Evaluation Tool. Retrieved from https://wave.webaim.org/

WebAIM. (n.d.). WAVE Browser Extensions. Retrieved from https://wave.webaim.org/extension/

WebAIM. (n.d.). WAVE Help. Retrieved from https://wave.webaim.org/help

Copyright Notice

Testing the Accessibility of Augmented Reality, Virtual Reality, and other Extended Reality Experiences

Jonathan Avila

Chief Accessibility Officer, Level Access
1600 Spring Hill Rd. 4th Floor
jon.avila@levelaccess.com

Abstract

This paper provides an overview of extended reality (XR) environments, describes the state of accessibility for XR experiences, and provides practical guidance for testing. Recommendations are provided for applicability of existing regulations, scoping of features, and outlining a functional approach to testing using current standards while considering best practices for inclusive experiences.

Overview

There is a continuum of real-and-virtual combined environments that are computer generated – between augmented reality (AR) and virtual reality (VR). Extended Reality (XR) is the term used to refer to these types of real and virtual combined environments that provide human-computer interaction. These environments which are relatively new and are continually progressing generally involve wearable technology and extend experiences by immersing the user or integrating content into the real environment.

Virtual Reality– a computer-generated simulation or 3D image or environment which is seemingly real using a helmet screen, sensors, etc. The user is immersed into a digital environment and cut off from the real environment (Example: Star Wars Jedi Challenge).

Augmented Reality– imposes a computer-generated view on top of the real world providing a composite view. The user is not cutoff from their own reality, but computer-generated objects are placed in their real world. (Example apps: Pokémon Go, Minecraft Earth, and apps that allow you to place furniture into your physical room.)

Mixed reality– fuses the augmented and virtual space together.

While formal accessibility of these environments is limited, there are many aspects that provide an inclusive experience to certain groups of users. For example, eye tracking and voice control features can provide access to some people who may not be able to use a pointing device. In addition, XR itself allows people with disabilities to experience locations that would typically not be reachable in the built environment, and so forth.

This paper will provide a high-level approach for testing XR for inclusive design in terms of an outcomes accessible to people with disabilities. Testing must involve the hardware, any companion devices, the platform, as well as software apps. The wholistic experience for the user needs to be considered to ensure that users with disabilities have an inclusive journey from the out-of-box experience all the way to the finish. Ultimately for the experience to be accessible, all aspects must be usable by people with varying disabilities and preferences. Thus, it is important to work with hardware as well as platform vendors in addressing inclusive experiences from the game or communication app to the desktop.

Depending on what you are testing, you will likely be testing the interactions between the hardware and sensors with the platform and software even if you are only working with a software provider. While the app provider may not be able to affect changes at the platform level, they will be impacted by the level of accessibility, or lack thereof at the platform level, and may need to build accessibility features into an app as a short-term solution.

Existing hardware, software, web, and functional standards can be applied to XR with adaption, gaps filled by best practices and with information from current research in the area. In addition, testing with users that have disabilities is crucial for access and provides valuable insight into practical experience that may not be addressed by current standards and guidelines. Ultimately providing a wide range of options will allow users to find a method of interaction that works best for them as “one size -- fits one”.

Hardware

Whether you’re testing hardware or testing an experience on a particular platform, it is necessary to understand the common hardware types and capabilities.

Standalone VR Sets

Standalone units generally have a built-in screen and computer. The performance and capabilities are dependent on the built-in hardware capabilities and users are not tethered to a separate desktop computer to use them. These may come with cameras and other sensors. Examples including Oculus Quest and Oculus Go.

Connected Sets (VR and AR)

Connected headsets include those connected to PC, PlayStation, etc., and may use some of the processing and computation power from the computer. These may come with cameras and other sensors. An example is PlayStation VR and the Oculus Rift.

Do It Yourself VR Models

These kits are often a container with no screen– just a device you wear, and you provide your own phone. The capabilities of the phone may be used, such as camera or sensors in the phone. An example is Google cardboard which allows you to use your phone in a cardboard container with plastic lenses to produce a 3D experience.

AR Systems

These systems often have headsets with lenses and use a laser to project images on your eyes or they overlay an image on a lens in front of you. There is no traditional screen – you look through a lens into the world that you regularly see with the AR experience overlaid on top. These types of systems include HoloLens and Magic Leap One and typically involve connection to a portable computer and controller unit.

Phone Only Apps

Some AR experiences simply work by using your phone without any additional gear. They use the camera and screen with sensors from your phone to augment your reality.

Other Components and Interactions to Consider

All interaction methods must be considered in the evaluation and if testing hardware or mobile companion apps, these must be tested as well. It is likely there are already multiple methods of access – some of which will be accessible to certain groups of people with disabilities but not good for others (for example, voice control, eye tracking, head tracking, etc.). Having multiple methods of input and output is best as users can find what works best for them. Examples of interaction controls include, but are not limited to:

- Controllers and gamepads (evaluate for tactile buttons and markings with contrast)
- Cameras and sensors
  - Motion and gesture recognition
  - Eye and head tracking
- Computer units (evaluate for tactile buttons and markings with contrast)
- Mobile Companion Apps – for control, configuration, and/or install of apps and services (evaluate against native iOS and Android accessibility requirements including WCAG)
- Headset/Microphones (depending on purpose)
  - Voice control and recognition
- Keyboard interface, mouse/touch pad

Evaluate the following when testing hardware and platforms:

- Physical System Controls
  - E.g., Powering on/off, controlling volume, etc.
  - Section 508 and EN 301 549 provide guidance on evaluating hardware and closed functionality in general
- Access to sensors and physical hardware, such as lights, as well as putting the device on
- Setup/configuration for use – consider the out-of-box experience
- Navigation of menus/options, user preferences, the desktop, and other common UI and settings in the platform
  - Section 508 and EN 301 549 provide guidance on evaluating platforms in general, including user accessibility preferences.
- Access to platform services, like communication, browsers, etc.
- Access to documentation and support services

Uses and Applicable Standards

Understanding the features, purpose and market of the app or gear is critical in knowing which regulations, if any, apply to the experience. For example, when two-way communication between human beings are involved the Twenty First Century Communications and Video Accessibility Act (CVAA) standards apply in the United States (US). If the unit contains video playback or recording capabilities the CVAA video programming requirements likely apply. These regulations are what is driving the initial discussions with many XR vendors as not only are the ACS features, but the path to reach those features, such as setup, login, desktop, app launching, app store, browser, any network or access related settings, as well as documentation and support.

Common Uses and Areas of XR:

- Communication (CVAA or Section 255 of the Communications Act likely applies)
- Video Programming Playback or Recording (CVAA likely applies)
- Health and Therapy (Section 504 and ADA may apply)
- Education (Section 508 like requirements may apply at the state level)
- Business/Retail (ADA may apply)
- Gaming and Entertainment, Recreation
- Training
- Navigation and Wayfinding
- Assistive Technology
- Artificial Intelligence (AI) and Virtual Assistants

Current Standards to Consider in applicability:

- CVAA (Advanced Communication Services (ACS) and Video programming)
- Section 508 (if sold into the US Federal government). Section 508 provides standards for web, documentation, software, hardware (including closed system) and authoring environments.
- EN 301 549 (European Standard suitable for procurement of ICT). EN 301 549 provides standards for web, documentation, software, hardware (including closed system) and authoring environments.
- Section 255 of the Communications Act (if non-ACS communication services are included, such as cellular, SMS, and non-interconnected VOIP services are integrated).
- Web Content Accessibility Guidelines (WCAG 2) (other W3C guidelines may apply as well, such as the User Agent Accessibility Guidelines when a browser is present)
- Game Accessibility Guidelines
- Accessible Player Experience Guidelines (formerly Includification Guidelines) (AbleGamers)
- International Game Developers Association Guidelines

The CVAA uses functional performance objectives to measure accessibility of ACS. A functional model provides the most flexibility to meet the standards while still meeting user needs. It can be used to measure the functional use of technology by people with disabilities as it is outcome based. Examples include, "Provide at least one mode [of operation] that does not require user vision" and "Provide auditory information through at least one mode in visual form".

Gaming accessibility guidelines can be applied to many aspects of XR as well. Other standards, such as the WCAG standards, can be applied but in a XR-centric way, like when captions are required – but with XR we need to make sure that 360-degree captions are available. For example, in an XR environment there may be people talking in front of and beside you while a television is playing on your right. There are many ways captions for these situations can be provided, and research is still ongoing to determine the best approach. But, flexibility exists in how accessibility options are provided at this time due to the functional nature of applicable standards. Examples of 360 caption techniques include showing a radar circle with dots showing where people are speaking around you and as you navigate those captions are displayed. Options might exist to show captions from different sources at once or show you an arrow in the direction of sound where someone is speaking.

The current best approach for testing is to test functional requirements with best practices aligning under those high-level goals. In addition, insight and best practices can be gained from looking at apps that have implemented accessibility features or usability features that turn out to be beneficial to accessibility and to look at other platforms, like gaming consoles, to discover the successes and failures and how they can be applied to the XR space. For example, Microsoft's narrator on Xbox and text/ speech APIs to support game developers at the platform level with text-to-speech and speech-to-text conversion to support real time communications.

Common AR/VR/XR Challenges

Understanding the current challenges and state of accessibility will guide the recommendations provided during testing. Some generally known challenges include:

- Motion/dizziness sickness/eye fatigue
    - Caused by display, your eyes ability to focus at a given distance, and disconnect between virtual and real worlds.
- Lack of accessibility features and assistive technology built into platforms
    - Stems from the fact that frameworks have not yet widely added accessibility properties
    - Cannot walk or hold things closer to make them bigger
    - Focus on visual stimuli
- Interfaces that have multiple means of input but do not consistently or completely support different modes of input/operation
    - Often rely on eye, head or visual/hand movement

State of Accessibility

Accessibility in this medium is in the early stages although many devices have many basic inclusive principles in their design. Some aspects work well for one disability but not another, for example, head and eye control may be advantageous for some people who cannot use a pointing device but can control eye movements, but eye control is impractical for people who are blind or visually impaired.

The level of accessibility of an app depends on different platforms (Windows, Android, Lumin OS, etc.) as well as any frameworks used. Similarly, the cross-platform engines used to create apps like Unity, Unreal based apps and games will impact the level of accessibility in an experience. Regarding the Unity engine, early research has created plug-ins that could be made available to game developers. For example, Microsoft's Seeing VR with Cornell University.

One area that there is access is in the area of use of AR for assistive technology (AT). Common uses are Enhancements for Low Vision Users and assistance for people who are blind/visually impaired. These solutions tend to be app based and have proprietary solutions rather than use of existing platform accessibility features. The ideas from these assistive technofixes could be applied to experience, for example, the same AT features used to highlight stairs and edges of a path in the real world via AR could be used to highlight the edges of a path in a virtual world.

Over time, regulation will drive more accessibility and specifically when devices are covered by CVAA, Section 508, or other international regulations.

Accessibility is Possible

One of the first challenges is educating people of what is possible. Most people have a knee jerk reaction that XR cannot be made accessible to people who are blind, and therefore, accessibility is out of the equation. While XR is a visual experience, it is helpful to communicate the following:

- The technology can map indoor spaces and communicate distance to points of interest and obstacles and know your placement in that space.
- The XR system can know what objects are in the environment, who is in a room, and other details, such as which direction they are facing.
- Understanding and exploring a virtual environment can be done through sounds, audio cues, tactile indications, smell, taste, temperature, balance, acceleration, and description of what is in the visual environment.
- There is a convergence of AI (machine learning) and VR/AR, e.g., services like Aira and Seeing AI used by people who are blind or visually impaired to gain access to the visual world.
- Visual stimulus is not the only method of navigation - 3D audio can be used for navigation. Audio and haptics could allow someone with good audio perception to navigate a virtual environment.
- Most systems support multiple methods of input, including keyboards and physical controllers.

Functional and Best Practice Testing

In addition to functional testing of inclusive design, best practices can be used to provide more specific techniques to ensure that experiences are accessible to a wide group of users with disabilities. Sample best practices by user type may include:

Deaf/Hard of Hearing

- Are there visual indicators for important ambient sounds and other noises?
- Are there visual/text indicators for important audio cues?
- Is there audio transcription for real time communications – text-to-speech and speech-to-text?
- Is there signing capabilities for video or Avatar chat?
- Are closed captions provided?
  - Are captions differentiated in a 3D space with multiple speakers and audio sources?
  - Is there control of caption presentation for video content?
- Are there ways to show emotions and feeling, not just text?
  - Are there quick inclusive ways to communicate, such as emoji or avatar body language?
- Can you control background volume from other sound volume control?

Vestibular Disabilities

- Are there settings to prevent motion sickness?
    - E.g., is there an option to slow down movement and animation in space?
    - Is there an option for placement of information in a central area?
    - Are there methods to move around without requiring physically moving around?

Visual Impairment

- Are there options to change size and color text for better contrast, including color filters?
- Are there options that do not rely on color alone?
- Can the user zoom in without getting physically closer?
- Are there audio cues for visual information?
- Is there an audible way to understand placement in an environment, ways to navigate to points of interest, and a way to summarize what is around you?
- Is audio description provided for video?
- Is information identifying objects and their properties spoken?
- Can users navigate sequentially in a meaningful order?
- Is there an audible way to identify who is speaking (when that information is shown visually)?

Cognitive and Learning

- Is there a tutorial?
- Are there text-to-speech options to speak displayed content?
- Is there a sandbox/playground mode?
- Are there difficulty settings for activities?
- Is there an intuitive menu system?
- Is there support for customizations and personalization?
- Is contextual guidance provided?

Motor Impairment

- Are there methods of navigation that do not require precise movements, motion, twisting or grasping?
- Can all operations be performed with one hand?
- Can motion sensors be disabled to prevent accidental activation?
- Are there navigation and input methods that can be controlled with a single switch?
- Is voice control and diction provided?

User Testing

Testing needs to be performed with users that have disabilities to ensure that the experience from start- to-end is inclusive, and methods of communicating are equivalent for all users. While some solutions may seem appropriate at conception – only testing with users can determine whether these solutions are accessible and provide the same experience.

Conclusion

While XR is still an evolving space and formal accessibility standards and settings have not been built out, there are many things experienced developers can do to make their apps more accessible and include more people. Testing of experiences can result in a useful set of issues and positive results can not only be used by individual app creators, but also shared with the broader group in building out more formal guidelines for apps and for platform creators and those who make hardware. Communications and video related areas of XR are a logical place to start addressing inclusion because there are current regulations with functional criteria that apply now in the US. Access to these areas requires access from setup through to the user interface as well as settings and thus, progress in these areas will benefit access to other experiences, including entertainment and gaming, business, healthcare, and more.

Copyright Notice

UX Research Methods supporting Text-to-Speech Accessibility Guidelines Development

Sam Kanta, PhD

The Sustainable Change LLC
8817 Camelot Trace, Sturtevant, WI 53177, USA
sam.kanta@thesustainablechange.com

Abstract

Presently, the capability of text-to-speech with existing Assistive Technologies (ATs) to pronounce speech accurately according to context is inconsistent and limited. Usability testing, from the User Experience (UX) research domain, augments the accessibility testing process (McNally, 2017). UX research methods can contribute towards the development of accessibility guidelines, from which practitioners have a foundation to test upon. This paper discusses how user scenarios, a narrative approach used in UX design research, were derived from various qualitative data sources to articulate user needs as part of text-to-speech (TTS) specifications being developed within the W3C Accessible Platform Architectures (APA) Working Group - the Pronunciation Task Force (PTF). The implications of future accessibility testing with UX methods for TTS and ATs in general are also considered.

Background

Text-to-Speech (TTS) is technology designed to transform digitized text into human sounding speech (Duggan and Deegan, 2003). In turn, the application of TTS in the accessibility domain provides access to text-based content for those with disabilities related to visual perception and/or cognition. Despite the advances in TTS since it's early development (see Dutoit, 1997), there are limitations to functionality that impacts negatively on the user experience.

Limitations of present-day TTS

There are three significant limitations to TTS identified so far by the Task Force, although the list is not exclusive of other findings in future work. The first limitation is the inability of TTS to determine the correct pronunciation for a given context. The second limitation is the lack of ability to present any phoneme existing in the world. The so-called dead languages - a term to denote those languages no longer being spoken, for example, Ancient Hebrew, Norse, and ancient Greek (Holmstedt, 2006) are also included. Third, and perhaps the most fundamental limitation, the incapability of current digital technology to convey prosody: a term encompassing key attributes that enable people to perceive patterns in speech. Nooteboom, Brokx, and de Rooij (1978, pp. 99-100) best summarized prosody as:

a) Temporal segmentation
b) Intonation
c) Rhythm
d) Rate
e) Volume fluctuations
f) Subjective quality of a speaker's voice.

The PTF was formed in 2018 to specify an optimal approach for browsers and operating systems to deliver text via speech synthesis; addressing the limitations previously mentioned and ultimately enhancing the experience of end-users with disabilities who rely on TTS as part of their daily lives.

UX Methods in Accessibility

There is an extensive body of research and practice that details the processes and artifacts produced in designing digital products for an optimum user experience. With introduction of legislation for products to meet accessibility conformance requirements in many countries, there has been a noticeable increase in the reference to inclusive user experience design; evidenced in the most recent edition of one of the most commonly cited UX Design and Research Methodology texts "Interaction design: Beyond human-computer interaction" by Preece, Rogers, & Sharp, (2019).

User Scenarios

The term "user scenario" has multiple interpretations in practice. The working definition from the Usability Body of Knowledge was adopted within the PTF as follows:

> A story which has the key elements of a realistic situation when the user would interact with the system being designed or evaluated. The scenario includes consideration of the user's goals, tasks, and interaction. Scenarios can be created for user groups, workflows, or tasks to explore, understand, and test the different types of needs and goals (UXPA, 2010).

Typically, the creation of user scenarios occurs before usability testing of low or high-fidelity prototypes; they can form the basis for test scripts and provide input into documenting use cases. In the case of the PTF effort, work on gap analysis and use cases were already in-progress. Valuable insights from user scenarios may occur irrespective when the research and analysis took place within a product development lifecycle. There were potentially thousands of user scenarios for a technology such as TTS. The data collection and analysis phase established the boundaries for what scenarios eventuated. The process for developing scenarios was in three stages: data collection, analysis, and generation. It is important to stress that these stages overlap in practice, resembling a more iterative approach.

Method

Data Collection and Analysis

Data collection included conducting semi-structured interviews, with questions focused on recalling a personal experience when using TTS that present challenges to completing a desired goal or task This is an ongoing activity at the time of writing this paper, with five participants. A significant portion of data collected were observational summaries of learners with disabilities in formal examination settings within the United States. Although qualitative data is often known for smaller groups of participants, compared to quantitative experimental designs, the summaries represented an aggregate in excess of 100 hours of individual observations.

The quality and accuracy of text pronunciation produced by ATs varied widely according to a user's context. Therefore, the actual software product type and brand and was not captured as an item for analysis. The focus of the data collection was on experiences lived or observed, rather than the specific technology. The emphasis on a solutions-neutral approach was to minimize any potential bias towards existing TTS products during analysis.

The analysis of qualitative data involved the identification of unique and recurring experiences. The process required asking questions about the setting and unique challenge faced by an individual interacting with TTS for whatever purpose. After several reviews, there were major themes evident, resulting in a simple topology of TTS use as depicted in Figure 1.

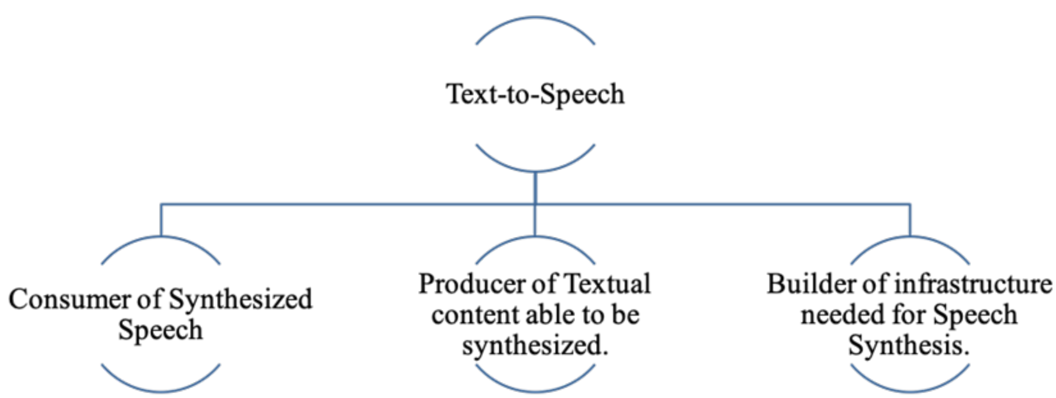

Figure 1. A Basic Typology of Text-to-Speech Use

Insights

What we arrived at were several perspectives on use and with it, implications for the TTS standard: There will be those who consume the synthesized speech to assist them interacting with their environment in order to achieve to achieve a goal or execute a task, as well as those who are required to produce text content that can be consumed by others via TTS.

A recurring context with data collected was from the education sector, specifically in the area of learner assessment, as noted in the draft user scenario document:

> In the educational assessment field, providing accurate and concise pronunciation for students with auditory accommodations, such as text-to-speech (TTS) or students with screen readers, is vital for ensuring content validity and alignment with the intended construct, which objectively measures a test takers knowledge and skills. For test administrators/educators, pronunciations must be consistent across instruction and assessment in order to avoid test bias or impact effects for students. (Ali, Kanta, Loew, Grenier, & Ran, 2019)

Key insights were shared by those in the user scenario sub-group to the team at-large. The outcome was two-fold: the contribution of other scenarios and the implications of those scenarios on the gap analysis and use case development.

Examples of User Scenarios for TTS

The next section of this paper presents some of the key user scenarios developed at the time of publishing the draft document, as they relate to the typology of TTS use in Figure 1.

End-User Consumption of TTS

The following user scenario example demonstrates the need for contextually accurate pronunciation of acronyms for a person using TTS to access online content and functionality to achieve a goal.

> Mary has a visual impairment and uses AT with TTS to help navigate through websites. She needs to travel to Ottawa, Canada and so goes to a travel website to book her flight. Mary already knows the airport code and enters "YOW". The site produces the result in a drop-down list as "Ottawa, CA" but her AT cannot pronounce the text. In frustration, she abandons her search and calls a friend to complete the booking online instead.

The version to follow is worded in a goal-oriented, gender neutral delivery, with the added clarification of a list that is regarded screen-readable:

> As a traveler who uses assistive technology (AT) with TTS to help navigate through websites, I need to hear arrival and destination codes pronounced accurately so I can select the desired travel itinerary.
>
> For example, a user with a visual impairment attempts to book a flight to Ottawa, Canada and so goes to a travel website. The user already knows the airport code and enters "YOW". The site produces the result in a drop-down list as "Ottawa, CA" but the AT does not pronounce the text accurately to help the user make the correct association between their data entry and the list item.

Digital Content Management for TTS

The management and editing of text content has given rise to tools such as Microsoft Word to produce text that is readable yet lacking in specificity needed for those using TTS to process and comprehend meaning. The following scenario summarizes the general use, with a text and graphics to demonstrate the challenge faced by those who are responsible in publishing content.

> As a math educator, I want to ensure that speech accuracy with mathematical expressions, including numbers, fractions, and operations have accurate pronunciation for those who rely on TTS. Some mathematical expressions require special pronunciations to ensure accurate interpretation while maintaining test validity and construct. Specific examples include:
>
> - Mathematical formulas written in simple text with special formatting should convey the correct meaning of the expression to identify changes from normal text to super- or to sub-script text. For example, without the proper formatting, the equation: a^3-b^3=(a-b)(a^2+ab+b^2) may incorrectly render through some technologies and applications as a3-b3=(a-b)(a2+ab+b2).
>
> (Ali et al., 2019)

The mathematical formula is also shown in Figure 2 as an image, where the alt text associated with the image file embedded in the digital format of this actual paper attempts to express through text.

$$a^3 - b^3 = (a - b)(a^2 + ab + b^2)$$

Figure 2 – Mathematical Equation Example.

Further clarifications from Ali, et al. (2019):

- Distinctions made in writing are often not made explicit in speech; For example, “fx” may be interpreted as fx, f(x), fx, F X, F X. The distinction depends on the context; requiring the author to provide consistent and accurate semantic markup.

- For math equations with Greek letters, it is important that the speech synthesizer be able to distinguish the phonetic differences between them, whether in the natural language or phonetic equivalents. For example, ε (epsilon) υ (upsilon) φ (phi) χ (chi) ξ(xi)

Software Engineering with TTS

The discussions within the PTF gave rise to considerations on the experience of those who would need to put the technical framework in-place for producers and consumers of TTS content. The following scenario resulted from discussions within the team about the implications of a TTS standard capable of addressing the current limitations:

> As a client-side user interface developer, I need a way to render text content, so it is spoken accurately with assistive technologies. (Ali et al., 2019)

The scenario is simple and to-the-point yet has significant ramifications to those in the field of designing and developing accessible digital products - be it software or hardware.

Implications for Accessibility Testing

People charged with the act of creating and generating text, be it in integrated development environments (IDEs), Graphical User Interfaces (GUIs), or even a basic text editor, will need a way to present digital text in such a way that the AT rendering the speech produces an experience for users that is accurate and consistent. The insights on producing TTS reinforced the rationale for developing specifications and best practice guidelines.

User Scenarios are one of many approaches applied by UX Researchers, Designers, in fact, anyone involved in the process of identifying the critical needs of users and the technologies to support them. The integration of user-centered design and research in the UX community-of-practice is complementary to software development of accessible products (Zacharias, Campese, Santos, Cunha, & Costa, 2019).

I can only reiterate McNally's (2017) assertion that "Accessibility testing professionals should be involved in design before coding starts (p.116)". Developing user scenarios to inform the design of digital experiences that are useful and accessible is an excellent place to start.

Next Steps

The user scenarios will undergo further refinement if others arise from ongoing analysis. As the TTS specifications continue formal reviews, there is the need to test its efficacy with users. One of the advantages of developing scenarios is the relative ease of transformation of text into usability tests.

Usability testing can begin when the proposed specification is at a stage ready for end-users to consume, and also produce TTS, as per the TTS Use typology mentioned previously. The findings from those studies will serve as an addendum to the normative specifications developed by the PTF and ultimately available for accessibility practitioners.

References

Ali, I., Kanta, S., Loew, C., Grenier, P., & Ran, R. (Editors). (2019) Pronunciation User Scenarios, W3C Editor's Draft 22 August 2019.

Cohn, M. (2004). *User stories applied: For agile software development.* Addison-Wesley Professional.

Duggan, B., & Deegan, M. (2003, September). *Considerations in the usage of text to speech (tts) in the creation of natural sounding voice enabled web systems*. In Proceedings of the 1st international symposium on Information and communication technologies (pp. 433-438). Trinity College Dublin.

Dutoit, T. (1997). *An introduction to text-to-speech synthesis (Vol. 3).* Springer Science & Business Media.

Holmstedt, R. (2006). Issues in the Linguistic analysis of a Dead language, with particular reference to Ancient Hebrew. *Journal of Hebrew Scriptures*, 6(11), 1-21.

McNally, P. (2017). *Why Usability Testing Should be Part of your Accessibility Testing Strategy*, presented at ICT Accessibility Testing Symposium: Automated & Manual Testing, WCAG 2.1, and Beyond, Washington, DC, USA.

Nooteboom, S. G., Brokx, J. P., & de Rooij, J. J. (1978). Contributions of prosody to speech perception. *Studies in the perception of language*, 75-107.

Preece, J., Rogers, Y., & Sharp, H. (2019). *Interaction design: Beyond human-computer interaction.* (5th ed.). Indianapolis, IN: Wiley.

Shintel, H., Anderson, N.L., & Fenn, K.M. (2014). Talk This Way: The Effect of Prosodically Conveyed Semantic Information on Memory for Novel Words. *Journal of Experimental Psychology: General*, (4), 1437-1442.

User Experience Body of Knowledge (BoK) (2010). *Scenarios.* User Experience Professionals Association. Retrieved from http://www.usabilitybok.org/glossary/19 - letters

Zacharias, I.C.S., Campese, C., Santos, T.B., Cunha, L.P., & Costa, J.M.H. (2019). 'User Stories Method and Assistive Technology Product Development: A New Approach to Requirements Elicitation', in Proceedings of the 22nd International Conference on Engineering Design (ICED19), Delft, The Netherlands, 5-8 August 2019. doi: 10.1017/dsi.2019.385

Copyright Notice

POURING RAIN: An Extended Model for Making Immersive AR/VR and Emerging Systems Accessible

Sunish Gupta

Northeastern University/Easy Alliance
20 Cameron St, Brookline MA 02445, USA
su.gupta@northeastern.edu

Anne Marie Rossi

Easy Alliance
20 Cameron St, Brookline MA 02445, USA
amrossi@swcp.com

Abstract

A decade ago, one would not have expected to use the web the way we do today. We can perform activities such as dependencies on digital assistants turning on appliances or control devices (IOT's) and turn-on the lights in one's home. Today, we are facing the movement of our 2-dimensional digital experiences of linear interactions into an enriched environment of continuous engagement and activity. This paper introduces a new model to design better interactions and experiences for people with disabilities for the AR/VR systems and emerging interfaces.

Introduction

In the future, we will be presented with many opportunities to experience enhanced interactions with technology. In the case of AR and VR, technology will affect reality in new and unexpected ways. Using AR, or Augmented Reality, a person can project onto the physical environment graphics, text, or a combination of digital enhancements. The popular Pokemon Go! (TM) is a successful example. Using Virtual Reality, or VR, a person uses an electronic device to create and interact with a seeming 3-dimensional reality. VR enables gamers to share invented worlds, golfers to play favorite courses in another country, and patients to manage chronic pain without drugs.

The testing of products, tools and devices using AR and VR, enabling people to engage with all their senses in new and fresh experiences, will also challenge our testing methodology. How can we be sure that these technologies will be delivered and used safely, not just by the majority of the population, but by people with disabilities such as limited vision, hearing, processing, speech and movement.

Thoughtful and well-designed accessibility testing will enable us to make the most of these opportunities and provide engaging, delightful, and successful experiences to all.

Motivation and Background.

The POUR (Perceivable, Operable, Understandable and Robust) accessibility testing protocol was devised when the web was in its first generation of evolution-Web 1.0. The Web 1.0 of the 1990's was mostly text-based and the information flow was uni-directional. Content was consumed only through text and simple form and graphic elements.

Since then, the web has evolved many times over. Web 2.0 introduced social networking, user-generated content and interactivity. Web 3.0 is the Semantic Web, providing a more meaningful interaction with proper formatting and communication protocols for larger and more complex and dynamic data sets.

The following first-generation Principles of Accessibility were proposed more than 20 years ago with heavy emphasis on web-focused technologies. They are described below:

Each guideline in the recommendation, with their associated success criteria, is organized around the following original four foundational principles9 that all web content must be Perceivable, Operable, Understandable, and Robust.

Under each of these principles are guidelines that describe specifically what needs to be achieved to conform including specific testable criteria.

Perceivable

Information and user interface components must be presentable to users in ways they can perceive. All users must be able to discern the content being presented to them. This means that content cannot be hidden from users regardless of the technology they are using to access it.

Operable

User interface components and navigation must be operable.

All users can navigate and interact with the web page. The interface cannot be built in such a way that requires a user to perform an action they are incapable of performing.

Understandable

Information and the operation of user interface must be understandable.

Users can understand the content being presented and how to interact with it. This involves two types of understanding. The content itself; the reading level used, content organization, etc., must be clear. And the interactions available on the page; users must be able to understand how to perform actions being required of them to use the application.

Robust

Content must be robust enough that it can be interpreted reliably by a wide variety of user agents, including assistive technologies.

Dependence upon one specific user agent, browser, or assistive technology should be minimized as much as possible. Attention to existing web standards and guidelines should be maximized. In general, a user should be able to select the technology that works best for them and expect a reasonably consistent experience.

This paper proposes to extend the existing POUR model. The principles based upon the new POURING RAIN model (acronym to inspire design and testing methodologies) will be relevant for AR/VR systems as well as emerging systems and technologies.

Today's emerging systems and interaction with web content are quite different. First, devices that access web content are selected, curated and presented in the desired format and mode. The versatility of today's interface systems to present the same information in different modes has evolved and matured that the users, disabled or not, simply expect more.

Secondly, there is more user-generated content and interactivity is deeper. Finally, points of access can use a PC, of course, but also a smartphone, assistant devices (Amazon's Alexa, Google Assistant, Apple's Siri, etc.) and emerging augmented reality and virtual reality (AR/VR) systems. The home environment now can include appliances, gadgets and health and asset-monitoring systems; all examples of this new breed of technologies that have, to a large extent, overlooked accessibility. Many of these systems and devices, controlled or connected through smartphones, both pose an advantage as well as introduce new challenges to their accessibility.

The POUR principle has served its purpose very well for more than two decades. Today's interfaces require us to take a new look at the POUR accessibility principles.

We hereby attempt to propose a revised set of testing and design principles for incorporating accessibility into such emerging and immersive systems.

These principles have been devised After careful research, and performing fundamental analysis, and discussion with the experts in the field, we propose to add seven principles to the existing POUR model.

Each of the additional seven principles enhance the existing POUR model to account for additional complexity. For each of the principle, the designer and developer must consider for multi-modality and how a user with certain limitations would comprehend, react and experience the interaction using the rest of their useful and sensory perceptions. Secondly, simplicity is achieved when user testing and evaluation are performed in each mode.

These seven principles are: Intuitive, Nudge, Gamify, Redundant, Alternative, In-situ and Next, hinted by the acronym POURING RAIN.

Components of Pouring Rain

P	Perceivable
O	Operable
U	Understandable
R	Robust
I	*Intuitive*
N	*Nudge*
G	*Gamify*
R	*Redundant*
A	*Alternative*
I	*In-situ*
N	*Next*

The first three principles (ING) make AR/VR systems simpler. The designer's challenge is to hide complexity.

Principle of Intuition:

With I, let us consider the aspects of the design that will create an intuitive experience, one that offers a comfortable and controllable interaction, prevents the user from getting lost, provides adequate reinforcement, feedback and timing, and results in expected success. The care in the development of the micro interaction is the key to the intuitive experience of a digital activity.

Principle of Nudge:

Provide a nudge to send the user in the right direction. The placement of the right information in the right time and space will reinforce or enable correction of the user's mind map. Having understood the goal, present a path of least resistance to that end. This will make possible better and faster decisions, leading to favorable outcomes, greater understanding and more confidence.

Principle of Gamification:

Gamification in its strict definition is the use of rewards, feedback and community support to motivate a game-player to complete goals. Let games inspire designers to present a seamless environment with consistent appearance and messaging, and to thoughtfully enable access so that everyone can play. The principles of successful game design remind us to create delightful and rewarding interactions, providing clear information about progress, warning us of danger, and encouraging the slaying of monsters that block the way to success.

Our RAIN reminders interact and reinforce each other as we endeavor to deliver digital experiences that engage, enable, motivate and entertain a broad audience in the future.

Principle of Redundancy:

Redundancy can be used to enable all users to access tools, information, and experiences. Design and test for the same information using multiple modes of presentation. Whether we're talking about images, interactivity, audio or video content, we need to consider each content element as either informative, functional or decorative. All types of content should fall into one of these categories and should be treated accordingly. Then, the alternate modal content should be created.

Principle of Alternate descriptions:

Alternate descriptions provide access and reinforcement to differently-abled users, and are useful to all users for providing information, feedback, and comfortable interaction. Audio cues, vibration, tactile feedback, and captioning are examples of possible modalities to employ. In the future, perhaps scent, temperature or taste will be options.

Principle of In-situ testing:

In-situ and interaction testing are key to assuring the delivery of an accessible web site, app, game or other digital product. Inviting testers of different abilities to contribute to the initial design, and in on-going testing, will provide insight for providing accessibility in the final product. This testing will reveal general as well as individual preferences for interaction.

Principle of Next anticipation action or behavior:

Control of the next anticipated action presents the opportunity for maintaining clarity of purpose and managing the user's behavior and expectations. Information provided in multiple modes inform a user about a new user environment, while clues and cues for navigation and orientation awareness help a user stay found.

The original expectation from the POUR model was that content must remain accessible, even as technologies and tools continue to change and evolve. Today the content sources can range from third-party cloud and API's, novel multi-sensors. Novel interactions using affective computing, 3D gestures or thought-control systems are still evolving, and the POUR does not stand up with the new rigorous requirements where reliability, efficiency and seamless experience are critical. This also means that the same content, features, and interactions should be available across technologies. For example, actions capable of being performed on desktop and mobile should also be available on immersive and future highly interactive systems that can take advantage of API's, SAAS and cloud services.

Today's user interactions involve more complex technologies and are beyond the capabilities of standard browser-based elements. The following complex interactive systems can be present in whole or presented as modules. True interactivity is achieved when it is aptly designed, coded and tested for accessibility. Such systems could include immersive systems, Kiosks, IoT or WebOT, API's, Third-party Cloud services, SAAS, and external systems that may or may not directly have been designed and test for WCAG. The test protocol and success criteria using the comprehensive POURING RAIN model are currently being researched and developed and will be available soon.

Conclusion

Accessibility to the opportunities and information provided by emerging digital technologies is a human right and, in some cases, a legal requirement. It is the responsibility of the designers and developers to ensure that accessibility is built in and evaluated at every stage of the design process. The decisions made by designers and developers are critical to providing a seamless and delightful experience for everyone regardless of technology and human sensory limitations. We present the use of the POURING RAIN model for more relevant and inclusive designing, evaluating and testing of AR/VR systems as well as other emerging technologies.

References

W3.org. (October 2016) *Understanding WCAG 2.0: A guide to understanding and implementing WCAG.2.0.* https://www.w3.org/TR/UNDERSTANDING-WCAG20/complete.html Accessed June 10, 2019

Ali, Irfan. (November, 2018) "WCAG:Go Beyond and Be Creative!" In *Proceedings of the 2018 ICT Symposium: Accessibility Testing Symposium:Mobile Testing, 508 Revision, and Beyond,* Arlington, VA, USA .

McNally, Peter. (October 2017). Why Usability Testing Should be Part of your Accessibility Testing Strategy. In *Proceedings of the 2017 ICT Accessibility Testing Symposium: Automated & Manual Testing, WCAG2.1, and Beyond* Washington, D.C. USA

Kushniruk, AW; Borycki, EM; Kannry, J. "Commercial versus in-situ usability testing of healthcare information systems: towards "public" usability testing in healthcare organizations." https://www.ncbi.nlm.nih.gov/pubmed/23388274 Accessed June 9, 2019

Copyright Notice

ICT Accessibility Testing: Recruiting and Training Techniques

Cammie Truesdell, M. Ed.

Project Manager / Senior Accessibility Analyst
New Editions Consulting, Inc.
103 W Broad St. Suite 500, Falls Church, VA 22046 USA
CTruesdell@NewEditions.net

Kristen Smith-O'Connor

Senior Accessibility Technician
New Editions Consulting, Inc.
103 W Broad St. Suite 500, Falls Church, VA 22046 USA
KSmith@NewEditions.net

Andrew Nielson, MPA (contributor)

Senior Program Manager / Senior Accessibility Specialist
New Editions Consulting, Inc.
103 W Broad St. Suite 500, Falls Church, VA 22046 USA
ANielson@NewEditions.net

Tyler Matney (contributor)

Principal
New Editions Consulting, Inc.
103 W Broad St. Suite 500, Falls Church, VA 22046 USA
TMatney@NewEditions.net

Abstract

Recruiting for successful accessibility testers requires identifying candidates who have good communication skills, show initiative, and are committed to continuous learning. Sound training approaches for new hires must incorporate informal and formal learning resources that provide a solid understanding of accessibility, Information Communication Technology (ICT) types, coding languages, testing tools, and hands-on learning opportunities. Co-testers or mentors can support learning approaches and activities. The success of testers largely depends on how effectively findings and recommendations to content development teams are explained. Additionally, the speed in which common issues are identified, patterns are located, and the quality of assistance to teams for devising solutions to remediate inaccessible ICT contribute to testers' success.

Recruitment

Job candidates who understand, communicate and explain concepts well, take initiative to find solutions, and are life-long learners (eager to learn new testing methods) make good testers. Excellent testers also know how to effectively and efficiently use provided tools, from testing and reporting to remediating.

Finding the right candidates to be accessibility specialists starts with recruitment. Posting a job listing is step one. The listing needs to indicate that at a minimum the ideal applicant has core competencies in HTML and Microsoft Office. A listing should also include how the candidate must have empathy, a passion for helping others, and the ability to clearly and concisely explain findings. The content developers and editors who remediate findings usually need details on the impacts of those issues to come up with appropriate solutions. Empathy and passion are important traits that allow the tester to perceive how an accessibility issue impacts persons with various disabilities and drives the tester to find viable solutions.

After gathering the details of what is needed for the position, it is time to locate potential candidates. There are many online career sites to advertise the posting as well as career fairs to speak with potential applicants. Aside from the necessary skillset, the target audience would be those who have a background in technology and show an interest in diversity issues. Users with disabilities is another group that has untapped talent. However, it is necessary to look for the requisite skills, not simply a user of assistive technology (AT).

Once there is a pool of candidates, it is necessary to conduct interviews. Trying to select candidates can be challenging - it is difficult to know whether people have true subject matter expertise or whether the resume is simply filled with buzz words. Asking certain types of questions can assist in narrowing down the field. To gauge how appropriate a candidate is for accessibility ask questions from multiple aspects of the field. For instance, a general question may involve determining whether multiple types of disabilities can be defined and/or types of AT. A technical question may focus on how to apply a code snippet. Other questions may focus on design specifics, accessibility standards, how to address specific scenarios, hardware, etc.

Training Techniques

After the hiring process is complete, the training must begin. A question that has plagued organizations for a long time is: what constitutes effective training?

While standards, techniques, and testing methodologies can differ across and within industries, the most commonly adopted set of standards for U.S. entities are the Section 508 standards and Web Content Accessibility Guidelines (WCAG). These rulesets provide criteria to measure accessibility. However, specific testing methodologies or testing tools are not outlined, which has resulted in varied testing approaches and increased training challenges. Both informal and formal learning can be leveraged to address these challenges and develop an effective training approach for an organization.

- A common learning methodology within the accessible world is informal learning. This "refers to activities initiated by people in work settings that result in the development of their professional knowledge and skills" (Lohman, 2009). Examples of informal learning may include exchanging resources with one another, gleaning information from the internet (blogs, coding sites, vendor publications, etc.), and testing with new tools or techniques. This learning method should be employed constantly to ensure a tester's skills continue to advance.

- In contrast, formal learning refers to "organized instructional activities that take place in educational or training institutions and often lead to some form of official recognition (e.g., a degree, certification, or credit)" (Lohman, 2009). Numerous formal trainings exist. It is necessary to carefully select those that meet organizational training needs and couple formal trainings with informal methods for continuous skill building.

The training approach must also take into account the testers' prior experience and existing skillsets, the applicable standards for the Information Communication Technology (ICT) being reviewed, and differences in testing approaches per industry. In short, there are multiple ways to approach training, but this task can easily be broken in to smaller segments so training can begin at the most logical starting point. Due to the overwhelming amount of information offered, only a few industry-leading training options are listed below to assist setting up a plan.

Start with the Foundation: What is Accessibility?

Successful testers must understand what accessibility means. Testers must also learn how inaccessible ICT impacts people with disabilities. This provides testers with a deeper understanding of accessibility barriers, helps develop empathy, and bolsters knowledge so that the impacts of inaccessible content on people with disabilities can be provided to stakeholders.

Informal learning options provide testers with a foundation for self-sufficiency in accessibility. Several common resources that testers can be directed to include: W3.org Web Accessibility Initiative (WAI) specifically, the How to Meet WCAG (Quick Reference) guide is helpful; Section 508 ICT Final Standards and Guidelines issued by the U. S. Access Board; and Section508.gov the General Services Administration's Government-wide IT Accessibility Program.

Formal learning options provide testers with a more structured environment to learn about accessibility. Several resources to consider for inclusion are: Department of Homeland Security's (DHS) Introduction to Section 508: What Is Section 508 and Why Is It Important?; Accessible Electronic Documents Community of Practice (AED COP); Deque's Accessibility Fundamentals and Accessibility Empathy Lab; Funka's Introduction to Accessibility; and webinars presented by vendors such as Deque, Level Access, Accessibility Online and 3Play Media. The International Association of Accessibility Professionals (IAAP) offers a Certified Professional in Accessibility Core Competencies (CPACC) certification. Several formal classes are offered through 3rd party vendors in order to prepare for this certification while informal content is offered through IAAP's website.

Building on the Foundation: Learn the Language

Next, testers must be familiar with the tools used to create the ICT, so they can speak the "ICT language". Testers do not necessarily need to know how to write code (although it is a plus) or be premiere content experts. However, the language of the ICT being tested should be comprehendible upon review, such as HTML, CSS, ARIA, Microsoft Office and Adobe PDF document formatting, or basic software code. Grasping the language assists testers in explaining issues to developers, as well as leading developers to a solution.

Many informal learning options for these languages are available. Several resources for testers include: W3Schools (coding classes from basic to advanced but is not accessibility-focused); Microsoft, Apple, Android, Adobe's Accessibility Resources; and Blogs and forums (e.g., Stack Overflow, blogs from leading accessibility experts).

While some formal learning options that are available do not have a specific accessibility focus, the information they provide is useful in learning concepts and languages. Online courses and tutorials are available through resources such as edx.org, Coursera.org, Lynda.com, Pluralsight, etc. that may be good starting points.

Putting it All Together: Testing

A person's sole or primary role does not have to be testing. A creative designer can identify contrast and color-alone conveying meaning issues. A business analyst can write requirements regarding focus management and help determine the reading order of content. A quality assurance analyst can look for appropriate table attributes and confirm form constraints and errors are properly identified. A project manager, content author or administrative assistant can ensure headings and lists are structured properly and meaningful image descriptions are provided. Regardless of whether someone is tasked with reviewing a full set of accessibility checkpoints or it is split among multiple roles, accessibility testing can be incorporated into any team member's duties.

No matter who is conducting testing, valuable results can be leveraged by the use of a combination of tools during the testing process. Keep in mind that the tools needed for testing take time to learn and vary by ICT type. One thing is always true, testers need to conduct manual tests and not solely rely on automated testing tools. Testers can also use AT to understand how someone with a specific disability encounters the content.

Manual testing is necessary because automated testing tools simply cannot check for all issues nor can they validate all input. Automated findings require human interpretation. Some AT has built-in mechanisms to compensate for bad code, so the software cannot be relied on fully. Therefore, the structure of content is the most vital piece to be evaluated. Ultimately, select the option(s) that make the most sense for the organization.

Tools for testing for accessibility range from using the keyboard-alone and browser favelets to full-fledged applications that provide diagnostic information on the structure of the content. There is no standard set of tools. Some tools will be mentioned within informal and formal learning resources - it is up to the organization and testers to determine which tools are most appropriate.

Applying a Training Approach

A training approach is only effective if 1) it applies to the ICT type(s) the testers will be testing and 2) covers all applicable accessibility standards for the industry being reviewed. Informal learning resources are available in the form of test scripts/processes, checklists, best practices, tips/techniques, and testing methods using AT. Examples include Section508.gov's Test for Accessibility, W3.org WAI Evaluating Web Accessibility Overview and Section508.gov's Test for Accessibility. Frequently test scripts and checklists offer a reporting vehicle as well. There are many formal learning resources too. These options typically require a fee. Examples include the DHS Trusted Tester Process for Web (based upon a test script (informal option)), Deque's virtual and in-person trainings, WebAim, and T-Base Communications Web Accessibility Training. These resources help new testers to identify the most common accessibility issues and guide them in how to repeatedly perform each test to get an accurate result, using the same tool and testing methodology. Note that formal training methods may or may not include the use of AT, but to reiterate testers may find it helpful to experience content like a user.

A training approach will be most beneficial and successful if it involves an interactive, hands-on environment. The testers will need to learn what accessibility issues to test for, how to use various tools to retrieve results, and how to interpret and record findings. Providing examples of accessibility issues alongside effective implementation of the same ICT concept will help a tester comprehend what to look for and strengthen testing and remediation skills.

Once initial training has been completed, consider co-testing or mentored testing to fortify and validate what testers have learned. Having someone to ask questions to, double check results, and review findings will help testers build confidence and improve skills. It is important that testers continue to cultivate skills with informal learning options (such as blog posts, updated understanding articles, instructional webinars, new web techniques, etc.) and/or formal learning options on specific ICT platforms, languages and testing tools to keep skills sharp. Being a life-longer learner is a key trait in a successful tester.

Success of a Tester

Effective accessibility testing can be conducted in a variety of environments and through different process flows. Depending on the organization's structure, testing may take place throughout the lifecycle or may be done prior to deployment or publishing. To successfully test, the expected level of conformance must be explicitly provided before testers begin. Subjectivity plays a role in stating something is a failure which is why testers need to comprehend what it means to be accessible. Moreover, testers need to be aware of usability issues. Content can technically be accessible but not useable. Though the expectations are that only accessibility issues will be identified, to a subjective degree usability has to be considered. Still, testers must be cognizant of not reporting usability issues simply due to opinion.

A successful tester must be able to confidently identify common issues, locate patterns, and prioritize findings based on severity. With little effort testers should be able to spot potential basic issues that different user groups face when encountering content. For instance, when viewing content testers could ask themselves questions such as: are the bolded phrases structured as headings; do the colors provide sufficient contrast; are the links and buttons in a logical tab order; do the images have meaningful descriptions; or are the table header cells properly identified?

When determining testing failures, it is important to remember the answer is not always simple. There are usually multiple ways to accomplish a task. A checklist is great as a general guide, but the single method it may describe is likely not the only option. Testers must be prepared to look for answers and solutions, instead of expecting senior staff to provide the answer or a black and white answer always being visible. Often it is necessary to create samples and try various tactics to find the most suitable answer for that particular ICT. Multiple solutions may be valid as long as the overall goal is achieved.

It is not likely that testers can record every accessibility failure. Rather, testers should recognize and document patterns to save time for themselves and the content developers or editors who will resolve the uncovered issues. Testers can simply indicate all pages with a particular look and feel that contain problematic elements. The content developer or editor may then be able to globally apply changes through templated elements. Testers must also determine the severity of issues. Some blocking issues should be considered "show stoppers," and should be addressed immediately. Alternatively, waiting until a new release or version for minor issues to be addressed may be sufficient.

Identifying risk in accessibility testing requires good judgement. Testers need to be able to keep an open mind and evaluate from different perspectives to determine the risk of inaccessible ICT to people with disabilities.

A testers' success will be measured, in part, by how well content development teams can interpret test results. These teams will no doubt have questions. However, the more the testers explain upfront, the less time the tester will need to spend with the teams reviewing issue reports. A good report identifies 1) what the problem is, 2) implications of the problem (including what groups of people with disabilities are affected), and 3) how to generally resolve the issue.

Part of a tester's job is to be proactive and eagerly seek new information regarding accessibility practices. With new versions of applications and new platforms of technology emerging testers need to adapt to evolving testing approaches. Content development teams often require creative assistance in formulating accessible solutions. It can take time to create samples and test various approaches to find an answer. Be sure to reiterate that the informal and formal learning resources made available during training should be used whenever needed.

References

Adobe (2015) Accessibility Resources | Adobe. Adobe.Com, www.adobe.com/accessibility/resources.html. Accessed 23 Aug. 2019.

Android Developers Google, (2016) Test Your App's Accessibility | Android Developers. developer.android.com/guide/topics/ui/accessibility/testing. Accessed 23 Aug. 2019.

Apple (2019) Accessibility - Apple Developer. Apple.Com, developer.apple.com/accessibility/. Accessed 23 Aug. 2019.

Coursera (2019) Coursera | Online Courses & Credentials by Top Educators. Join for Free. Coursera.Com. www.coursera.org/. Accessed 23 Aug. 2019.

Department of Homeland Security, 22 (Aug 2019) Section 508 Trusted Tester Conformance Test Process Version 5. www.dhs.gov/trusted-tester. Accessed 23 Aug. 2019.

Deque Systems (2018) Accessibility Empathy Lab accessibility.deque.com/accessibility-empathy-lab. Accessed 23 Aug. 2019.

Deque Systems (2018) Accessibility Training | Deque. Deque.Com, www.deque.com/training/. Accessed 23 Aug. 2019.

edX (2019) EdX Home." Www.EdX.Org, www.edx.org/. Accessed 23 Aug. 2019.

Funka (2019) Introduction to Accessibility. www.funka.com/en/we-offer/training/open-training/introduction-to-accessibility/. Accessed 23 Aug. 2019.

International Association of Accessibility Professionals (2019) IAAP Home. www.accessibilityassociation.org/. Accessed 23 Aug. 2019.

Lohman M. C., (2009) A Survey of Factors Influencing the Engagement of Information Technology Professionals in Informal Learning Activities." Information Technology, Learning, and Performance Journal, vol. 25, no. 1, pp. 43–53, citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.546.6902&rep=rep1&type=pdf.

Microsoft (2019) Create Accessible Office Documents. Office.Com, support.office.com/en-gb/article/create-accessible-office-documents-868ecfcd-4f00-4224-b881-a65537a7c155. Accessed 23 Aug. 2019.

Pluralsight (2019) Pluralsight - Unlimited Online Developer, IT, and Cyber Security Training. www.pluralsight.com/. Accessed 23 Aug. 2019.

T-Base Communications (2017) Accessibility Training for Customer Experience - T-Base Communications. Www.Tbase.Com, www.tbase.com/accessibility-training/. Accessed 23 Aug. 2019.

U.S. Access Board (2016) Create Accessible Documents | Section508.Gov. www.section508.gov/create/documents. Accessed 23 Aug. 2019.

Web Accessibility Initiative (WAI) 2018, Evaluating Web Accessibility Overview." www.w3.org/WAI/test-evaluate/. Accessed 23 Aug. 2019.

U.S. Access Board (2019) Test for Accessibility. www.section508.gov/test. Accessed 23 Aug. 2019.

W3Schools (2019), W3Schools Online Web Tutorials. Www.W3schools.Com, www.w3schools.com/.

Web Accessibility Initiative (WAI) WAI Home. www.w3.org/WAI/. Accessed 23 Aug. 2019.

WebAIM, (2019), WebAIM: Accessibility Training. webaim.org/services/training/. Accessed 23 Aug. 2019.

Yanchulis, D. (2017) Final Rule - United States Access Board. Www.Access-Board.Gov, U.S. Access Board, 2017, www.access-board.gov/guidelines-and-standards/communications-and-it/about-the-ict-refresh/final-rule. Accessed 23 Aug. 2019.

Copyright Notice

Remediation Order Matters: What to Fix First?

Sean Kelly

Digital Accessibility Engineer
Optum
11000 Optum Cir, Eden Prairie, MN 55344, USA
sean_kelly@optum.com
sean.kelly.a11y@gmail.com

Karen Herr

Digital Accessibility Engineer
Optum
11000 Optum Cir, Eden Prairie, MN 55344, USA
karen.herr@optum.com
karen.herr.a11y@gmail.com

Abstract

Whether one is remediating a digital property (e.g., website or mobile application) or providing guidance in the creation of a new one, the order in which issues are addressed is important and affects both the work of the professionals creating the property and its end users.

Conventions of Nomenclature

In writing this abstract, all digital properties are in scope. The most common examples include websites and mobile applications. Most of these ideas apply beyond those two categories but for the sake of brevity most properties will be referred to as “sites.” This should not be interpreted as excluding other types of digital properties, whether they are a mobile application running on a tablet, a smart TV or an internet-enabled kitchen appliance.

Remediations Large and Small

Clearly it is preferable to create products that are accessible from the very beginning. Before a single wireframe or composite is created, and long before the first line of code is written, decisions are being made that affect accessibility.

Even in the most mature accessibility practice, some artifacts will be produced, code with be changed, that requires going back and remediating. As soon as one scrum team is ahead of accessibility oversight, to some degree, your carefully planned project has acquired some aspects of a remediation.

Even a small-scale remediation benefits from avoiding rework via effective prioritization. This is the case when a smaller set of issues arises after a new feature is added to an otherwise compliant site or application.

Prioritization of accessibility defects is crucial for large-scale remediation efforts. For example, if your site has recently undergone an accessibility audit, and you have a full report of the defects, how do you integrate remediation back into your site? How do you make your site accessible quickly and for the greatest number of users?

Prioritization of Success Criteria and Checkpoints

Whether one looks at prioritizing WCAG Success Criteria (SC), checkpoints on a more granular checklist or specific WCAG techniques and failures (labeled by WCAG Number such as G19, H32, etc.), effective prioritization has a number of important benefits. Note that in some cases we are prioritizing issues differently than is stated or implied in WCAG 2.0/2.1. WCAG 2.0/2.1 broadly describes three levels of conformance: A, AA, and AAA.

There are undocumented and under documented dependencies and relationships between Success Criteria as well as issues that are foundational to broader accessibility needs that make a compelling case for separating WCAG 2.0 CR-5 issues ("Conformance Requirement 5 Non-Interference") into a category that preempts and precedes even the level A Success Criteria. (Kelly and Tyler, 2017)

Simplicity

There are multiple places where one Success Criterion overlaps another. In many cases, the stricter criterion or technique is actually simpler to implement and test than the more permissive one. One prominent example points again to SC 2.3.1, level A, "Three Flashes or Below Threshold" that is more difficult to describe, measure, and test than its level AAA equivalent SC 2.3.2 "Three Flashes". The latter is stricter, but much easier for purposes of definition, training and measurement.

Foundational Issues

Issues such as programmatic keyboard focus and operation of keyboard functionality affect very wide range and large quantity of users. Multiple Success Criteria for both WCAG 2.0 and 2.1 are covered by Guideline 2.1 "Keyboard Accessible". An approach which also prioritizes keyboard accessibility helps a large swath of the population, with or without identified disabilities, but also improves testability both "manually" and with automated tools.

Analytics: Key Workflows and High Traffic Pages

One axis of prioritization is to examine site analytics to determine the traffic that a page or section receives and remediate the high traffic areas first. Another is to focus on key workflows, such as searching for a product, adding it to a shopping cart, entering payment data, and completing a checkout.

Focusing on higher-traffic pages and key workflows early in a remediation, in combination with other prioritization schemes, can maximize the effectiveness of your remediation work and reduce the risk of most users encountering inaccessible content.

Fix and Reveal (FaR)

A further case for prioritization of concerns comes into play with the concept Seán coined as "Fix and Reveal." While the idea that any given bug might conceal another problem or set of problems is not new to anyone who has participated in the software development process, there appears to be a broad lack of appreciation of how this applies to electronic accessibility.

Simply stated, Fix and Reveal describes a design pattern where some issues are likely to obscure other issues. FaR identifies critical relationships between WCAG 2.0/2.1 Success Criteria. (Kelly and Tyler)

For example, fixing SC 2.1.1 "Keyboard" and SC 2.1.2 "No Keyboard Trap" reveals issues with SC 2.4.7 "Focus Visible", which in turn reveals issues with SC 2.4.3 "Focus Order," through multiple other SC that lead all the way down to 4.1.2 "Name, Role, Value."

We will refer to these levels as "00" or "level zero" (the base level, just described), "10" (mostly level A), "20" (mostly level AA), and "30" (mostly level AAA).

Fix and Reveal (FaR) level	Description
00	Foundational issues that interfere with use or are most likely to hide other issues. Overlaps WCAG 2.0 CR5
10	the rest of the level A Success Criteria that aren't already in 00 + a few AA ones that we selected as more important
20	the rest of the level AA Success Criteria
30	AAA Success Criteria

Taken as groups, generally the "level zero" issues will tend to obscure "level 10" issues which will in turn be obscured by "level 20" issues, etc.

Especially when taking an iterative approach to remediation, where issues are triaged and addressed in order of severity for successive releases to production sites, addressing the lower numbered groups of issues first will facilitate a number of desirable outcomes.

- Issues that affect more users or that have greater impact, will be addressed earlier.
- Testing and retesting processes are made more efficient and re-work is reduced.
- Progress toward the goals set forth in a business plan or a legal settlement can show continued, incremental improvement.
- Expectations of stakeholders can be managed ahead of concern over lists of defects that will not have a straight-line downward trend.

Once all defects have been revealed, one can then assign priority by impact.

Assigning Priority by Impact

A11y practitioners are, for good reason, in the habit of thinking about prioritizing by impact to users. What will have the greatest negative effect on the greatest number of users, categories of disabilities and assistive technologies?

One common example is keyboard-only functionality covering both sighted keyboard-only users and users of many Assistive Technologies such as screen readers. At Optum, we would consider this failure of 2.1.1 Keyboard Operation to have a "Critical" impact. (W3C, 2008)

Impact Level	Definition
Critical	User is completely prevented from accomplishing a task. There is no work-around.
High	User can perform task only through alternate path or workaround. Accomplishing the task likely only possible for power users.
Medium	User can perform task, but doing so will be frustrating or time consuming. User may require assistance from a co-worker or support staff.
Low	User experiences inconvenience or moderate frustration.

Software Development Methodologies

Agile vs. Waterfall

When undergoing a large-scale remediation effort, Agile's iterative approach lends itself well. Fixes can be introduced earlier, thus having a positive user impact right away. The waterfall approach delays fixes and thus prolongs the inaccessible user experience.

Separate Workstream vs. Integrated Workstream

A workstream dedicated to remediation fixes allows developers to focus on accessibility defects alone, without the distraction of other work. This separate workstream can be effective if you have enough developers and testers to support it. The danger here is that the other development work streams may introduce new accessibility errors while the previously identified issues are being remediated.

Keep in mind, that accessibility in the agile lifecycle always requires a multi-pronged approach: design with accessibility in mind, code with accessibility in mind, and test for accessibility. Doing any of these things in isolation will result in a constant cycle of rework. To ensure accessibility is taken into consideration at every phase, at Optum we embed accessibility engineers within agile teams.

Integrated workstreams, where accessibility fixes are incorporated with new development and fixing non-accessibility related bugs, can work when you don't have the resources for a dedicated accessibility team. Understand, however, that the remediation effort will take longer with shared resources.

Internal Team vs. Remediation Consultant

One might hire a remediation consultant if there is a lack of accessibility experience on the internal team. If the internal team is not educated in the issues being resolved, they will not improve their accessibility practices, and there will be a need for outside assistance.

If the design/development team is internal, it is beneficial to educate them on accessible practices so that they are not creating more projects requiring remediation work and creating additional accessibility issues on existing projects as they are updated.

Expense

The reputation that accessibility has for being "expensive and difficult" is typically from large-scale remediations where both can be the case compared to having created an accessible product from the outset.

Investing in design, development, and test teams to create an accessible product from the very beginning is far less costly than remediation after accessibility issues have been built into a product—not to mention the cost to your credibility when you need to go back to client and ask to change their designs and/or branding.

Remediation for the recently added WCAG success criteria has the potential to be much costlier. For example, if your site is not currently responsive, WCAG 1.4.10 "Reflow" will require a complete redesign effort. (W3C, 2018)

This additional effort may be a factor for clients' decisions on when to adopt WCAG 2.1 vs 2.0.

Summary

By re-thinking the prioritization and order of operations in addressing accessibility issues, digital properties can be made accessible with greater economy of effort, while making an iterative approach to remediation more effective in earlier stages of the process and initial releases of fixes.

The efficiencies gained by effective prioritization can substantially impact how quickly a site can be made accessible to the greatest number of users, in the shortest amount of time, with the least expense possible.

References

Kelly, Sean and Taylor, Bill (2017-03-03). *Success Criteria: Dependencies and Prioritization*. CSUN 2017, San Diego, CA. Retrieved from https://www.slideshare.net/seankelly612/csun-2017-success-criteria-dependencies-and-prioritization.

College of Information Studies, University of Maryland (2016) *Trace Center's Photosensitive Epilepsy Analysis Tool (PEAT) and Information on Photosensitive Epilepsy.* Retrieved from http://trace.umd.edu/peat.

W3C WCAG 2.0 Success Criteria and Techniques (2008-12-11). Retrieved from https://www.w3.org/TR/WCAG20/ and https://www.w3.org/TR/WCAG20-TECHS/.

World Wide Web Consortium (W3C), (2008) *W3C WCAG 2.0 Conformance Requirements, (2008)*. Retrieved from https://www.w3.org/TR/UNDERSTANDING-WCAG20/conformance.html.

World Wide Web Consortium (W3C). (2018-06-05). *Web Content Accessibility Guidelines (WCAG) 2.1. Success Criteria and Techniques*. Retrieved from https://www.w3.org/TR/WCAG21/.

Copyright Notice

A Case for a Risk-Based Sampling Approach to Manual Accessibility Testing

Mark Turner, Ph.D.

CEO
Optimal Solutions Group
Discovery District
5825 University Research Court, Suite 2800
College Park, MD
United States
mturner@optimalsolutionsgroup.com

Arthak Adhikari

Research Analyst
Optimal Solutions Group
Discovery District
5825 University Research Court, Suite 2800
College Park, MD
United States
aadhikari@optimalsolutionsgroup.com

Abstract

Automated testing can only assess up to 30 percent of the WCAG standards. To ensure that web products are compliant, it is necessary to conduct manual accessibility testing. Manual WCAG testing is labor intensive and costly. To ensure that organizations are able to routinely monitor web products' WCAG compliance that are statistically valid, Optimal Solutions Group ("Optimal") has designed an iAccessible module to generate a risk-based stratified random sample of webpages for manual testing based on the universe of webpages that had previously gone through automated testing. Embedded algorithms calculate standard errors and the resulting lower- and upper-bound compliance rate for domains. This paper makes the case that this approach is a more efficient way to guide webmasters and compliance managers on where to focus their remediation efforts.

Introduction

Automated testing can only assess up to 30 percent of the WCAG standards. To ensure that web products are compliant, it is necessary to conduct manual accessibility testing. The problem is, testing all a web product's content would be cost prohibitive and time consuming. Randomly selecting a statistically representative sample of webpages for manual testing can provide insight into web products' compliance rates but it does not provide insight about how to prioritize remediation efforts or the risks associated with non-compliance.

Optimal has designed a next generation approach by randomly selecting a stratified sample of web content for manual testing based on the following risk factors or strata:

- automated WCAG standard testing results
- volume of user traffic
- structure or format of webpages[1]
- frequency and intensity of web page structure/format/content changes

Embedded algorithms, designed by statisticians and data scientists, would enable iAccessible to generate risk-based strata, based on the outlined risk factors, from the population of an organization's webpages. The webpages within the sampled strata could then be manually tested and greater insights into high areas of non-compliance and where to allocate resources for remediation efforts could be provided to the webpages' webmasters and compliance managers.

The remainder of the paper is structured as follows:

- Literature review
- Random sampling approach
- Proposed methodology
- Conclusions and next steps

Literature Review

A similar paper by Rowland and Joeckel, titled "Validating a Sampling Process for Automated Accessibility Testing of Websites in a National Network", from the 2018 ICT Accessibility Symposium, looked at using automated testing and a simple random sampling technique to answer a dual research question. First, how do errors detected in a small sample of web pages correlate with the errors detected on the entire website. Second, how many webpages are needed to construct a 95 percent confidence interval of the entire website.

To answer their dual research question, the authors created two data sets based on 60 websites they deemed to be appropriate. For the first data set, they created a small sample of 14 webpages, for each website. Their second data set consisted of all the webpages for the 60 websites.

For the first portion of the research question, the authors correlated the errors in the small sample to the large sample. The average correlation, for all 60 websites, was 0.57. However, when they grouped the websites based on number of pages, they found the correlation fell as the number of webpages on a website increased.

[1] Webpage format may include the following categories: 1. Zig-Zag, 2. F layout, 3. Full Screen Photo, 4. Grid Layout, 5. One-Column Layout, 6. Featured Image Layout, 7. Asymmetrical Layout, 8. Split Screen Layout, 9. Headline and Thumbnails Gallery Layout, 10. Modular Layout, 11. Single Page Layout, 12. Radial Symmetry Layout

To answer the second research question, the authors used a bootstrapping method. After calculating the means, they found that 42 percent of all websites' webpages had to be scanned to create the desired 95 percent confidence interval.

The authors mention the added benefit of manual accessibility testing but lamented that the labor intensity of such a venture would make it cost prohibitive. Optimal believes that its methodology of automatically selecting a sample of webpages for manual accessibility testing combined with an integrated manual testing workflow and reporting functionality could address this problem.

Random Sampling Approach

The first portion of the dual research question in Rowland and Joeckel's paper required the creation of a sample. They created the sample by identifying the home page for each of the 60 websites and then using a random number generator to select an additional 13 pages for a total sample of 14 webpages for each of the 60 websites. The total sample size was therefore 840 webpages (60 multiplied by 14) and since the mean average per website, in their population, was 644 their sample constituted 2.17% of the entire population (844 divided by 38,640) (Rowland and Joeckel, 2018).

The above methodology is an example of a random sampling approach, specifically simple random sampling without replacement (SRSWOR) (Pathak, 1988).

For SRSWOR, observations are selected from a population in such a way that every observation in the population has the same probability of being selected for the sample. Sampling is stopped when the total sample size is reached, and no observation can be picked twice for the sample. This is the case in the above example for the 13 webpages randomly selected by Rowland and Joeckel.

The figure below depicts a visual representation of this type of sampling (Thompson,2012):

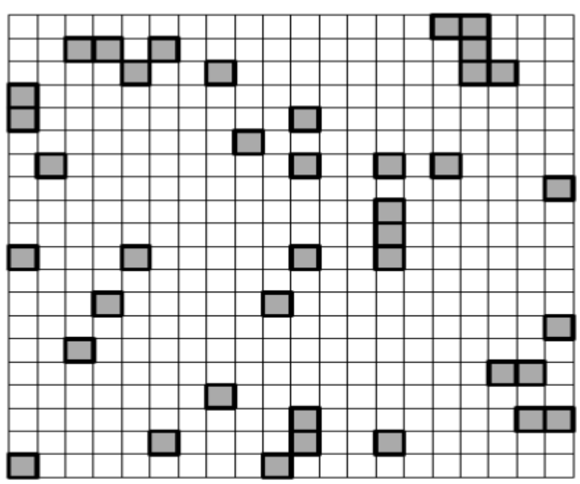

Figure 1: Random sample of 40 observations from population of 400

SRSWOR sampling is beneficial in that it is conceptually simple and, when done correctly, it can result in unbiased estimates of population parameters. It, however, does not provide insight about how to prioritize remediation efforts or the risks associated with non-compliance. This weakness is particularly evident if the underlying data is skewed.

In the specific scenario of sampling webpages for accessibility issues, skewness of data are a reasonable assumption. For example, it is reasonable to assume that webpages that are dynamic, i.e. constantly changing content, would have higher rates of WCAG testing errors than webpages that are static.

To illustrate why skewness of data are important please see the figure below:

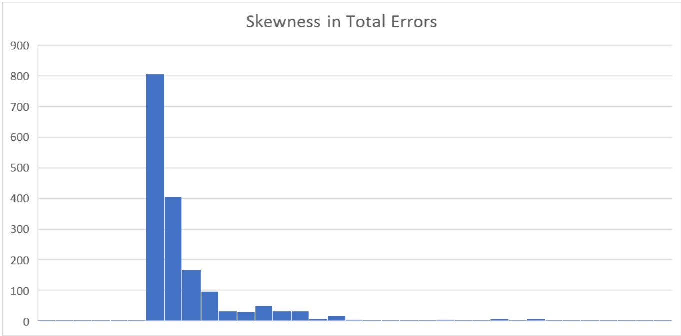

Figure 2: Total 508 Errors on Website

The histogram is produced using data from an Optimal client. Any identifiable information was removed in order to maintain confidentiality. Using WAVE, Optimal conducted automated WCAG tests of 1,727 webpages of a client website. The histogram shows the distribution of "total errors" (errors plus contrast errors as identified by WAVE).

We see visually from the histogram that the data are skewed. Calculating the skewness measure from the population, we get 3.68 which indicates highly skewed.

In deriving a sample of webpages for manual testing using this approach we would only gain insights into the compliance rates of these webpages as a whole but not much else about important components or strata within the population. For example, in this population, there are 34 webpages with greater than 150 total errors based on automated WCAG testing. Since in random sampling each observation has the same probability of being chosen, assuming a sample size of 100, we would only expect two of these high error pages to be included in the sample. This is problematic within the context of accessibility testing. In this example, the highest number of automated WCAG errors on a webpage is 237; a simple random sampling approach does not differentiate selection between webpages that had zero detectable errors and those that had many (in this case 237) errors as determined by automated WCAG tests.

To reiterate, using a simple random sampling approach is appropriate when clients are only interested in the overall rates of compliance. However, if clients are interested in insights on how to focus remediation efforts then using a stratified sampling approach may be more cost-effective despite being more difficult to administrator than simple random sampling. The methodology for stratified random sampling is described in the proceeding section.

Proposed Methodology

This paper outlines two distinct approaches determining strata or assignment to a subgroup and then the sample size determination for each stratum.

Proportional stratification. In this case, the size of the strata is proportionate to the size of the subgroup in the population. For example, if strata were based on webpage structure and it was determined that a domain used had three distinct layouts (e.g., Featured Image Layout, One-Column Layout, Modular Layout). During the automated testing phase, embedded algorithms determine distinct types of webpage layouts or formats and the proportion within the domain. Based on these parameters and the selected level of confidence (say 95%) the resulting sample size is determined.

Webpage Layout Stratum	Population size	Standard deviation	Mean
Featured Image Layout			
One-Column Layout			
Modular Layout			

Disproportionate stratification. With disproportionate stratification, the sample size of each stratum does not have to be proportionate to the population size of the stratum.

iAccessible Algorithms and Workflow

Implementing these sampling approaches outlined above are technically complex. Embedded algorithms and configured workflows in the iAccessible platform attempt to simplify the implementation of risk-based stratified random sampling approach that includes the following steps:

1. *Universe of webpages*. A webcrawler scans and catalogs webpages within the bounds of the targeted domain to a specified depth level.
2. *Construct analytic data table*. The cataloged list of webpages is joined with other data that include risk-based metrics including:
   - automated WCAG standard testing results;
   - volume of user traffic;
   - assessment and categorization of web page structures; and,
   - frequency and intensity of web page structure/format/content changes.
3. *Select risk factors*. User selects the specific subset of risk factors.

4. *Pre-defined strata*. iAccessible's embedded algorithms create pre-defined strata based on distribution levels (e.g., 90th percentile of user volume).
5. *Manual testing assignment*. iAccessible automatically assigns selected webpages to the reviewer pool.
6. *Reporting*. iAccessible automatically generates compliance reports that includes confidence intervals for both automated and manual testing results.

Conclusions and Next Steps

Stratified random sampling has the potential to generate more precise manual testing results than manual testing that uses simple random sampling. Moreover, using stratified random sampling also enables clients to identify specific subgroups are resulting in violating WCAG standards.

The following conditions must be met to select a sampling based on stratified random sample:

Automated scrawl and assessment or other automated assessments much be able to identify each subgroup or strata and classify each of them into only one subgroup. Defining an exhaustive and definitive list of options and categorizes them in an automated fashion in some cases may be difficult (e.g., webpage layouts).

Implementing an approach that involves two or more strata simultaneously becomes increasing complex and increases the resulting sample size to achieve the same level of confidence.

Next Steps. Optimal is currently piloting single strata sampling approach on the Revelo-powered iAccessible platform.

Copyright Notice

Investigating Factors that Affect Web Accessibility Implementation in Non-Profit Websites

Brittani S. Washington

Department of Computer and Information Sciences
Towson University, Towson, MD, United States
bwashington@towson.edu

Jinjuan Heidi Feng

Department of Computer and Information Sciences
Towson University, Towson, MD, United States
jfeng@towson.edu

Abstract

Non-Profit organizations (NPOs) continue to fall short with properly implementing accessibility features. This study aims to understand possible factors that may affect the accessibility implementation of non-profit websites through data collected using two automated accessibility assessment tools, namely the Functional Accessibility Evaluator (FAE) and the WAVE Web Accessibility Evaluation Tool (WAVE). The factors examined in this study include geographical location areas, types of NPOs, and the number of webpages contained in the website. Quantitative analysis was conducted to examine the impact of each of these 3 factors on both the overall accessibility scores and the number of violations in 9-rule categories identified by the FAE. The result suggests that the types of NPOs have significant impact on the FAE score and the number of violations in the 'site navigation' category.

Introduction

Web accessibility is defined as "focusing on how a disabled person accesses or benefits from a site, system or application" (U.S Department of Health & Human Services, 2008). The range of things that accessibility touches continues to grow as a result of new awareness's unfolding by the exposure of the lack of inclusion. Web Content Accessibility Guidelines (WCAG) 2.1 and Section 508 provides standards and recommendations for making web content more accessible, but organizations especially in the Non-Profit Sector continues to fall short of abiding by those standards and recommendations.

The Non-Profit is a group of organizations who raise monetary funds or other resources in order to carry out their missions. According to the Internal Revenue Service Databook, in 2017 approximately 1.79 million NPOs were registered with the Internal Revenue Service (IRS), which is an increase of 15% compared to 2015 (Kautter et al, 2018). Out of those 1.79 million organizations, C3 organizations represented approximately 71.5%, the highest percentage of registered NPOs (Kautter et al, 2018). C7 organizations represented approximately 2.73% of the total NPOs (Kautter et al, 2018). According to the Internal Revenue Publication 557, 501(c)(3) organizations are those described as being affiliated with one or more of the following types: Religious, Educational, Charitable, Scientific, Literary, Testing for Public Safety, to Foster National or International Amateur Sports Competition, or Prevention of Cruelty to Children or Animals Organizations and their contributions are deductible (Department of the Treasury, 2019). 501(c)(7) organizations are those described as being affiliated with Social and Recreational Clubs and their contributions are not deductible (Department of the Treasury, 2019).

We have conducted a preliminary study to understand the current status of the implementation of Web Content Accessibility Guidelines in the non-profit sector (Washington and Feng, 2018). In this study, we expanded the scope of the previous study to 96 non-profit websites and collected more data so that we can conduct quantitative analysis to understand the relationship between various factors, by focusing on the following objectives:

1. Examine whether the overall accessibility implementation scores are affected by the specific types of the organization, the geographical area, and the size of the website
2. Examine whether each accessibility rule category (e.g., forms, headings) is affected by the specific types of the organization, the geographical area, and the size of the website

Research Method

We analyzed a sample of 96 NPOs based on the criteria identified in the 2018 study, "Web Accessibility: Where is the Non-Profit Sector" (Washington and Feng, 2018). Those criteria are geographical location areas, non-profit types, and the number of webpages. The geographical location areas are grouped based on the targeted service area of the organization: local, state, regional, or national. The non-profit types are grouped as 501(c)(3) and 501(c)(7) organizations. The C3 organizations were categorized as religious, educational, or charitable. The C7 organizations were categorized as combined social and recreational clubs. The number of webpages is grouped as 50 pages or less, 51-100 pages, or 101 or more pages.

Two automated tools were used to analyze the websites, namely the Functional Accessibility Evaluator (FAE) and the WAVE Web Accessibility Evaluation Tool (WAVE). These tools aim to identify the presence of accessibility features. Compared to the 2018 study, we doubled the sample size of the websites from 48 to the 96 used in this study. The other difference between the two studies is the number of webpages that were analyzed. The maximum number of webpages that were analyzed by the FAE tool in the 2018 study was 25 webpages. In this study all webpages on each website were analyzed using both tools.

The FAE tool computes an implementation score for each of the 132 individual rules which represents the compliance requirements derived from the WCAG Success Criterion using the following equation: the number of passed items divided by the sum of the number of passed items, the number of failed items, and the number of items requiring manual check multiplied by 100 (Lazar et. al. 2017).

$$RIS = \frac{P}{P + F + MC} \times 100$$

The overall implementation score is the average of the implementation scores of the individual rules. The rules were grouped into 12 categories; audio/video, forms, headings, images, keyboard, landmarks, links, site navigation, styles/content, tables, timing, and widgets/scripts to serve as a way to organize the analysis results returned by the tool. However, audio/video, keyboard, and timing were not categories used in this evaluation due to the fact that they required manual testing. Each rule category was represented by the total number of violations associated with each particular rule category. This number served as a dependent variable in determining significance of accessibility implementation within the rule categories later on in the study.

Category	**Definition**
Landmarks	The use of landmarks will, in many cases, reflect the visual styling and page layouts that web designers utilize to set apart various sections of content.
Headings	Use heading elements (H1-H6) to provide appropriate labels for landmarks, and to identify subsections of content within landmarks.
Styles/Content	Ensure that text is readable by adhering to color contrast guidelines, and that information is not conveyed solely by the use of color, shape, location or sound.
Images	Provide appropriate text alternatives for static images and graphics.
Links	Use link text that properly describes the target of each link.
Tables	Provide table captions or other meta-information as needed. Provide row and column header references for data cells of data tables.
Forms	Provide meaningful labels for form elements and usable and understandable error feedback as needed.
Widgets/Scripts	Ensure that custom widgets created using JavaScript support keyboard interaction and include ARIA markup to describe their roles, properties and states.
Site Navigation	Ensure the consistent labeling and ordering of recurrent page sections across all pages within a website.

Table 1: FAE rule categories analyzed in this study and their definition. (University of Illinois, 2018).

The WAVE tool analyzes compliance issues found based on Section 508 and the WCAG guidelines (Smith and Whiting, 2001). The WAVE score used in this study is the total number of errors identified by the WAVE tool.

Results

Functional Accessibility Evaluation Tool

Accessibility Implementation Scores

We conducted One-Way Analysis of Variance (ANOVA) tests to examine the impact of geographical areas, types of NPOs and the sizes of the websites on the FAE website accessibility implementation scores. ANOVA tests with the FAE website implementation score as the dependent variable and geographical area as the independent variable suggests that there is no significant difference in the FAE scores between the NPOs from different geographical areas ($F(3, 95) = 1.15$, n.s.).

ANOVA tests with the FAE website implementation score as the dependent variable and types of NPOs as the independent variable suggests that there is significant difference in the FAE scores between different types of NPOs ($F(3, 95) = 3.04$, $p < 0.05$). Post-hoc Tukey tests suggest that the FAE scores of the educational organization websites are significantly higher that the scores of the social club websites ($p < 0.05$). There is no significant difference in the FAE scores between the charitable websites, the educational websites, and the religious websites.

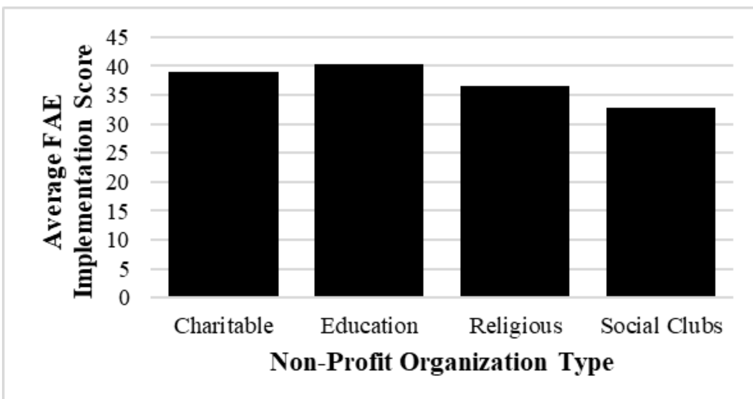

Figure 1 – The average FAE Website Implementation Score for each type of Non-Profit Organization

ANOVA test with the FAE website implementation score as the dependent variable and the website size (number of pages) as the independent variable suggests that there is no significant difference in the FAE scores between the NPOs websites of different sizes ($F(3, 95) = 0.27$, n.s.).

Accessibility Implementation Rule Categories

We conducted ANOVA tests to examine the impact of geographical areas, types of NPOs, and the sizes of the websites on the number of violations under each rule category. ANOVA tests with the number of violations under each of the 9-rule category as the dependent variable and geographical area as the independent variable suggests that there is no significant difference in the number of violations between the NPOs from different geographical areas for any of the 9-rule categories.

ANOVA tests with the number of violations under each of the 9-rule category as the dependent variable and types of the NPOs as the independent variable suggests that there is highly significant difference in the number of violations between the different types of organization for the navigation rule category ($F (3, 95) = 6.59$, $p < 0.001$). Post hoc Tukey tests suggest that the number of violations for the social club websites are significantly higher than that of the educational websites ($p < 0.001$) and the charitable websites ($p < 0.05$) (Figure 2). No significant difference is observed in the number of violations between the different types of organization for the other 8 rule categories.

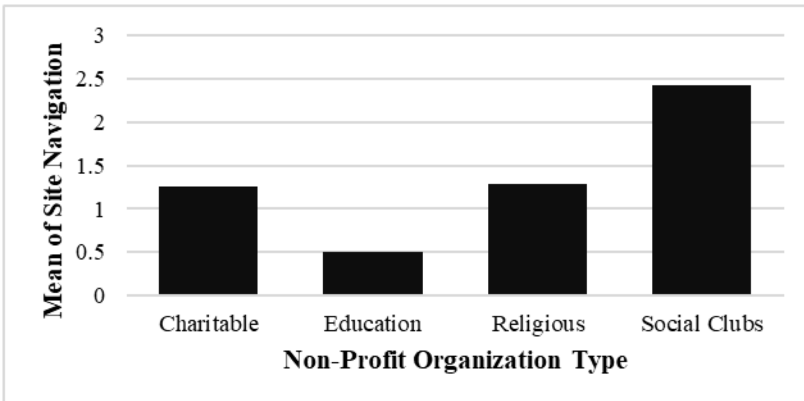

Figure 2 – The average number of violations under the site navigation rule category for each type of Non-Profit Organization

ANOVA tests with the number of violations under each of the 9-rule category as the dependent variable and website size as the independent variable suggests that there is no significant difference in the number of violations between the non-profit websites with different sizes for any of the 9-rule categories.

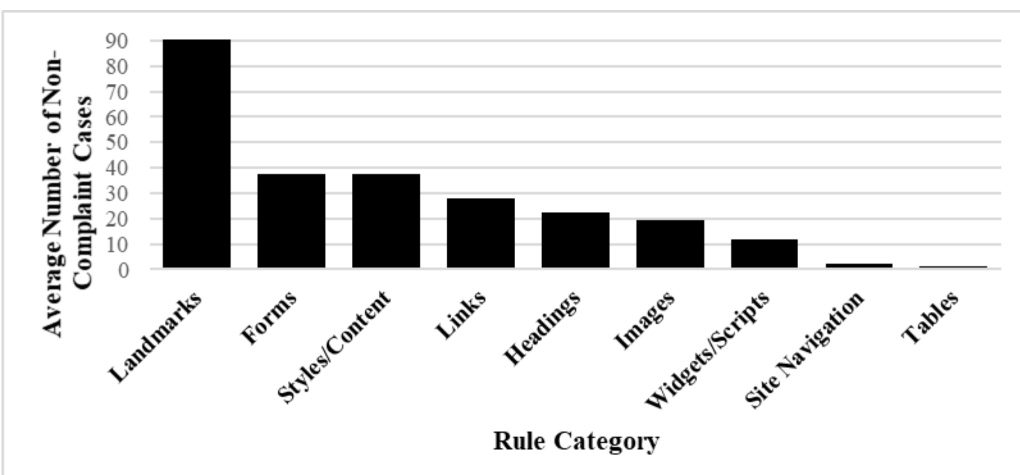

Figure 3 – Non-Complaint Cases by Rule Category

We also conducted a Repeated Measures ANOVA test to study whether there is any significant difference in the number of violations among the 9-rule categories. The test suggests that there is a highly significant difference among the 9-rule categories ($F(8, 760) = 34.55$, $p < 0.001$). Post hoc tests suggest that there is significant difference in the number of the violations between all categories except for 3 pairs: 'Forms' and 'Style/content'; 'Forms' and 'Links'; and 'Links' and 'Headings' (Figure 3).

WAVE Web Accessibility Evaluation Tool

In comparison, we conducted ANOVA tests to examine the impact of geographical areas, types of NPOs, and the sizes of the websites on the WAVE score. Three ANOVA tests were conducted using the WAVE score as the dependent variable and geographical area, type of NPO, and website size as independent variable respectively. The result suggests that there is no significant difference in the WAVE scores between the NPOs from different geographical areas ($F(3, 95) = 1.54$, n.s.), organization of different types ($F(3, 95) = 1.44$, n.s.), or organizations with different sizes of websites ($F(3, 95) = 2.61$, n.s.).

Discussion

The continuance of the study shows that NPOs still lack proper web implementation of accessibility features that are used to improve the interaction of those that may have a disability. As it relates to the impact of the geographical area, neither the FAE analysis nor the WAVE analysis revealed significant difference. This finding suggests that the level of accessibility implementation among NPOs is not affected by the geographical areas that the organization covers. National, regional, state, and local organizations all need to improve their accessibility practice.

Regarding the impact of the types of NPOs, we found significant relationship between the type of the organization and the FAE scores. The social and recreational organizations; which are categorized as a C7 type, had significantly lower FAE implementation scores than the educational organizations which are categorized as a C3 type organization. The majority of the social clubs used in the study have a specific target audience. They do not operate on a catchall approach to compete with the chains. They pick their target audience and stick to it. Whereas the educational based organizations are attempting to reach out to the general public in order to bring awareness on a specific topic. In order to attract the general public, they have to accommodate to the needs of a much broader audience. Compared to the C3 type organizations, the C7 type websites tend to have something in common regarding their use of images and widgets and scripts. A high volume of scrolling images and scrolling text are largely visible on the home pages of these websites which could result in increased number of failed items in those categories.

The analysis of the impact of the geographical location, non-profit type, and website size on the number of violations under 9-rule categories suggests that the type of NPOs has significant impact on the number of violations related to 'site navigation'. The post hoc tests revealed that the social club organizations had significantly more navigation violations than the educational organizations and the charitable organizations. Site navigations should only display short text that link over to the full content of that information topic. Site viewers go to educational websites in search of specific topic; such as, admission into the institution, accessing certain student related resources, and more. If the site navigations are not clear and precise it substantially increases the level of difficulty when visiting the site. Educational organizations tend to do a better job at allowing their audience to access and find the information they need quickly. In contrast, many social clubs do not have the same focus. Social clubs reach majority of their audience via effective word-of-mouth marketing, because people tend to be more willing to join social clubs through the recommendation of someone that they deem trustworthy, credible, and has something interesting to say (Kelly, 2007). The different foci in reaching out to target audience might have contributed to higher number of violations in the navigation category for social clubs. However, with the urgent need to appeal to the millennials whose decisions are heavily influenced by information online, improving the navigation design may help attract new members to these organizations.

There is highly significant difference in the number of violations between the 9-rule categories. The number of violations under the 'landmark' category was more than double of any of the other categories. The FAE tool identifies landmark errors when roles; such as, main, navigation, banner, etc. are missing in places where they are needed. The Landmark rule category is also the second highest rule group evaluated in the study with the 2^{nd} highest number of rules analyzed, closely following the Styles/Content rule category. Landmarks are used to identify sections of a webpage with similar content. It allows developers to create a space where users of assistive technologies could navigate more efficiently and bypass repetitive content (Dodson, 2019). Without the proper use of landmarks, users of assistive technologies have a difficult time identifying the main content of a page (Washington and Feng, 2019). This is due to the fact that there is no programmatic structure to a webpage.

Conclusion

With this study we aimed to continue to bring awareness to how the lack of proper implementation impacts those that benefit from accessibility inclusion. Analyzing these websites painted the picture that they were not developed and designed with accessibility in mind, which increases the level of frustration amongst users with disabilities. According to the 2017 IRS Databook, the number of NPOs who file continues to grow each year. This means that the amount of services provided by the non-profit sector continues to grow. Accessibility barriers to those services are likely to have a significant impact on those with disabilities in areas where that community mostly utilizes; such as educational resources. The result of the study suggests that the educational organizations out-performs the social organizations in both the FAE implementation scores and the site navigation category. We will conduct follow-up studies to validate this finding. If the finding is confirmed by further evidence, we will investigate the contributing factors to the positive trend in the educational organizations so that it might be generalized to other types of NPOs.

Regarding the limitation of the study, although automated testing allows testers to run tests faster than human users and allows for easier repetitions of executing the same operations, it only accounts for 10-30% of the identified errors (Kuykendall, 2017). Manual testing will need to be conducted in future research in order to analyze violations within those rule groups that automated testing cannot detect such as audio/video, keyboard, and timing. The results of manual testing could also be used to support the justification of the results identified by automated testing. The importance of testing using assistive technology tools paints a more accurate picture of the effect these websites have on those that utilize the tools. It may also be beneficial to include additional tools for automated testing in order to fully compare the results. We acknowledge that the sample size in this study is still a low percentage of the total number of non-profit C3 and C7 filings reported to the IRS. Future studies are needed to investigate the related factors, especially the type of NPOs on a substantially larger scale.

References

Dodson, R. (2019). How To Do an Accessibility Review | Web Fundamentals | Google Developers. Retrieved from https://developers.google.com/web/fundamentals/accessibility/how-to-review

Kautter, D., Herndon, B., Johnson, B., Paris, D., and Kei, W. (2018) Internal Revenue Service 2017 Data Book. 2017 Publication 55B, from https://www.irs.gov/pub/irs-soi/17databk.pdf

Kelly, L. (2007). Beyond buzz the next generation of word-of-mouth marketing. New York: AMACOM.

Kuykendall, H. (2017). Automated Web Accessibility Testing Tools: The Strengths and Weaknesses of Automated Checkers, from https://www.microassist.com/digital-accessibility/role-web-accessibility-testing-tools/

Lazar, J., Gunderson, J., and Golderfeld, Z. (2017). Federal Website WCAG 2.0 Compliance and Accessibility Trends. In: Proceedings of 2017 ICT Accessibility Testing Symposium: Automated & Manual Testing, WCAG 2.1, and Beyond. 27-34.

Smith, J., & Whiting, J. (2001). WAVE Web Accessibility Evaluation Tool. Retrieved from http://wave.webaim.org/

University of Illinois. (2018). Functional Accessibility Evaluator 2.0. Retrieved from https://fae.disability.illinois.edu/

U.S. Department of Health & Human Services (2008). Accessibility of State and Local Government Websites to People with Disabilities, from https://www.ada.gov/websites2.htm

Washington, B. and Feng, J. (2018). Web Accessibility: Where is the Non-Profit Sector? In: Proceedings of the 2018 ICT Accessibility Testing Symposium: Mobile Testing, 508 Revision, and Beyond. pp.165--172

Washington, B. and Feng, J. (2019). Proper Implementation of Website Features Affecting the Use of Screen Readers. In: Ahram T., Falcão C. (eds) Advances in Usability and User Experience. AHFE 2019. Advances in Intelligent Systems and Computing, vol 972. Springer, Cham. pp. 813-823

Copyright Notice

How WCAG 2.1 benefits people with cognitive impairment

Valorie Sundby
Digital Accessibility Engineer
Optum
11000 Optum Cir, Eden Prairie, MN 55344, USA
valorie.sundby@optum.com

Karen Herr
Digital Accessibility Engineer
Optum
11000 Optum Cir, Eden Prairie, MN 55344, USA
karen.herr@optum.com

Vanessa Howle
Digital Accessibility Engineer
Optum
11000 Optum Cir, Eden Prairie, MN 55344, USA
vanessa.howle@optum.com

Abstract

As accessibility practitioners, the more we understand the conditions of impairment, the better we are at understanding what we test for and why. In this paper, we examine the impacts of cognitive impairment, including how people intake and process content presented on the web and in other digital formats. Specifically, we look at the concepts of cognitive load: intrinsic, extrinsic, and germane. The W3C Cognitive Accessibility User Research (https://www.w3.org/TR/coga-user-research/#reasoning-and-executive-functions) is a public working draft, published January 15, 2015, presented by the Cognitive and Learning Disabilities Accessibility Task Force (COAG). The COAG research provides information about people with select cognitive disabilities. We will build on the research of COAG and other professionals to explore and collate how the WCAG 2.1 success criteria enable users to mitigate cognitive load.

Cognitive impairment categories

Cognition is the mental action or process of acquiring knowledge and understanding through thought, experience and the senses (Oxford University Press, 2019).

There are many conditions and situations that impact cognition and create cognitive impairment. These conditions include: impairments to the senses, damage to the brain, health disorders and illness, genetics, medications and medication side-effects, psychological trauma and unresolved trauma, proximity to trauma in time or location, learning disabilities, and situational environments.

Cognitive impairments have been categorized to understand the wide range of experience an affected person can have:

Cognitive malfunction

Cognitive malfunction conditions diminish the ability to receive or process information, or alter the way information is processed in the brain. Cognitive malfunction includes impairments such as learning disabilities, neurodegenerative disorders and disease, psychological disorders, injury or impairment of senses or sensory intake processes.

People with cognitive malfunction may adopt alternate methods of processing information. The alternative method is independently learned or learned and practiced with a professional. Assistive technology can be part of these alternate methods.

Cognitive instability

Cognitive instability refers to a variance of cognitive ability (or impairment) occurring over a short period. Short term impacts include memory loss, confusion and loss of coordination. Individuals who experience cognitive instability may recover from the impairment when conditions or situations improve.

When the situation or condition is persistent, individuals may experience permanent or extended impairment. Extended impairment may include forgetting names of people or objects, an increase in emotional instability and decrease in self-control (PsychGuides.com, 2019).

Causes of cognitive instability include high stress, lack of control, excessive/lack of noise or stimulation, panic or substance use/abuse. Depression can result in cognitive instability. Physicochemical changes also cause cognitive instability including insulin fluctuations, thyroid imbalance, or medication.

Waiting for cognition to improve or the condition to pass is the most common strategy for people with cognitive instability. Developing alternate methods to process information may not be used if the person knows that cognition will improve.

Progressive cognitive decline

Progressive cognitive decline is a deterioration of cognitive ability over time. Diseases and situations that cause progressive cognitive decline include Alzheimer's and Parkinson's. Substance use or abuse over time can cause progressive cognitive decline.

Interview with an assistive technology trainer: Roberto Perez

We interviewed Roberto Perez, assistive technology instructor for the blind, during which he shared his process and observations from working with someone who is both blind and has a cognitive impairment.

- First, find a routine or process with which the client is comfortable and try to repeat this process as many times as needed.
- Be prepared with recordings and written documentation of the training. Depending on the situation, some people don't feel comfortable with recordings, whereas others are more into the recordings and won't require written documentation.
- Reduce all possible distractions. Remove any applications (i.e., email and instant messenger notifications) running in the background that may distract the person.
- Provide a cheerful environment. Share humorous stories to make the person laugh and put them at ease. Tune the environment to the person so they can relax.
- Manage the amount of content. Cover one topic or review until the person is comfortable.

> "The cognitive impairment will always be there. Usually, when there is cognitive impairment, there is a level of anxiety that mounts quickly and stress can develop quickly. So perhaps, by creating a very positive environment, I can minimize anxiety. Of course, the most effective way is to find what works for each particular client. That's always very unique based on the individual." (Perez, 2019).

Assistive technology for cognitive support: Vanessa Howle

For me, with my Dyslexia, having personalization options is important. These options include margins, text spacing, text size, and background color. I can define and select specific combinations of text or words to highlight, which makes the page easier to use and more comfortable for me to read. Reading tools, where I can set the speed, select the block of text to read, control the volume, and block background noise, are important.

I have found that WYNN from Freedom Scientific is a great tool for individuals with Dyslexia. I have replaced my use of Kurzweil 3000 with Wynn due to its ease of use. While using both, and being a long time end user of Kurzweil 3000, I have found WYNN is much easier for use and understanding with my learning disability. The WYNN tool bars are highly intuitive and easy to learn compared to Kurzweil 3000.

Kurzweil 3000 is good for individuals who need a combination of a scanner and software that allows others with Dyslexia an easier experience in reading and understanding books, magazines, or notes.

Both WYNN and Kurzweil 3000 have the following features which are helpful for me:

- Word definitions and access to various synonyms and antonyms of words in the text
- Highlighted text in different colors similar to what you can do in a physical document
- Ability to apply notes to the page and then extract the information for study guides

Assistive technology and accessible digital content has helped me be successful in school, my career and able to enjoy pleasure reading.

How can WCAG 2.1 AA compliance help people with cognitive impairment?

There is no way to know an individual's cognitive ability or impairment when they visit a web page or digital content. We can, however, minimize the cognitive load caused by the content presentation. Meeting the WCAG 2.1 success criteria provides users with more choices so they can take the path of least resistance.

What is cognitive load and how does it impact cognition?

Cognitive load refers to the effort placed on working memory resources and may be defined as three types: extraneous, intrinsic, and germane (Sweller, 1988).

Extraneous load is the effort it takes to perceive information as it is presented in a particular format. Things added to a website that misdirect a visitor's attention from a task, such as colorful advertisements flashing content, or varying and inconsistent fonts, add to the extraneous load.

Intrinsic load is the effort it takes to process information on a particular topic. The intrinsic cognitive effects of brain injury include slowing the "speed of thought, memory and understanding, concentration, solving problems and using language." (Headway: the brain injury association, 2019). When the information on a topic is presented in an organized, structured and consistent manner, the intrinsic load can be reduced.

Germane load is the effort it takes to permanently process information into a retrievable format in the brain. Memory schema, the conceptualization of content, assists the brain to remember (Pschologist World, 2019). Consistency in web content presentation, well-organized and good semantic markup allows assistive technology and other user agents to represent content in alternative structures and allow the use of organizational tools.

WCAG 2.1 success criteria and the impact on extraneous load

Reduce extraneous load by eliminating distractions and allow customization, such as allowing visitors to apply their own color palette or font. Designers, developers and content creators have the most control over extraneous load.

- 1.2.2 Captions, 1.2.4 Live Captions - Accurate captions that have proper capitalization and punctuation is critical to understanding the meaning of written text and ensuring that it matches what is being said. The intake of information through two senses, (hearing and sight) is a common strategy of people with cognitive impairment. When the captions are poorly edited with misspellings, missing capitalization and punctuation, the extraneous load for the visitor increases.
- 1.3.4 Orientation - Continuous eye tracking can cause fatigue. Providing a mobile user, the ability to view an application in landscape view, and thus with more characters per line, can lessen the extraneous cognitive load. Users that increase the text size on their devices see the line lengths become even shorter and are more likely to use a landscape view (Butler, 2018).
- 1.4.10 Reflow - Horizontal scrolling to read text is an extra, unnatural step that disrupts the reader’s flow, attention, and comprehension. By reflowing the content at 400%, rather than requiring horizontal scrolling, the user can focus on the content, rather than the scrolling (Wayne e Dick, 2017).
- 1.4.13 Content on Hover or Focus - Content that displays on hover or focus should be dismissed when the user chooses to dismiss it. When content dismisses without user control, the user may not have had enough time to process the content (Shawn Lawton Henry, Editor, 2019).
- 2.2.1 Timing Adjustable - Information processing delays are a common consequence of cognitive disorder. Timeouts without warning or ability to extend time can disorient people with cognitive impairments. Extraneous load is created from being logged out of an application because it appears that the site is idle when the visitor is taking the extra time needed to comprehend content (Shawn Lawton Henry, Editor, 2019).
- 2.2.2 Stop Pause Hide - Animated content is a distraction for all users. For people with cognitive impairment, moving, updating or flashing content can prevent them from completing a task or finding content for which they are visiting the site (Sethfors, 2017).
- 2.5.2 Pointer Cancellation - The accidental clicking of a control can be caused by distractedness, tremors, a lack of understanding, etc. Having the ability to undo an action can be imperative, particularly for financial or legal transactions.
- 3.1.1 Language of the Page, 3.1.2 Language of the Parts - For all screen reader users, proper language definition ensures proper pronunciation, inflection and syllabic emphasis to the best of a screen reader's ability. When the content is read properly, all people and especially those with cognitive impairment benefit. When content is read correctly, the extraneous load for all screen reader users decreases.

- 2.4.3 Focus Order, 3.3.2 Labels and Instructions - Testing a custom component for good focus order, labels and instructions should cause us to think about the complexity of the component and the impact on extraneous load. "Rather than trying to force a complex task to become accessible and usable, which instead results in something that is technically conformant with WCAG, but not usable in the real world, one needs to rethink the task into a more accessible and usable model that achieves the same goal." (Ali I. , 2018).
- 2.6.1 Motion Actuation - Allowing the visitor to disable motion actuation reduces the possibility that content will be accidentally triggered. Unexpected changes in context can be unsettling at best and derailing for people with cognitive impairment.
- 3.3 Input Assistance - It is important that people are provided with enough information so they can succeed at a task the first time. The impact of errors can increase anxiety, contribute to low self-confidence or trigger depression. Providing formatting and other instruction so users can succeed the first time reduces extraneous load.
- 4.1.3 Status Change - For users of assistive technology, moving page visible or keyboard focus unnecessarily can cause distraction and the user can become lost in the page. Having a status change announced by screen readers without having focus moved, allows the user to retain their place in the page and continue their workflow.

WCAG 2.1 success criteria impact on intrinsic load

Intrinsic load is reduced through simplification: less content per page, smaller steps in a process, or guided steps through complex processes. The following table explains the correlation between the success criteria and intrinsic load.

- 1.1.1 Alternative Text - Cognitive support images: A common strategy to reduce intrinsic load is the use of cognitive support images, which are images paired with text. The images are considered decorative and should be ignored by assistive technology (AT) as the visible text is the alternative. The simplification of written language with the use of supporting images supports the ability to take in information properly. (System Concepts, 2018).
- 1.3.5 Identify Input Purpose - Providing programmatic association of inputs to browser or AT data enables users to pick from a list of previously stored data or modify the presentation of content to a user's specifications. The auto-complete functionality eases memory efforts and can reduce anxiety (System Concepts, 2018).
- 1.3.1 Info and Relationships - Semantic markup: How content maps to AT functionality such as lists of links, headings, buttons, and regions is vital for cognitive support. In addition to screen readers, AT such as Kurzweil 3000 and browser add ons/plugins may be used to assist with visualizing content hierarchy; producing mind-maps, organizing notes, and completing tasks. Sound semantic structure enables AT to perform at its best to simplify and adapt content through the use of AT (Professor Barbara A Wilson, 2018).

- 1.3.1 Info and Relationships, 2.4.6 Headings and Labels, 2.4.4 Link Purpose, 3.3.2 Labels and Instructions - "Errorless learning" is a useful strategy for people with memory impairment (Professor Barbara A Wilson, 2018). To provide errorless learning for a person completing a task such as registering for Medicare or support services, the most important question for testers is, “Can the visitor complete the task on the first try?” This question helps the tester consider more questions: Is the information and relationships visually (and for AT users) programmatically associated to inputs? Is sufficient instruction provided for formatting answers? Do labels make sense?
- 1.4.12 Text Spacing - Increases in leading (the space between paragraphs) has been shown to improve text reading performance. Providing additional spacing between paragraphs allows for chunking of content. Chunking is a term from cognitive psychology, which when applied to text is the technique of formatting in smaller blocks that are less intimidating to view and easier to understand (Moran, 2016) (Marco Zorzi, 2012).
- 2.2.1 Timing Adjustments - The time adjustment success criterion is probably the most important for people with cognitive impairment. Allowing for time extensions, timeout removal or reset is vital for people who need more time to process information. Slower processing affects reasoning and executive function, memory, understanding language and perception (Cooper, 2015).
- 2.5.3 Label in Name - When actionable content is consistently identified between the visual presentation and the accessible name, voice control users can successfully operate controls (Watson, 2017).
- 3.2.3 Consistent Navigation, 3.2.4 Consistent Identification, 4.1.2 Name, Role, Value - Consistency in navigation and identification is important for people with cognitive impairment in providing predictability. Knowing that menus will be in the same location and in the same order reduces the time it takes to navigate to content. When content is identified consistently (buttons look like and are identified as buttons, links look like and are identified as links) visitors who use screen readers for cognitive support can reliably interact with components as presented [Optum Best Practice; Optum DPL – mobile design].
- 2.5.1 Pointer Gestures - Unless essential, all multi-point or path-based gestures must allow single-point activation. Simplification of gestures reduces the intrinsic load necessary for activating controls or interacting with content by reducing the need to remember complex motions (Mattes, 2018).
- 2.6.1 Motion Actuation - Allowing the visitor to disable motion actuation and use familiar interfaces such as a keyboard or switch reduces the need to remember motions and provides the visitor with known activation methods.
- 1.3.5 Identify Input Purpose - Autocomplete benefits all users. For people with cognitive impairment, not having to remember or copy information has an added benefit that reduces errors and effort. The intrinsic load of completing forms is greatly reduced when

the information is locally stored and retrieved. Visitors do not need to switch applications or refer to handwritten information.

WCAG 2.1 success criteria impact on germane load

Below is the correlation between the success criteria and germane load. Provide well-structured and consistently presented content that allows assistive technology to regroup content.

- 1.1.1 Non-Text Content - When an image provides information, people with cognitive impairment who use screen readers for cognitive support will hear alternative text that aligns with an image. This helps them understand the meaning of the image. Providing cognitive support images or icons with text provides the brain with alternative or supportive information that can enhance the ability to remember [Optum Best Practice; Optum DPL – mobile design].
- 1.3.4 Orientation - Greater reading comprehension loads occur while readers are accessing infrequent words, integrating information from important clauses, and making inferences at the end of sentences. When a reader is on a mobile phone or reading in portrait view, the clauses are more likely to be broken up and distributed across multiple lines. Allowing a user to change to landscape view increases their visual viewport and, in turn, their comprehension (Carpenter, volume 87 number 4 July 1980).
- 2.1.4 Character Key Shortcuts - Shortcuts can be reprogrammed to match shortcuts in other applications. This reduces the need to remember multiple sets of shortcuts.
- 2.2.1 Timing Adjustable - Testing for the user's ability to extend, adjust or turn off a time limit is essential for people with cognitive impairment. Cognitive impairment creates a need for more time to intake, process and store information. The most frequent accommodation for testing given to students with disabilities in higher education is additional time (Belkin, 2018).

Citations

Ali, I. (2018). WCAG: Go beyond and Be Creative! *The 2018 ICT Accessibility Testing Symposium* (pp. 113-121). Arlington, VA: www.ictaccessibilitytesting.org.

Ali, I. (2018). WCAG: Go Beyond and Be Creative! *The 2018 ICT Accessibility Testing Symposium* (pp. 113 - 121). Arlington, VA: www.ictaccessibilitytesting.org.

Belkin, D. (2018, 5 24). *Colleges Bend the Rules for More Students, Give Them Extra Help*. Retrieved 07 24, 2019, from The Wall Street Journal: https://www.wsj.com/articles/colleges-bend-the-rules-for-more-students-give-them-extra-help-1527154200

Butler, E. (2018, 11). *The Readability Formula: Making Your Website Easy-to-Read.* Retrieved 08 19, 2019, from Kick Point: https://kickpoint.ca/the-readability-formula-making-your-website-easy-to-read/

Carpenter, M. A. (volume 87 number 4 July 1980). A Theory of Reading: From Eye Fixations to Comprehension. *Psychological Review*, 329-354.

Cooper, E. L. (2015, 1 15). *Cognitive Accessibility User Research*. Retrieved 07 24, 2019, from W3C: https://www.w3.org/TR/coga-user-research/#research-on-cognitive-function

Headway: the brain injury association. (2019). *Cognitive effects of brain injury.* Retrieved 07 23, 2019, from Headway: https://www.headway.org.uk/about-brain-injury/individuals/effects-of-brain-injury/cognitive-effects/

Marco Zorzi, C. B. (2012, 08 10). *Extra-large letter spacing improves reading in dyslexia.* Retrieved 08 19, 2019, from Proceeding of the National Academy of Sciences of the United States of America: https://www.pnas.org/content/109/28/11455

Mattes, K. (2018, 03 1). *WCAG 2.1: Success Criteria for Cognitive Disabilities*. Retrieved 08 06, 2019, from The Paciello Group: https://developer.paciellogroup.com/blog/2018/03/wcag-2-1-success-criteria-for-cognitive-disabilities/

Moran, K. (2016, 03 20). *Nielsen Norman Group.* Retrieved 08 19, 2019, from How Chunking Helps Content Processing: https://www.nngroup.com/articles/chunking/

Oxford University Press. (2019). *cognition | Definition of cognition in English by Oxford Dictionaries*. Retrieved 05 30, 2019, from Oxford Dictionaries: https://en.oxforddictionaries.com/definition/cognition

Perez, R. (2019, 08 15). Digital Accessibility Engineer. (V. Howle, Interviewer)

Professor Barbara A Wilson, O. (2018, 07 16). *Memory problems after brain injury.* Retrieved 07 23, 2019, from Headway: the brain injury association: https://www.headway.org.uk/media/3996/memory-problems-after-brain-injury-e-booklet.pdf

Pschologist World. (2019). *Cognitive Load Theory*. Retrieved 08 06, 209, from Psychologist World: https://www.psychologistworld.com/memory/cognitive-load-theory

PsychGuides.com. (2019). *Cognitive Problem Symptoms, Causes and Effects.* Retrieved 08 06, 2019, from PsychGuides.com: https://www.psychguides.com/neurological-disorders/cognitive/

Sethfors, H. (2017, 07 04). *Accessibility according to actual people with disabilities.* Retrieved 08 19, 2019, from Axess Lab: https://axesslab.com/accessibility-according-to-pwd/

Shawn Lawton Henry, Editor. (2019, 03 08). *Cognitive Accessibility at W3C.* Retrieved 08 19, 2019, from Web Accessibility Initiative W3C: https://www.w3.org/WAI/cognitive/

Stockman, T. a. (2016, 12 19). *The Influence of Screen-Readers on Web Cognition.* Retrieved 07 29, 2019, from ResearchGate: https://www.researchgate.net/publication/241022137_The_Influence_of_Screen-

Readers_on_Web_Cognition?enrichId=rgreq-96c3d2b55fdc24ed6889f67660615c6c-XXX&enrichSource=Y292ZXJQYWdlOzI0MTAyMjEzNztBUzo0NDEwNDE2NzQzNDY0OTZAMTQ4MjE2MzkzNzQwNA%3D%3D&el=1_x_3&_esc

Sweller, J. (1988). Cognitive Load during provlem solving: Effects of learning. *Cognitive Science 12(2)*, 257-285.

System Concepts. (2018, 01). *Accessibility and cognitive impairment*. Retrieved 07 24, 2019, from System Concepts: https://www.system-concepts.com/insights/accessibility-cognitive-impairment/

Watson, L. (2017, 04 11). *What is an accessible name?* Retrieved 08 19, 2019, from The Paciello Group: https://developer.paciellogroup.com/blog/2017/04/what-is-an-accessible-name/

Wayne e Dick, P. (2017, 2 1). *Operational Overhead Caused by Horizontal Scrolling Text.* Retrieved 08 19, 2019, from Nose to the page: http://nosetothepage.org/Fitz/2dScroll.html

Wild, G. (2018). Testing social media for WCAG2 compliance. *The 2018 ICT Accessibility Testing Symposium* (pp. 61 - 68). Arlington, VA: www.ictaccessibilitytesting.org.

Copyright Notice

Testing Video Players for accessibility

Gian Wild

AccessibilityOz
Melbourne, Victoria, Australia
gian@accessibilityoz.com

Megha Rajopadhye

AccessibilityOz
Melbourne, Victoria, Australia
megha@accessibilityoz.com

Abstract

We conducted a series of tests of 27 major video players, both free and paid, on the current market. This is the fourth year testing has been conducted. We tested on a PC on Chrome and Firefox, and on an iPhone with Safari and an Android phone with Chrome. Six players were excluded from testing because they contained critical accessibility issues. Of the remaining 21 players, OzPlayer rated the highest with 91%, followed by AblePlayer with 79%, PayPal player with 71% and Video.js with 68%. In comparison with last year, several players now support audio descriptions, and keyboard accessibility on mobile devices has improved significantly. It should be noted that the author is involved in the development of OzPlayer.

Video and Accessibility

Web accessibility is about making sure web sites, web applications and mobile apps (including video) are accessible to people with disabilities. Almost everyone understands the need for accessibility features like transcripts, captions and audio descriptions for people with disabilities. However, fewer people realize that it is also important that the video player itself must be accessible. Lack of accessibility in video players can be the consequence of a number of things, but usually includes inadequate keyboard access, inoperable captions and absence of audio descriptions and transcripts.

About the testing

We tested the following video players: AblePlayer, Acorn, BlueBillyWig, Brightcove, BridTV, Elite, Facebook, Flowplayer, jPlayer, JWPlayer, Kaltura, Media Element, MediaSite, Microsoft, The O Player, OzPlayer, Panopto, PayPal player, Plyr, Video for Everybody, Video.js, Vidyard, Vimeo, Vimeo embedded, Wistia, YouTube and YouTube embedded.

We conducted four types of testing: desktop testing on Windows 10 Chrome, desktop testing on Windows 10 Firefox, mobile testing on an iPhone and Safari with a keyboard and mobile testing on an Android phone and Chrome with a keyboard.

- Seventeen accessibility items covering control accessibility, keyboard accessibility, transcripts, captions, audio descriptions and multiple languages were tested on the two desktop browsers and two mobile browsers. For every failure, the player scored one point.

Players Excluded From Testing

Five show-stoppers were identified based on the four non-interference clauses of WCAG2 (W3C, 2016), as well as basic video player requirements. As with testing in previous years, these show-stoppers were deemed so serious that they could necessitate a user having to close the browser entirely. The five show-stoppers were:

- Does not work: video will not play
- Auto-play: a video plays automatically without the ability to pause the video in a WCAG-compliant way (failure of WCAG2 Level A Success Criterion 1.4.2: Audio Control);
- No pause feature: a video once started, cannot be paused or stopped
- Keyboard trap: the keyboard becomes trapped in the video player (failure of WCAG2 Level A Success Criterion 2.1.2 No Keyboard Trap); and
- Full-screen inverse keyboard trap: at full-screen the keyboard is **not** trapped in the player and traverses the page underneath the player (failure of WCAG2 Level A Success Criterion 2.1.2 No Keyboard Trap). Please note that if a player had this issue but the video player could be closed by using the Escape key, then this was **not** deemed a failure.

The players excluded from testing were:

- BlueBillyWig: No pause feature.
- JWPlayer: Keyboard trap on the search feature on Android
- Kaltura: Full-screen inverse keyboard trap – although the video would close on Escape, the audio continued
- Microsoft: No pause feature on Android with the keyboard
- The O Player: Does not work on iOS or Android
- YouTube: Auto-play, where the feature to turn off auto-play is difficult for the keyboard user to access

Overview of Results

The top-rated player was OzPlayer with 91%, followed by AblePlayer with 79%. The next-ranked player was PayPal player at 71%, followed by Video,js at 68%. Plyr, Panopto, YouTube embedded, Vidyard, Acorn and Wistia all scored between 50 and 65%. MediaElement and Brightcove scored between 40 and 50%. Vimeo, Vimeo embedded, Elite, Video for Everybody, Facebook and BridTV scored between 30 and 40%. MediaSite and JPlayer scored 27%, and FlowPlayer was the lowest scoring of the included players at 17%.

In this fourth year of testing, OzPlayer and AblePlayer have remained at the top of the list of accessibility players.

Common Accessibility Errors

Controls

The accessibility of the controls of a video player are very important. We tested to determine if the volume could be changed independent of system volume, whether color alone has been used to convey information and whether controls met color contrast requirements. All video players, with the exception of OzPlayer failed at least one of these requirements.

All video players with the exception of OzPlayer, Wistia, Video for Everybody, Facebook, Video.js and Vidyard used color alone somewhere in the player. In Figure 1, color alone has been used to convey whether the transcript is enabled or not in the Panopto player.

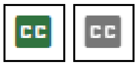

Figure 1 –Panopto Captions controls
(left: captions enabled, right: transcript disabled)

All players with the exception of OzPlayer, Acorn, Video for Everybody and Vidyard failed color contrast requirements within the player. In Figure 2 the color contrast of the slider in the YouTube embedded player does not meet WCAG2 color contrast requirements.

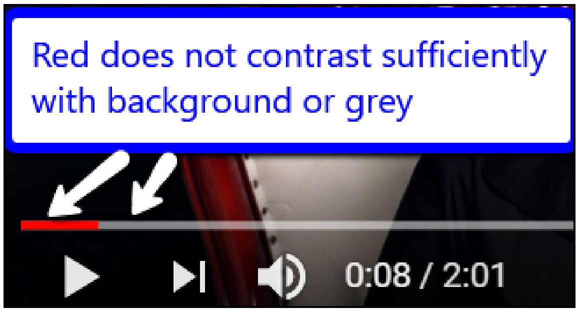

Figure 2 –YouTube embedded player

All video players, with the exception of OzPlayer, Video.js, Plyr, Acorn, Elite and FlowPlayer failed the volume requirement (that a volume control be provided) on Android. In Figure 3, there is no volume control in the Vimeo player on Android.

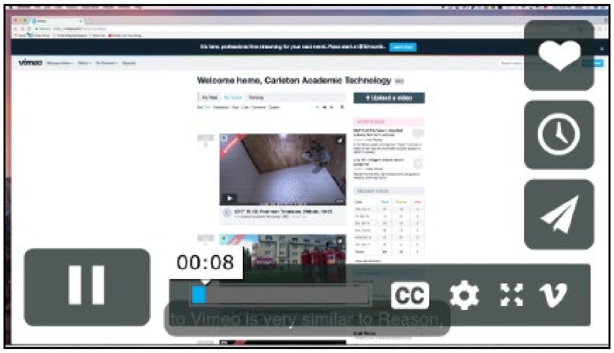

Figure 3 - Vimeo player on Android

Keyboard

Some users are restricted only to the keyboard, or to an assistive technology which mimics the keyboard. We tested whether the video player controls were accessible to the keyboard, whether they received keyboard focus in the correct order, whether there was a keyboard focus indicator and whether that keyboard focus indicator was highly visible.

The Elite player and FlowPlayer could not be used by the keyboard on desktop or any other device. Acorn, Brightcove, jPlayer, MediaSite, Plyr, Video.js, Vimeo, Vimeo embedded, Wistia, YouTube embedded and OzPlayer could not be played by the keyboard on the iPhone.

All video players with the exception of PayPal player had at least one keyboard failure. In many cases, videos could only be paused and played using a keyboard on a mobile.

In Figure 4, the full-screen control in OzPlayer does not work more than once in Firefox.

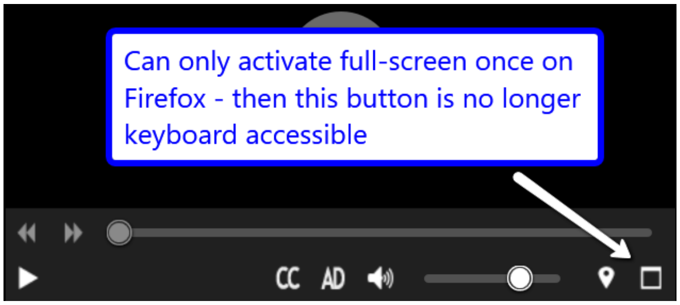

Figure 4 –OzPlayer in FireFox

Most videos utilized the correct keyboard focus order, with the exception of Acorn, Brightcove, MediaSite, Video for Everybody, Video.js and Wistia. In Figure 5 the Fullscreen control in the Acorn player receives focus prior to the Captions control even though the Captions appear first in the visual order.

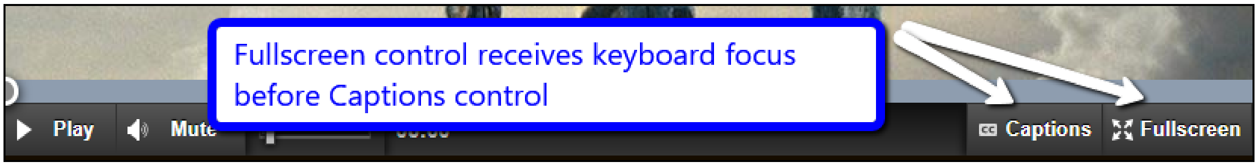

Figure 5 – Acorn video player and keyboard focus indicator

Of the videos that were keyboard accessible, all failed the highly visible keyboard focus indicator with the exception of AblePlayer, Plyr and OzPlayer. In Figure 6 the Chapters, Captions, Audio Descriptions and Picture-in-Picture controls in the Video.js player do not have a keyboard focus indicator.

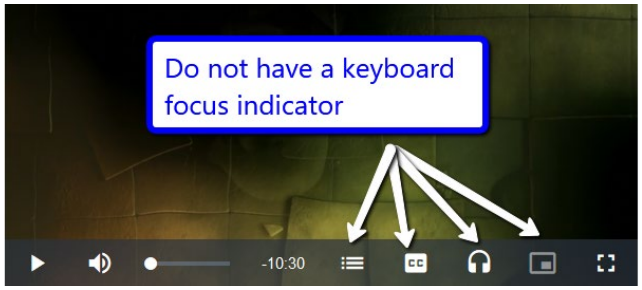

Figure 6 – Video.js and keyboard focus

Transcript

WCAG2 requires that all videos contain captions and either audio descriptions or a transcript at Level A. Incorporating a transcript feature into a player is an important part of making that player accessible. Although a web site owner can meet this requirement by adding the transcript under a video, if it is not disseminated with the player then there is a risk that it will not be available to people with disabilities. This is especially the case as so few video players support audio descriptions. The only players which incorporated a transcript feature were OzPlayer, AblePlayer, Acorn and Panopto. Figure 7 shows the transcript feature in OzPlayer.

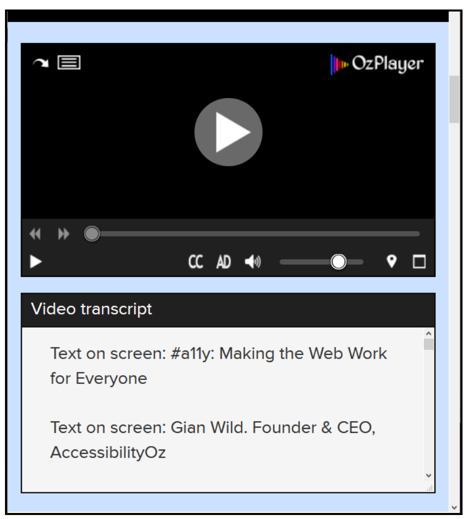

Figure 7 - OzPlayer

Captions

All video players contained caption features that operated on both desktop browsers and mobile with the exception of MediaSite, Video for Everybody, jPlayer, Facebook, FlowPlayer, BridTV and Video.js. In Figure 8, captions are not available on Facebook in Firefox.

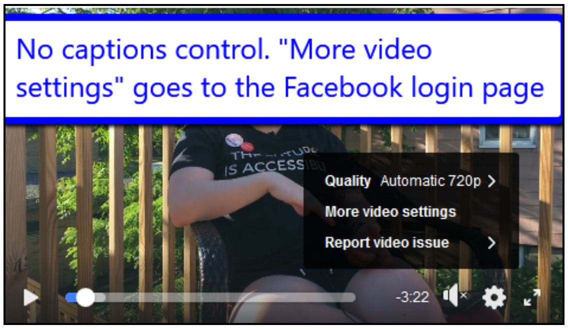

Figure 8 – Facebook in Firefox

Audio Descriptions

Where captions are essential for people who are Deaf or Hard of Hearing, audio descriptions are essential for people with vision impairments. In past years, only OzPlayer and AblePlayer supported audio descriptions. This has changed: this year Brightcove and Video.js also support audio descriptions. In Video.js the audio descriptions can be accessed on an iPhone using the iOS player in the same way that captions are accessed (see Figure 9).

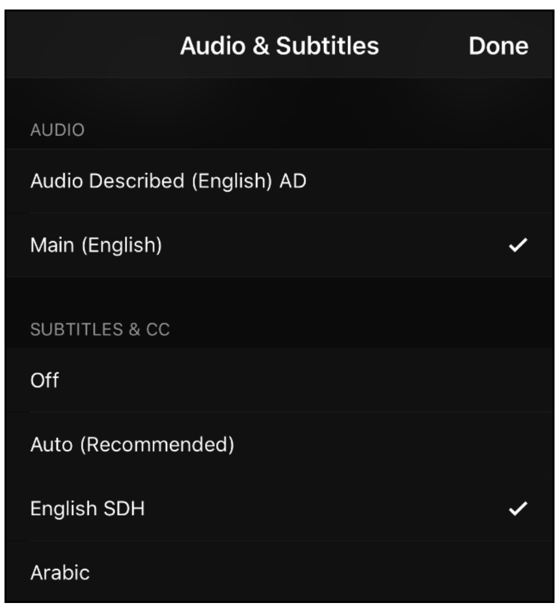

Figure 9 – Options to select captions and / or audio descriptions in the Video.js player on an iPhone

Figure 10 shows the audio descriptions feature in Brightcove.

Figure 10 - Audio descriptions control in Brightcove

Multiple languages

Captions, and especially subtitles, are used by screen reader users. Therefore, it is essential that the LANG element is correctly encoded where there are multiple language text tracks. In Figure 11, the Arabic transcript in the Acorn player is shown. Following this, is the code for this snippet. The code does not have an encoded LANG or AR to indicate the text is Arabic. Therefore, a screen reader will not know how to pronounce this content.

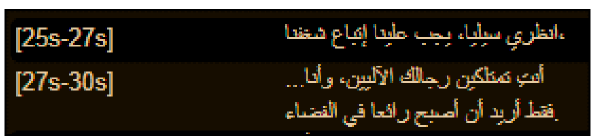

Figure 11 – Arabic transcript in the Acorn media player

```
<span data-begin="25" data-end="27"> انظري سيليا، يجب علينا إتباع شغفنا،</span><span data-begin="27" data-end="30">أنتِ تمتلكين رجالك الآليين، وأنا...<br> فقط أريد أن أصبح رائعا في الفضاء.</span>
```

In-depth Results of Highest Scoring Players

OzPlayer

OzPlayer contained three accessibility errors: on Firefox the fullscreen button is not keyboard accessible, the video cannot be played using the keyboard on an iPhone and audio descriptions are not available on an iPhone.

AblePlayer

AblePlayer continues to use color alone to convey whether a control is operational or not. Color contrast is not sufficient on inactive controls. The transcript contains options to move, resize and close, and none of these options can be activated by the keyboard on desktop, iPhone or Android. On an iPhone at small screen size it is only possible to play and pause the video – although other controls receive keyboard focus, they can't be actioned. On iPhone and Android there is the ability to mute and unmute the video, but there is no way to adjust volume independently of the standard phone volume. AblePlayer supports audio descriptions on all devices.

PayPal player

Color alone is used to indicate the position of the video on the slider and this does not meet color contrast requirements. PayPal player also uses color alone to indicate whether a control is operational or not. On Android the keyboard focus indicator is not highly visible, and there is no way to adjust volume independently of the standard phone volume. PayPal player does not support an associated transcript or audio descriptions.

Video.js

On Firefox the keyboard is not trapped in the video when it is in full-screen mode. However even though the keyboard traverses the page underneath the player, the player can still be closed using the Escape key. In Firefox the Chapters, Captions, Audio descriptions and Picture-in-Picture controls do not have any keyboard focus indicator. Video.js is not accessible via the keyboard at all on an iPhone, and non-visible items receive keyboard focus in full-screen mode on Android. Video.js does support both captions and audio descriptions, but does not support an associated transcript. Neither the captions nor the audio descriptions can be accessed on Android.

Conclusion

This is the fourth year that this testing has been conducted. As in previous years, AblePlayer and OzPlayer were the top two players, however it is heartening that most of the other players did not contain showstoppers, and some major video players have begun to support audio descriptions.

References

W3C. (2016). Conformance Requirements. Retrieved from https://www.w3.org/TR/WCAG20/#conformance-reqs

Copyright Notice

How to measure digital accessibility and legal exposure

Claudio Luis Vera

cvera@simple-theory.com
Fort Lauderdale, FL

Abstract

If you're responsible for the accessibility of a website or application, here's an approach that combines various testing methods and business intelligence (BI) tools to measure your product's usability and quantify your legal risk. There are four universally accepted principles in web accessibility, and the first three — *perceivable*, *operable*, and *understandable* — require different approaches for testing and evaluation.

The hierarchy of accessibility

Figure 1: Hierarchy of accessibility

Perceivable

In a typical website, content is *perceivable* when it's available to all assistive technologies such as screen readers. A site or app would be in compliance with this principle if the abled version were to show roughly the same content as the assistive technology's version. This type of conformance is easily tested through automated testing tools and bots.

Operable

A digital experience is operable when the interactive elements can be operated by users of assistive technologies (AT). On websites and apps, this is most commonly determined through manual testing of keyboard equivalents to pointing devices.

Understandable

Quite often, the AT version of a digital experience doesn't make much sense to a human user, even if it's perceivable and operable. In this case, usability testing by human testers with disabilities is needed to determine if the site or app is usable and *understandable*.

Each of these types of testing returns a different type of data, and aggregating the results poses several unique challenges. Together, the principles and testing methods form a hierarchy of accessibility, as seen in figure 1.

Automated testing for perceivable content

Many major sites use automated testing platforms such as AMP, WorldSpace Attest, and Siteimprove for monitoring accessibility violations on their sites. These tools primarily test for the *perceivability* of content by ensuring that equivalent content exists for assistive technology. At the time of this writing (mid-2019), automated tools are not able to parse the meaning or accuracy of alternate content.

Most of the major testing platforms have browser-based interfaces that allow site owners to see raw test results and perform some rudimentary analysis. While they provide an overall two-digit accessibility score between 0 and 100, each platform vendor uses a different proprietary method to arrive at that score. This makes interoperability and portability of data between platforms difficult, if not impossible for site owners.

An alternative for managing automated testing data

A few vendors provide an API to their platforms, allowing site owners to export data from the platform(s) or create queries through a business intelligence tool. Depending on the breadth of the APIs, this creates an opportunity to perform detailed analytics far beyond the limitations of the platforms' web-based interfaces.

All automated testing tools share a common workflow for gathering automated testing data: a web page is loaded, and roughly 150 accessibility checks are conducted on a snapshot of the page's markup. If the tool is linked to a subscription on a testing platform, then the results are captured in the platform's database. While the data structure usually differs too much for a direct migration from one platform to another, it can be aggregated and merged with a business intelligence (BI) tool.

Patterns in data structure

Each page scan or crawl through a platform should return the following data, which should be exportable as columns in a simple data table through an API:

- URL
- Type of violation
- Number of violations
- Timestamp

Some testing platforms allow a site to be divided into page groups based on URL patterns, giving the tester more granularity in reviewing the results. Additionally, the platform may assign an ID number to each accessibility check and match it against a WCAG success criterion. Others group violations into categories relating to headers, images, forms, and other HTML entities. In an ideal data model, vendors would provide these as additional columns in our data table:

- Page group id
- Page group name
- Check ID
- Success criterion (SC)
- Page views
- Count of pages with formsHowever, different vendors offer varying levels of availability and different structures for their test data. In the best of circumstances, a site owner would be able to work with a complete data dump of their website's accessibility performance, offered as a downloadable table in CSV format. To this author's knowledge, no platform vendor offers this convenience yet to its clients.

API queries offer an alternative, but they're more limited in their scope than data dumps. It can take dozens of queries to assemble a broad picture of a client's website through an API. However, a skilled analyst can use business intelligence tools like Power BI to merge multiple queries into one large fact table for dimensional modeling.

Quantifying legal risk

The past few years have seen an explosion of legal activity around web accessibility. Much of this comes from repeated "copy-and-paste" lawsuits, where the plaintiffs' law firms run scans with off-the-shelf accessibility scanning software and present the results as evidence in their claims. Thus, lawsuits tend to focus on easily-detected violations where the content fails to be *perceivable* to all users.

If we were to reverse-engineer one of these copy-and-paste lawsuits, we'd find clear patterns which can be used to quantify legal exposure. In 2018, the most active plaintiff law firm was Cohen & Mizrahi, LLP, which filed 372 federal website accessibility lawsuits, including *Mendez v. Apple.* Mendez and other lawsuits repeat a common pattern that includes a list of 17 common potential violations, followed by a list of the violations that the law firm actually found. The evidence often consisted of a printout of a scan performed by one of the firm's partners using Sortsite, a standalone site scanning application on their PC. A crawl with this application can catalog violations for up to 22,500 pages and provide a PDF report within a few hours.

Site owners can mirror these techniques by performing broad-based site scans with their automated testing platforms to track the top detectable violations, or "low hanging fruit". Using this approach, the following metrics can be derived easily for the top violations:

Error density = number of errors detected / number of pages scanned
Another of thinking of this metric is as “violations per page”.

% Incidence = pages with errors / number of pages scanned
What is the percentage of pages that have this violation?

Violations served = Error density * pageviews

For the purpose of measuring overall legal risk, the total of violations served is a powerful metric similar to measuring toxic emissions by a polluter. Here’s an example:

40 violations per page x 1,000,000 pageviews per month = 40,000,000 violations served monthly

Using dimensional modeling, these metrics can be treated as aggregate totals, or they can be filtered, sorted, or drilled down by page group, success criterion, violation type, etc. Additionally, timestamps can be used to measure progress and see trends. This approach is especially useful when managing multiple web properties and tracking legal exposure (in violations served) over time, as in the examples below:

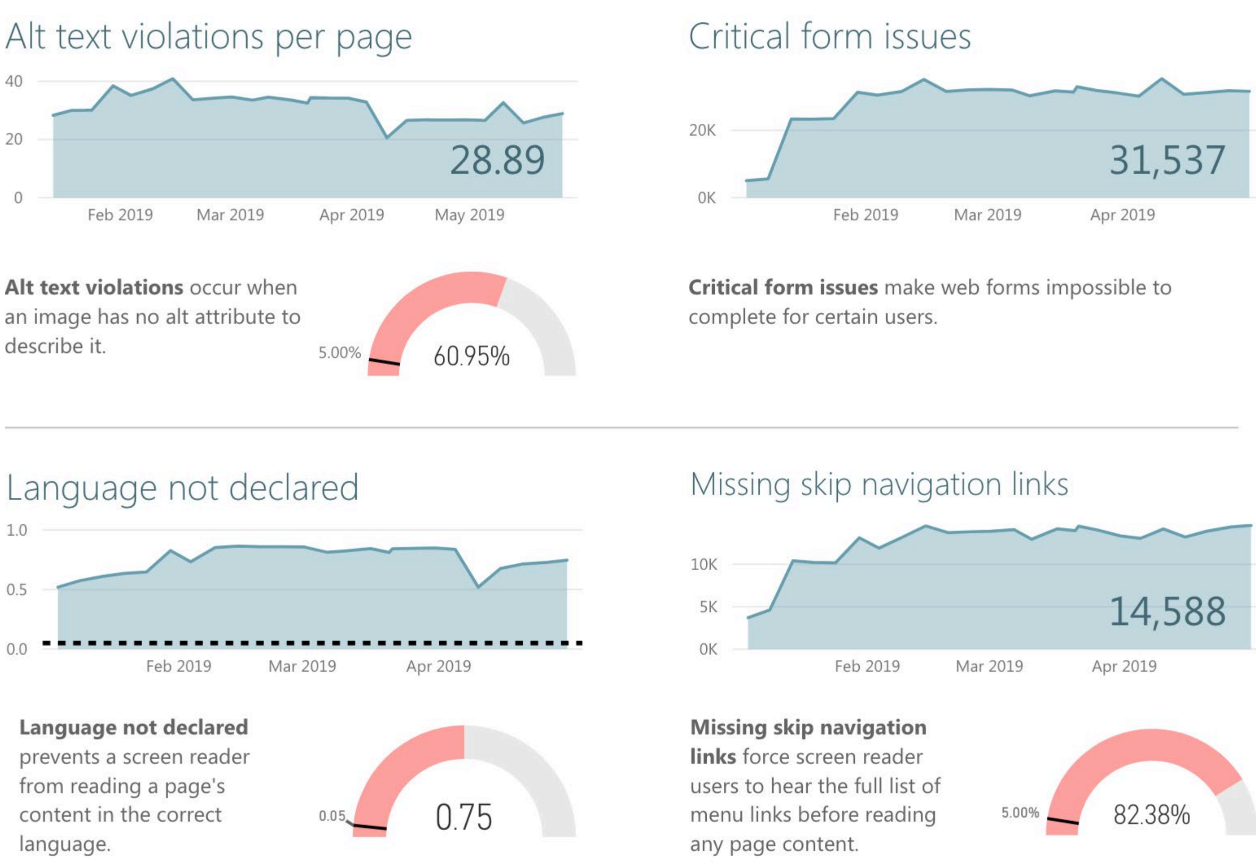

Figure 2: Sample BI report showing trends over time of error density and percent incidence of common violations

A manual testing scorecard

Automated tools are incapable of testing the *operability* of controls on a web site or determining the quality of alternative content. Instead, manual testing by skilled QA personnel with knowledge of accessibility standards is required. With native and mobile apps, automated testing is extremely limited, making manual testing the preferred method.

Unlike bots, humans can't detect every instance of a violation on a web page or app. For this reason, each manual check needs to be scored **pass – fail – n/a**, as opposed to counting violations like a testing tool.

Each check that gets performed is weighted on a 1-4 scale depending on severity:

- **Critical** (4): prevents some users from performing critical tasks with no workaround.
- **High** (3): prevents some users from performing critical tasks, but a workaround may exist for skilled users.
- **Medium** (2): where the user experience is seriously degraded for users of certain assistive technologies
- **Low** (1): the user experience is degraded for users of certain assistive technologies

This scale is commonly used for weighting issues in QA testing, which allows developer teams to give the same importance to critical accessibility issues.

Here's an example of how a scorecard can be set up in Excel using DAX formulas:

Manual test results for

Sample page 66.0%

Keyboard accessibility - general							
no keyboard traps	FAIL	2.1.2	A	4	4	0	0
no reverse keyboard traps	PASS	2.1.1	A	4	4	1	4
Keyboard accessibility - navigation							
main nav can be accessed with keyboard	n/a	2.1.1	A	4	0	0	0
can access child elements with keyboard	n/a	2.1.1	A	4	0	0	0
navigation is operable with keyboard	n/a	2.1.1	A	4	0	0	0

Description	Pass/fail	SC	Level	Severity	Logic 1	Logic 2	Score
No keyboard traps	PASS	2.1.2	A	4	=IF(B2="n/a", 0, E2)	=IF(B2="PASS", 1, 0)	F2*G2

A linear weighted check method can be set up to multiply the severity (1 to 4) by a binary logic number (1 or 0). In this example, a **PASS** rating returns a score of 4, while a **FAIL** or **n/a** rating would return a score of 0. The severity column could also have a logarithmic weighting scale instead, or even a completely subjective one if that's preferred.

The weighted checks are then added up to calculate a percent score for the page or component. This derived by dividing the actual score by the highest potential score (total for Logic 1) for the tests that were performed.

Each page's percent score can then be aggregated into an overall score for the entire section or page group being tested. This scoring method is ideal for audits of less than 15 pages or components.

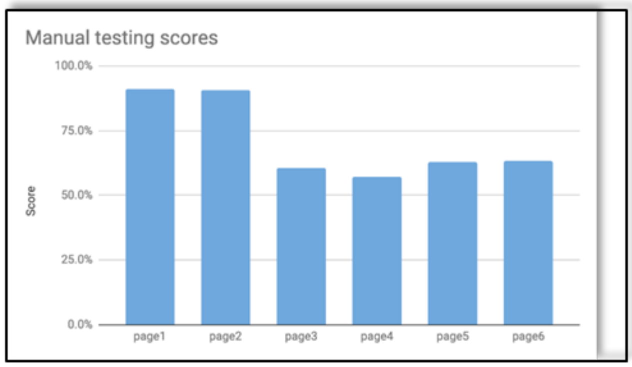

Figure 3 sample bar chart of manual testing scores for an app

If you were to match the pages from your manual test sample to those of a page group with automated scans, then a composite picture emerges of perceivability and operability. For this reason, it's a good practice to create some page groups of manageable size in an automated testing platform, like a Top 10 or a Select 25.

Evaluating usability

Automated and manual tests only prove that a user has access to content and can operate components, which is the bare minimum of legal compliance. Those testing methods alone don't provide insight into the extent that a page or component is understandable or usable by a person with a disability like blindness.

Testing with groups of mixed disabilities

Usability testing with participants of different abilities allows us to determine whether all users can complete critical tasks, like creating an account, product search, making a transaction, updating profile information. The completion rates can be organized into a matrix for each task like this:

Task #1

Disability	Completion rate
Blind screen reader user	62.5%
Blind users with mobility issues	87.5%
Users with mobility issues	56%
Hearing impaired users	100%
Users with cognitive impairments	62.5%

Like the manual testing scorecard, the scores for the individual tasks can then be aggregated into an overall mean score for the entire sample.

Scoring feelings

A more affective score can be determined by applying the System Usability Scale (SUS). With this method, the test participant is asked a series of ten very specific questions about the ease of using a digital product, which are answered using a Likert scale. Here's a sample of the questions that would be asked for a mobile app:

	Strongly disagree	Disagree	Neutral	Agree	Strongly agree
I think that I would like to use this app frequently.					
I found the app unnecessarily complex.					
I thought the app was easy to use.					
I think that I would need the support of a technical person to be able to use this app.					
I found the various functions in this app were well integrated.					
I thought there was too much inconsistency in this app.					
I would imagine that most people would learn to use this app very quickly.					
I found the app very cumbersome to use.					
I felt very confident using the app.					
I needed to learn a lot of things before I could get going with this app.					

The answers are then tabulated in a few steps to yield a score between 0 and 40, then multiplied by 2.5 to for a final score between 0 and 100 for the entire product.

Net Promoter Score

In addition to the SUS questions, it can be useful to ask, “How likely would you be to recommend this product to a friend?” as well. This question is used to derive the Net Promoter Score (NPS), which is the most-favored metric for gauging customer satisfaction.

Conclusion

By combining data from different testing methods, web site owners can gain a broader picture of their sites’ accessibility and evaluate it over time. And by comparing data from multiple testing platforms, major brands can derive a more accurate picture of their compliance efforts. This is comparable to the “spaghetti model” diagrams used to forecast the path of hurricanes based on the convergence of numerous individual forecasts.

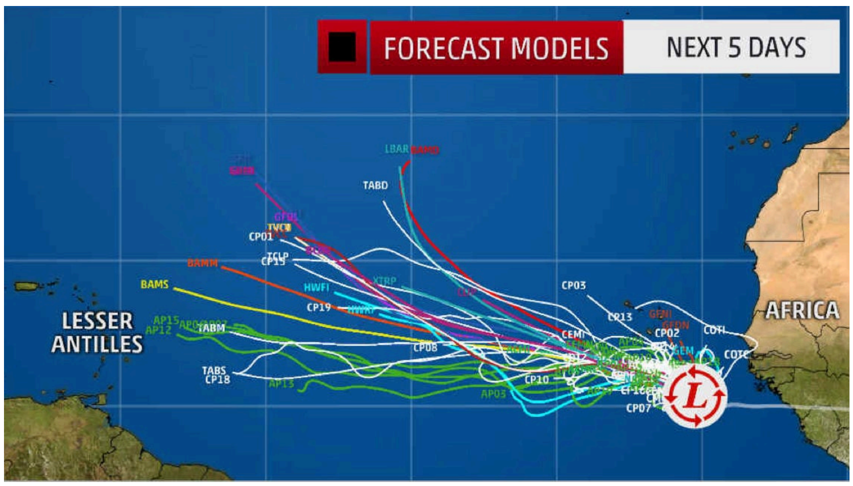

Figure 4: example of a "spaghetti model"

Broad-based automated testing is powerful for tracking performance over time for large web properties, but that is not available for other digital platforms (such as mobile) at this time. Still, many of the quantitative techniques described in this paper are appropriate to testing mobile or native applications, games, augmented reality and other digital experiences.

The combination of BI tools, dimensional modeling, and data visualization can provide far greater insight than that provided by the web interface of an automated testing platform, which benefits both vendors and their clients.

By allowing large data tables to be accessed through APIs or downloaded, testing platform vendors can open the door for improved business analytics. Vendors have less pressure to anticipate the client’s needs and provide appropriate analytics and data visualization through their platforms. It also creates new revenue opportunities through metered API services, custom metrics, advanced BI reporting, and consulting.

Both vendors and clients would also benefit from a common set of data fields for automated testing as described in this paper. Even a small amount of standardization would also provide for portability of test data and usher in a new generation of future in-depth analysis tools.

Copyright Notice

Utilizing "Big" Accessibility Data to Transform the Web

Jared Smith

Associate Director, WebAIM
6807 Old Main Hill, Logan, UT 84322-6807
jared@webaim.org

Abstract

Analysis of wide-scale accessibility data, such as the WebAIM Million evaluation of 1,000,000 home pages, can help transform efforts toward making web content more accessible.

Introduction

In February 2019 WebAIM conducted an accessibility evaluation of the home pages for the top 1,000,000 web sites. While this research focused only on automatically detectable issues, the results paint a rather dismal picture of the current state of web accessibility for individuals with disabilities. Despite the poor state of home page accessibility, these data have helped to shape efforts to better impact web accessibility for good.

A follow-up analysis on the original 1,000,000 home pages was conducted August 2019. This analysis identifies changes in web accessibility of these pages over the course of a six month time period.

This paper presents a high-level overview of the WebAIM Million findings and re-analysis, but more importantly, explores how these data (and future data sources) can be informative and transformational:

- What do the detectable errors tell us about where our efforts should be placed?
- What patterns are present that could allow accessibility remediation efforts to be more effective?
- How do various technologies impact accessibility for good or bad?
- What types of sites or characteristics of sites align with better accessibility?
- Can deep data analysis and machine learning on big accessibility data sets provide additional insight?
- Can these tools transform the longitudinal accessibility data from being descriptive of the current state of accessibility to being prescriptive about optimal future efforts?

The WebAIM Million

The WebAIM Million research project (a summary of initial findings is at https://webaim.org/projects/million/) generated an immense amount of data on the current state of accessibility of web site home pages. The WAVE stand-alone API was used to collect automatically-detectable accessibility issues, with additional tools in place to collect site technology parameters (such as the presence of WordPress, jQuery, React, Google Ads, or any of 1200 other web technologies). The resulting database holds 168,000,000 distinct data points on home page accessibility. This provides the first known wide-scale baseline of web accessibility.

59,653,607 distinct accessibility errors were detected across the 1 million home pages—an average of 59.6 errors per page. 782,481,056 distinct HTML elements were analyzed, meaning there was an average of 782 elements per home page. This results in approximately 7.6% of all home page elements having a detectable accessibility error. Users with disabilities would expect to encounter detectable errors on 1 in every 13 elements with which they engage. 97.8% of home pages had detectable errors that indicated WCAG 2 compliance failures—though because this only includes detectable failures, the actual WCAG conformance rate is certainly very low—likely well below 1%.

Despite these data painting a poor picture of current home page accessibility (which is known to correspond with overall web site accessibility), the accessibility field should not be discouraged, but must instead utilize these data to better effect change. For example, while the number of accessibility errors on home pages was significant, most errors fell into 6 primary categories:

1. Low contrast text
2. Missing alternative text for images
3. Empty links
4. Missing form input labels
5. Missing document language
6. Empty buttons

Simply focusing efforts on these error types would have a significant positive impact on users with disabilities. Fully addressing these issues on the 1 million home pages that were tested would result in a decrease of well over 50 potential end-user barriers on the average home page.

Of primary interest in the data is that the 60.1% of home pages that had ARIA present averaged 26.7 more detectable errors than pages without ARIA. While pages that need ARIA are inherently more complex, the fact that ARIA is in place would typically suggest some attention to accessibility, when in fact, these pages have significantly more errors. This suggests the need for better tools and frameworks and increased education about proper ARIA implementation.

Web Technologies and Accessibility Data

Associating errors, error types, and error patterns with the technologies present on home pages provides additional insight. There are strong correlations between certain technologies and increases or decreases in errors. When the following common web technologies were present on home pages, the pages exhibited fewer accessibility errors than pages that did not have this technology present (the % difference in errors is provided):

- Joomla (-17.2%)
- Drupal (-3.1%)
- MooTools (-12.6%)
- YUI (-12%)
- ASP.net (-12.6%)

Unfortunately, most technologies corresponded with an increase in detected accessibility errors. The following common web technologies exhibited more errors than pages without this technology present (the % increase in errors is provided):

- WordPress (.6%)
- Blogger (237%)
- React (10.1%)
- Angular (19.2%)
- jQueryUI (25.3%)
- jQuery (43.3%)
- Bootstrap (10.3%)
- Google AdSense (87.8%)

These data can help drive technology changes that would undoubtedly be very impactful throughout the web. As one example, jQuery, which was present on 77.4% of the analyzed home pages correlates with a 43% increase in detectable errors. While it cannot be asserted that jQuery itself has introduced these errors, the correlation is very strong. Focusing accessibility efforts on jQuery and jQuery users could therefore result in notable improvements across a wide swath of the web.

The WebAIM Million database certainly holds many additional answers to which the questions have not yet been asked. More comprehensive exploration of these data may provide additional guidance and insights. WebAIM invites data analysts and researchers to explore these data.

WebAIM Million Re-analysis

The million home pages were re-analyzed in August 2019, six months after the original analysis. These updated data provide informative longitudinal data that could perhaps be used as an indicator of future web accessibility progress.

While 6 months is a relatively short time frame, changes in accessibility errors and features, and in technology usage, were detected. There was an average of 59.1 errors per home page, a slight decrease from 59.7 errors per home page in February. While the number of errors decreased slightly, the WCAG 2 failure rate based on automatically detectable errors increased slightly to 98.0% in August compared to 97.8% in February.

Other data indicated that HTML5 usage increased from 74.1% to 77%, image alternative text and form labeling errors decreased slightly (though overall accessibility is very poor), and implementation of HTML regions and ARIA landmarks increased.

Perhaps most notable was the increase in pages that utilize ARIA – from 60.1% in February to 64.5% in August. The average number of ARIA attributes per home page with ARIA increased sharply (10.9%) from 37 to 41 in that same period. This aligns with a 4.3% increase in the number of page elements – from 783 to 816 elements per home page on average. Home pages are getting much more complex with notably increased ARIA, yet the overall accessibility error rate of home pages is changing but very little.

When comparing accessibility data for specific web technologies, minor changes were detected for certain technologies – most popular frameworks and libraries exhibited a decrease in detectable page errors compared to 6 months prior.

The WebAIM Million data and reanalysis data provide valuable longitudinal data on the state of web accessibility. At the time of this publication (August 2019), only initial analysis of the re-analysis data had been conducted - deeper analysis of these data, and future benchmarks, especially using big data analytics and machine learning methodologies, could provide even more useful insights into the potential long-term trajectory of web accessibility.

Utilizing Future Data

An exciting prospect from the collection and analysis of "big" web accessibility data is the ability to detect trends, patterns, and correlations. Knowing how, where, and (perhaps) why accessibility errors and improvements are occurring can allow targeted interventions, education, and tools. Data analytics and machine learning methodologies could also allow us to better predict how to effect positive change – and in a fully automated manner. This could allow targeted and very customized efforts in the areas that will most beneficially impact the disabled.

Adding additional data, such as additional site parameters (sector, technology versions, content age, etc.), machine learning models that define optimal accessibility, data on impacts on end user of certain error types and patterns, correspondences of automatically-detectable errors to manually-verified errors, etc., could allow any number of significant results and findings. For example, data showing correlations of certain accessibility error types in a particular sector when a particular Content Management System is in place could allow very targeted education, bug fixing activities, and awareness activities. And more so, patterns of success can be detected – if a certain code pattern in a specific world region corresponded with increased accessibility, this could be studied and replicated elsewhere.

In the future, machine learning could be implemented to build models of accessibility (and inaccessibility). This could allow better detection and prevention of accessibility issues via automated processes. It could also provide better end user tools to support users encountering accessibility barriers – ML processes could perhaps transform inaccessible content or direct users to suitable alternatives.

There are innumerable possibilities in which these data could be utilized, once collected and properly analyzed. The nexus of such data will be the ability to transform it from being descriptive of the current and past state of accessibility, to being prescriptive about the future and, more importantly, about best practices for making the web more accessible. The WebAIM Million provides an initial dataset for analysis and benchmarking. It is believed that an increased focus on "big" accessibility data could be used to significantly transform the web for good.

Copyright Notice

Symposium Committee

Event Committee

Chris Law (Symposium Chair)
Organizational Accessibility Consultant, Accessibility Track

Matt Feldman (Symposium Co-Chair)
Director of Customer Success, The Paciello Group, A Vispero Company

Laura Renfro (Workshop & Courses Chair)
Senior Accessibility Lead, Renfro Consulting

Janet Sedgley (Marketing & Outreach Chair)
Technical accessibility coordinator, University of Montana

Erin Kirchner-Lucas (Competition Chair)
Accessibility Coordinator, RedShelf

Andrew Nielson (Awards & Presentation Chair)
Senior Program Manager, New Editions Consulting, Inc.

Kathryn Weber-Hottleman (Communications Chair & Webmaster)
IT Accessibility Coordinator, University of Connecticut

Christine K. Loew (Content Chair)
Director, Accessibility for Digital Assessments, The College Board

Peer Review Committee

Shadi Abou-Zahra
Strategy and Technology Specialist, W3C Web, Accessibility Initiative (WAI)

Jennison Asuncion
Engineering Manager (Accessibility), LinkedIn

Jon Avila
Chief Accessibility Officer, Level Access

Jennifer Chadwick
Lead Accessibility Strategist (North America), Siteimprove

Anne Marie Davis
Senior Accessibility Technician, New Editions Consulting, Inc.

Katherine Eng
ICT Accessibility Specialist, US Access Board

Karl Groves
Founder and President, Tenon.io

Jon Gunderson
Coordinator, Accessible Information Technology Group, University of Illinois at Urbana/Champaign

Sunish Gupta
CEO & Founder, Easy Alliance

Katie Haritos-Shea
Vice President, Accessibility, EVERFI

Allen Hoffman
Deque Systems, Inc.

Trudy Knudsen
Accessibility Consultant

Mark Lapole
Accessibility UX Lead (G Suite), Google

Peter McNally
Senior Consultant, User Experience Center, Bentley University

Eduardo Meza-Etienne
Director of Compliance, eSSENTIAL Accessibility

Ryan Pugh
Director of Accessibility, Onix Networking

John Rempel
Quality Control & Training Specialist, Centers for Inclusive Design Innovation (CIDI) at Georgia Tech

Norman Robinson
IT Change Control Board Change Manager, Department of State

Madeleine Rothberg
Senior Subject Matter Expert, WGBH National Center for Accessible Media

Cyndi Rowland
Director, WebAIM & NCDAE at Utah State University

Anna Vanyan
Systems Analyst, Westat

Kathryn Weber-Hottleman
IT Accessibility Coordinator, University of Connecticut

Dominique Wheeler
Technical Quality Assurance Specialist, Paltech Planning and Learning Technologies

Gian Wild
CEO, Founder and President, AccessibilityOz

Author Index