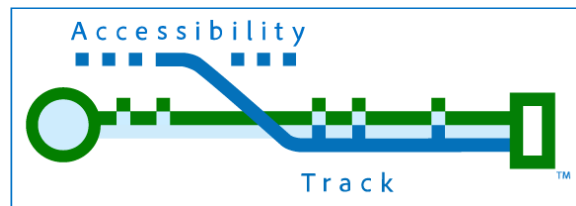


Proceedings of the 2020 ICT Accessibility Testing Symposium: Time for Testing in Testing Times (Remote Work, Commerce, Education, Support...)

Online

October 21 – 23, 2020

Organized by Accessibility Track Consulting, LLC



www.ictaccessibilitytesting.org

Contents

Introduction from the Chairs	1
Keynote: “The Agilization of A11Y: Why Continuous Change is Good!”	3
Seminar: Mobile Site and Native App Testing	5
Workshop: Understanding ARIA 1.2 and the ARIA Authoring Practices Guide	19
Panel: Accessibility Overlays.....	25
Panel: COVID-19: The Great Accelerator (Part 1).....	29
Panel: COVID-19: The Great Accelerator (Part 2).....	33
How to evaluate video conferencing tools for accessibility: Navigating new features and fixes every week	37
Broadening the definition of ‘interaction’ for accessibility testing	43
Introducing the Section 508 ICT Testing Baseline Alignment Framework.....	55
Leveraging Trusted Tester for Web with Test Automation	65
Demodocus: Automated Web Accessibility Evaluations.....	81
Approaches to Remote Testing Using People with Disabilities to Achieve Inclusion ...	91
Developing and Testing Accessible eLearning Courses	101
Focus first: a new front-end approach	115
The Importance of Switch Testing.....	123
Using Personas for Accessible Design.....	129
Social Media Accessibility Testing.....	137
Symposium Committee	145
Author Index.....	149

Introduction from the Chairs



Dr. Chris M. Law, Chair



Matt Feldman, Co-Chair

It goes without saying that these are testing times for everyone as society collectively deals with the Covid-19 global pandemic. Workers, retailers, restauranters, teachers, students, and customer support operators are finding themselves obliged to move to online as the primary means of collaboration with colleagues and customers. It also goes without saying that a considerable number of these systems featured no or poor accessibility development/testing.

Given this rapid transition of many who are new to such online tools, the amount of additional accessibility blockers is growing by the day. Do developers and administrators of new websites and other digital content have accessibility in mind? Most likely not. Do the tools they use to quickly implement new websites, Apps, and electronic documents automatically generate accessible code? Commonly, no.

As a profession, we have had to cope with our own business issues in the same ways as those in other industries. We've also had to step up to support those in other industries to help solve their accessibility problems, many of which have only emerged as a result of the massive switch to online operations. A lot has happened in a short time, and there is much that can be gained by sharing our positive (and perhaps, negative?) experiences in ICT Accessibility Testing research and practice. To play our part, we have to be here to provide support to those tackling their technology issues, so that we may help them ensure equitable access for all members of our society.

And, to play our part for our own profession, we are offering sponsored registrations with our Covid-19 Hardship Fund. If you or a colleague you know has lost their job or had a substantial reduction in income as a result of Covid-19, we want to help.

Our annual symposium has traditionally been a place for in-person knowledge sharing and for professional face-to-face networking. We're using this year's online symposium to continue sharing what works during the current situation and beyond, and we're continuing to network

Introduction

with our peers—but this time online—the same way that so many of us are getting used to while we go through this together.

New for this year, we are introducing a Product Demo Day. Those returning attendees will know that we have not traditionally had product exhibits, demos or promotions at our event. We recognize that this year is very different. With a lack of face-to-face conference events, it has been harder for testing companies to network and engage with audience. With the gathering in one (virtual) place, we have a large number accessibility testers and accessibility program managers together. We are offering this sponsored Demo Day opportunity in the pre-conference week to help our colleagues' network in these unusual times.

The event committee has been working hard this year to bring you a fairly regular program. We have kept the same level of quality with peer reviewed paper submissions and added some very interesting panels too. We're looking forward to connecting with you again this October.

Sincerely,
Chris M. Law, Chair, & Matt Feldman, Co-Chair, 2020 Symposium Committee

Keynote: “The Agilization of A11Y: Why Continuous Change is Good!”



Mike Paciello

We live in unparalleled times in which societal norms have seemingly polarized how we live, work, and interact. Advances in accessibility and usability of technology for people with disabilities and older adults is cyclical, if not frozen in time. We continue to fight for the same goals, herald the same message and recommend the same solutions. The purpose of this presentation is to provoke new thinking, to incite visionary innovation and to suggest that the paradigm shift we all desire requires the agilization of accessibility.

Mike Paciello has been a pioneer and influential figure in the accessibility industry for more than three decades. He wrote the first book on web accessibility and usability (Web Accessibility for People with Disabilities) and has since achieved many notable milestones. He is the founder of WebABLE.Com and co-founder of WebABLE.TV. Mike served as co-chair of the United States Federal Access Board’s Telecommunications and Electronic and Information Technology Advisory Committee (TEITAC), co-founder of the International Committee for Accessible Document Design (ICADD), and was recognized by President Bill Clinton for his contribution to the W3C Web Accessibility Initiative (WAI). He was the recipient of the 2016 Knowbility Lifetime Achievement award. Mike is currently retired but remains active as a mentor and contributor to the accessibility industry.

Seminar: Mobile Site and Native App Testing

Gian Wild

AccessibilityOz
Melbourne, Victoria, Australia
gian@accessibilityoz.com

Keywords: Accessibility; Mobile Site; Native App; Testing Methodology; Responsive Web Site.

Track/Theme: Mobile Web Site and Native App Accessibility.

Abstract

As Co-Chair for the 2018 and 2019 Mobile Committees, Gian Wild will cover the finalized testing guidelines. Learn what you can test on your laptop and what needs to be tested on a device. The testing topics will include handling traps, keyboard use, standard user interface controls, JavaScript events, touch targets and interactive space, links, images, forms, and navigation aids. Participants will receive a copy of the test process. This seminar covers the accessibility of mobile web sites and native app accessibility.

A new methodology for testing mobile sites and native apps

The ICT Accessibility Testing Symposium has developed a methodology for evaluating the accessibility of mobile web sites and native apps. This document is an amalgamation of accepted mobile site and native app accessibility testing standards from around the world, including additional developments from the ICT Accessibility Testing Symposium's Mobile Sub-Committee (for more information, see Acknowledgements).

WCAG2 success criteria are applicable to mobile sites and native apps, however, not all aspects of mobile accessibility are specifically covered by WCAG2. It is the opinion of this committee that merely conforming WCAG2 (or WCAG 2.1) does not provide for a fully accessible experience for users with disabilities.

Please note that this methodology does not include those issues already included in WCAG2, however does include issues identified in WCAG2.1. This guide was written with the intent to clarify the unique needs of users with disabilities who use mobile web sites and native apps and to raise the bar for the web development community. This is a work-in-progress, and, as such, we do not make a claim that conforming to these requirements will ensure that your mobile site is fully accessible to all users.

A note on WCAG2.1

The Committee decided that it was important to also include issues added in WCAG2.1, so that testers who were testing against WCAG2 would also benefit from the mobile-related errors published in WCAG2.1. This methodology differs from WCAG2 and WCAG2.1 in some areas and these are detailed in the Test Cases documents.

Mobile sites versus native apps

There is a great difference between mobile sites and native apps – native apps utilize a completely different codebase. Therefore, the ICT Accessibility Testing Symposium has decided to separate the native app methodology from the mobile site testing methodology.

A note on hybrid native apps

Hybrid mobile apps consist of both HTML and native code. The HTML may be included as a specific page within an application, or within a specific container. When testing a hybrid app, you will need to use this methodology and the Mobile Site Accessibility Testing Methodology.

The detailed methodology

All these documents can be found on the [Mobile Testing Methodology page](https://www.accessibilityoz.com/resources/mobile-testing/) on the AccessibilityOz web site (<https://www.accessibilityoz.com/resources/mobile-testing/>).

Mobile Site Testing Methodology

There are two overview documents:

- [Mobile Site Accessibility Testing Methodology \(Word, 5.48 MB\)](https://www.dropbox.com/s/tz8px2paf4hkkze/1%20-%20Mobile%20Site%20Methodology.docx?dl=0)
(<https://www.dropbox.com/s/tz8px2paf4hkkze/1%20-%20Mobile%20Site%20Methodology.docx?dl=0>)
- [About Mobile Site Testing – Devices, assistive technologies, site types, variations of a page and capturing errors \(Word, 23.4 MB\)](https://www.dropbox.com/s/144h01jivdyvws/2%20-%20About%20Mobile%20Site%20Testing.docx?dl=0)
(<https://www.dropbox.com/s/144h01jivdyvws/2%20-%20About%20Mobile%20Site%20Testing.docx?dl=0>)

There are three sets of test cases documents, which detail how to test a particular requirement in the methodology, why it is important and example passes of the requirement:

- [Mobile Site Accessibility Testing Methodology – Critical Test Cases \(Word, 22.3 MB\)](https://www.dropbox.com/s/xjuzkrzjmtsc2pd/3%20-%20Mobile%20Site%20Critical%20Test%20Cases.docx?dl=0)
(<https://www.dropbox.com/s/xjuzkrzjmtsc2pd/3%20-%20Mobile%20Site%20Critical%20Test%20Cases.docx?dl=0>)
- [Mobile Site Accessibility Testing Methodology – Test Cases \(Word, 90.77 MB\)](https://www.dropbox.com/s/gc3yzurwptiq/4%20-%20Mobile%20Site%20Test%20Cases.docx?dl=0)
(<https://www.dropbox.com/s/gc3yzurwptiq/4%20-%20Mobile%20Site%20Test%20Cases.docx?dl=0>)

- [Mobile Site Accessibility Testing Methodology – Test cases for assistive technologies and mobile features \(Word, 48.14 MB\)](https://www.dropbox.com/s/sj1eu45dlhpwjy7/5%20-%20Mobile%20Site%20Assistive%20Technology%20Test%20Cases.docx?dl=0)
(<https://www.dropbox.com/s/sj1eu45dlhpwjy7/5%20-%20Mobile%20Site%20Assistive%20Technology%20Test%20Cases.docx?dl=0>)

Native App Testing Methodology

There are two overview documents:

- [Native App Accessibility Testing Methodology \(Word, 2.05 MB\)](https://www.dropbox.com/s/z9cnrcua3amapzc/1%20-%20Native%20App%20Methodology.docx?dl=0)
(<https://www.dropbox.com/s/z9cnrcua3amapzc/1%20-%20Native%20App%20Methodology.docx?dl=0>)
- [About Native App Testing – Devices, assistive technologies and capturing errors \(Word, 2.06 MB\)](https://www.dropbox.com/s/ic8nj4sfg13f9ty/2%20-%20About%20Native%20App%20Testing.docx?dl=0) (<https://www.dropbox.com/s/ic8nj4sfg13f9ty/2%20-%20About%20Native%20App%20Testing.docx?dl=0>)

There are three sets of test cases documents, which detail how to test a particular requirement in the methodology, why it is important and example passes of the requirement:

- [Native App Accessibility Testing Methodology – Critical Test Cases \(Word, 37.13 MB\)](https://www.dropbox.com/s/438yg7pqb2a5tbw/3%20-%20Native%20App%20Critical%20Test%20Cases.docx?dl=0)
(<https://www.dropbox.com/s/438yg7pqb2a5tbw/3%20-%20Native%20App%20Critical%20Test%20Cases.docx?dl=0>)
- [Native App Accessibility Testing Methodology – Test Cases \(Word, 101.94 MB\)](https://www.dropbox.com/s/wi9uij0i0tjc8r0/4%20-%20Native%20App%20Test%20Cases.docx?dl=0)
(<https://www.dropbox.com/s/wi9uij0i0tjc8r0/4%20-%20Native%20App%20Test%20Cases.docx?dl=0>)
- [Native App Accessibility Testing Methodology – Test cases for assistive technologies and mobile features \(Word, 77.9 MB\)](https://www.dropbox.com/s/oxwk4c6pbdmugce/5%20-%20Native%20App%20Assistive%20Technology%20Test%20Cases.docx?dl=0)
(<https://www.dropbox.com/s/oxwk4c6pbdmugce/5%20-%20Native%20App%20Assistive%20Technology%20Test%20Cases.docx?dl=0>)

Overview of the methodologies

Please note that this methodology does not include those errors already included in WCAG2. In order to ensure your mobile site or native app is fully accessible, you need to meet WCAG2 and this mobile site and native app testing methodology.

Items that refer only to the Mobile Site are preceded with [MS].

Items that refer only to the Native App are preceded with [NA].

The following steps should be followed to ensure that a mobile site or native app is accessible to all users:

- Step 1: Identify what needs to be tested
- [MS] Step 2: Identify site type and variations [END MS]

- [NA] Step 2: Define application functionality [END NA]
- Step 3: Test critical issues
- Step 4: Test mobile-specific issues
- Step 5: Test mobile assistive technology and feature support

Step 1: Identify devices

What needs to be tested is dependent on the native app, [MS] site type, variations of the page [END MS] and chosen devices.

Identify devices and browsers to test on

Recommended devices and browser combinations:

- iPhone, [MS] Safari [END MS]
- iPad, [MS] Safari [END MS]
- Android phone, [MS] Chrome [END MS]

Other devices to consider

- Android tablet (for example, Samsung Tab A or ChromeBook) [MS] Chrome [END MS]
- Alternative devices such as a Kindle device

Recommendations

Test on the latest version of iOS.

Test on last two versions of Android.

[MS] Where a site is directly aimed at people with particular kind of disability it is worth considering including assistive devices and/or other assistive technologies used by potential users. Including these devices will help to ensure that a tested page is not only accessible, but also user-friendly. A good example of such a site is [Bookshare](#) – an online library for people with print disabilities. Among the users of this service you can find users of braille notetakers and other mobile devices and their user experiences are likely to be different from the experience of standard screen reader users. [END MS]

[MS] Step 2: Identify the site type and variations of the page

- **Desktop web sites:** non-responsive sites that have only one site view or display, whether viewed on desktop or mobile or tablet device – a viewer would see the exact same thing;
- **Responsive web sites:** user views or experiences that change depending on the screen size or other feature as determined in break points by the developer – usually the mobile is simpler than the tablet than the desktop web site;

- **m.dot sites:** that have a particular display for mobile and tablet sites. The m.dot site must also be tested against the entirety of WCAG2, **in addition** to the standard www version of the site.

The testing methods will be dependent on what site type is being tested, and, in the case of responsive sites, how variations in content are controlled.

Desktop web site testing

The desktop web site must be tested against the entirety of WCAG2 on the mobile.

Recommended testing:

- WCAG2 testing of the desktop site on the desktop
- WCAG2 testing of the desktop site on mobile
- This mobile methodology of the desktop site on the mobile and desktop

m.dot site testing

The m.dot site must also be tested against the entirety of WCAG2, **in addition** to the standard www version of the site.

Recommended testing:

- WCAG2 testing of the www site on the desktop
- WCAG2 testing of the www site on mobile
- This mobile methodology on the www site on mobile and desktop
- WCAG2 testing of the m.dot site on the desktop
- WCAG2 testing of the m.dot site on mobile
- This mobile methodology on the m.dot site on mobile and desktop

Responsive web site testing

It is important that **each variation of the page is tested** and that **all functionality is available on all variations of the page**. The testing methods for responsive web site testing are dependent on whether there are variations of the page. All versions of the site on the various devices will be required for comprehensive testing. The user experience and accessibility can vary greatly between these differentiators, never apply results from one situation to another.

Determine which of the following triggers the variation of the page:

- The **device** (e.g. iPhone, desktop, Android, etc.);
- The **operating system** (e.g. Windows, iOS, OS, etc.);
- The **browser** (e.g. Safari, IE 11, Chrome, etc.); and
- The **screen size** (e.g. 280 by 720, 1920 by 1080, 320 by 480, etc.).

Recommended testing:

- WCAG2 testing of each variation on the desktop
- WCAG2 testing of each variation that can be accessed on mobile
- This mobile methodology of each variation on mobile and desktop [END MS]

[NA] Step 2: Define application functionality

Through your understanding of the purpose of the native mobile application, define which functionality is critical to its purpose and use, and that must be tested for efficacy, operability, and workflow from a user experience perspective.

Common elements to test

All functionality should be accessible within the native application; however, it is important to define and include the critical functionality for each individual app to be prioritized in your testing. [END NA]

A good question to ask: how would the experience be impacted if the functionality failed, the content could not be reached, and or the experience caused a barrier to the user?

- Navigation – Menus, header, footer
- Landing screen(s)
- Emergency sections and content
- Login flows
- Settings
- Account and profile
- Contact Us
- Real-time updates (eBay, Uber)
- Privacy policy, Terms and Conditions
- Interactional functionality (adding items to a shopping cart, payment details, live chat, selections for a product in a catalogue, scanning a barcode, VR, QR code)
- Help section
- Widgets (calendars, date pickers)
- Third-party integrations (geo-locational maps)
- And/or High-traffic areas

Step 3: Test critical issues

Exit traps – Applies to all users

Ensure there is always an accessible actionable item (eg. a close button that meets color contrast requirements and has an accessible name) that closes any feature or page that overlays the current page (such as a full-page ad) or an important mobile feature.

Swipe / Scroll traps – Applies to touch users

Ensure you do not override standard mobile touch functions (swiping, scrolling etc) on the majority of the page unless the entire screen is a map.

[NA] Text-to-speech traps – Applies to screen reader users

If the site has an ability to provide content via text-to-speech, the screen reader user must be able to pause or stop the app speaking in a simple manner, e.g. by performing a swipe on a screen.
[END NA]

[NA] Headset traps – Applies to Headset users

Headset users must always be able to pause media (audio or video) content by using the Pause/Play control on the headset. [END NA]

Layer trap – Applies to all users (but mostly encountered by screen reader users)

The user should not be trapped on a non-visible layer.

Step 4: Test mobile-specific issues

Alternatives

Functionality that can be operated by device motion or user motion, interaction and/or gesture can also be operated by user interface components, and responding to the motion, interaction and/or gesture can be disabled to prevent accidental actuation, except for certain situations (for more information see [SC 2.5.4: Motion Actuation](#)).

Any touch gesture must have an alternate gesture, such as a link (for more information see [SC 2.5.1: Pointer Gestures](#) and [SC 1.4.13: Content on Hover or Focus](#)).

Alternatives are provided for geolocation functionality that is mandatory (for example, requiring a specific geolocation before functionality appears), unless the geolocation is essential for legal reasons, or doing so would invalidate the activity). This applies to geolocation via GPS, user statement, IP address or other methods.

Changes of state of non-standard controls (e.g. hamburger menu, star ratings) are clearly indicated.

Audio cues have an equivalent, accessible, visual cue.

Status messages are available to all users without receiving focus (for more information see [SC 4.1.3: Status Messages](#)).

All abbreviations are expanded the first time they are used on the page or a glossary of abbreviations and their expansions is provided (for more information see [SC 3.1.4: Abbreviations](#)).

Where the text requires reading ability more advanced Flesch Kincaid level 8, a summary or description of content is provided (for more information see [SC 3.1.5: Reading Level](#)).

Controls, primary headings, links, field labels and page titles are not ambiguous when read aloud.

Display

Web pages do not contain more than three flashes in a one second period (for more information see [SC 2.3.2: Three Flashes](#)).

Changes of context must always be user-initiated unless it is time-sensitive or an emergency (for more information see [SC 3.2.5: Change on Request](#)).

Size of touch targets is at least 44 by 44 CSS pixels (approximately 7 to 10 millimeters) (for more information see [SC 2.5.5: Target Size](#)).

Touch targets have sufficient inactive space between them (inactive space of at least 22 by 22 CSS pixels -or 4 to 5 mm - should be provided around active elements).

Do not use fixed size containers for blocks of text, unless the display is essential.

Justified text has not been used.

Text and actionable items (including text and non-actionable items that will become actionable), or items that convey meaning should have a minimum color contrast ratio of 4.5:1 when compared with the surrounding background.

The site or native app can be used in portrait mode (for more information see [SC 1.3.4: Orientation](#)).

The site or native app can be used in landscape mode (for more information see [SC 1.3.4: Orientation](#)).

The site or native app does not swap orientation unexpectedly.

Animation triggered by interaction can be disabled (for more information see [SC 2.3.3: Animation from Interactions](#)).

[MS] Pinch zoom is operable, unless an accessible font resizing feature has been included in the web site that allows the user to increase the size of content at least two times the size of the standard font size. [END MS]

[MS] Horizontal scrolling is not required at all when the page is resized. [END MS]

Actionable items

When additional content appears on hover, focus or input it is dismissable, hoverable and persistent (for more information see [SC 1.4.13: Content on Hover or Focus](#)).

Native UI controls, objects, alerts and elements have been used.

Where text links are used the text is visually descriptive (for more information see [SC 2.4.9: Link Purpose \(Link Only\)](#)).

When direct input via the keyboard is not required provide options for the user to achieve the same result (i.e. use dropdown, radio buttons & checkboxes, etc.).

Infinite scrolling has not been used.

Color alone should not be used to indicate actionable items (if not underlined) within inline text. A secondary method, such as underlines should be used, in addition to color.

Actionable elements are triggered only on removal of touch (for more information see [SC 2.5.2: Pointer Cancellation](#)).

Navigational aids

Visual indicators, (such as arrows, next and previous buttons) have been used to indicate swipe or scroll areas or additional functionality.

Single character key shortcuts can be turned off, modified by the user or are active only on focus (for more information see [SC 2.1.4: Character Key Shortcuts](#)).

Blocks of content have descriptive headings (for more information see [SC 2.4.10: Section Headings](#)).

Prior to starting a process, users are warned if there is a timeout and the length of time of inactivity that will trigger the timeout. Please note you will also need to comply with WCAG2 SC 2.2.1 Timing Adjustable (for more information see [SC 2.2.6: Timeouts](#)).

[NA] Navigation features such as back buttons, breadcrumbs, next and previous buttons are provided (for more information see [SC 2.4.8: Location](#)). [END NA]

[MS] Navigation features such as breadcrumbs, next and previous buttons are provided (for more information see [SC 2.4.8: Location](#)). [END MS]

[MS] ARIA document landmarks have been used to appropriately describe document structure. [END MS]

Audio and Video

All audio and video have an accessible transcript (for more information see [SC 1.2.8: Media Alternative \(Prerecorded\)](#)).

Captions must be Closed Captions.

Live captions and audio descriptions are provided for any live audio or video.

Forms

Visual and audio CAPTCHAs are not used.

All complex forms contain context-sensitive help as instructions at the beginning of the form and/or specific instructions at each field (for more information see [SC 3.3.5: Help](#)).

All submitted forms are reversible by the user, checked for errors by the application or confirmed prior to submission (for more information see [SC 3.3.6: Error Prevention \(All\)](#)).

Field labels are positioned adjacent to their input field, and appear closest to their respective input field in relation to other field labels and other input fields.

Fields must have an associated visible label which is also programmatically associated with the field (please note that placeholder characters do not meet this requirement) (for more information see [SC 2.5.3: Label in Name](#)).

If there is visible text label for a input field or component, the accessible name matches the visual name (for more information see [SC 2.5.3: Label in Name](#)).

[NA] Forms interact appropriately with the keyboard, for example, providing submission on Enter, moving between fields and can be dismissed. [END NA]

[MS] Forms interact appropriately with the keyboard, for example, providing submission on Enter and moving between fields. [END MS]

[MS] All HTML5 INPUT TYPES and AUTOCOMPLETE values are used according to the specification and where they are applicable (for more information see [SC 1.3.5: Identify Input Purpose](#) and particularly the list of [input types that should be used](#)). [END MS]

[MS] Mobile / desktop interaction

Item labelling between different types of a site (desktop, m.dot and/or responsive), and different variations of a responsive site, is consistent.

Links between different types of a site (desktop, m.dot and/or responsive) have been provided, where the site is not solely a responsive site.

Step 5: Test mobile assistive technology and feature support

All actionable items and important content can be accessed and activated by assistive technologies or mobile features.

Why test with assistive technologies and mobile features?

Mobile devices provide a multitude of accessibility features. As an accessibility tester for mobile devices, it is important that you understand what these features are, how they work, and how they improve an experience for someone with a given disability. [MS] As both mobile operating

systems and mobile assistive technology are structured in a different manner from their desktop counterparts, it is important that mobile testing is conducted independently from desktop testing of a website. [END MS] As a rule of thumb, if a website or native app is not compatible with a device's assistive technology then the website or native app is inaccessible.

[NA] Native App inheritance of settings

Native applications should permit the following user preferences to be inherited from platform settings:

- Color
- Contrast
- Font type
- Font size
- Focus
- Mono audio
- LED flash for alerts (iOS) / Flash alerts (Android) [END NA]

Assistive technologies and mobile features

Modern mobile devices provide a multitude of accessibility features, many of them go above and beyond common accessibility testing methodologies. This document was created to provide guidance for the most recognized accessibility features, with particular attention to features that are common on all device types.

For the latest information on screen reader usage, please see the [WebAIM Screen Reader Survey](#).

Samsung includes an additional screen reader called “Voice Assistant,” however TalkBack is still available as part of the Accessibility Suite. Amazon Fire also utilizes a different screen reader called “Voice View”. These may need to be incorporated into testing.

Please note that iOS 13 features, such as Dark Mode and Voice Control are not included in this list. [MS] Please note: As of July 2019, “dynamic type” on iOS does not change the appearance of web pages in Safari. [END MS]

We recommend testing the following on iPhone and iPad:

- VoiceOver
- Keyboard
- Keyboard and switch
- Zoom
- Reduce Motion
- Invert colors
- Grayscale

- [NA] Larger Text [END NA]
- [MS] Reader View and increase text [END MS]

We recommend testing the following on Android:

- TalkBack
- Keyboard
- Keyboard and switch
- Magnification
- Remove Animations
- Color Inversion
- Grayscale
- [MS] Increase text size with Android Chrome [END MS]
- [NA] Increase font size [END NA]
- [NA] Increase display size [END NA]
- [MS] Simplified view [END MS]

Acknowledgements

Relationship to existing Accessibility testing standards

This document is based on:

- W3C Web Content Accessibility Guidelines, Version 2.0
- W3C Web Content Accessibility Guidelines, Version 2.1
- BBC Mobile Accessibility Guidelines
- AccessibilityOz Mobile Testing Methodology
- TPG Mobile Testing Guide

ICT Accessibility Testing Symposium Native App Sub-Committee

This document was developed by the ICT Accessibility Testing Symposium Native App Sub-Committee. Members include: Gian Wild (Co-Chair), Jennifer Chadwick (Co-Chair), Kathy Eng, Ryan Pugh, Kathryn Weber-Hottelmann, Brent Davis, Laura Renfro, Peter McNally, Karen Herr, Steve Sawczyn, Sunish Gupta, Tom Lawton, Sam Bouchat, Rafal Charlampowicz, Damon Wandke, Morgan Lee Kester, Mona Rekhi, Corbb O'Connor and Chris Law.

ICT Accessibility Testing Symposium Mobile Site Sub-Committee

This document was developed by the ICT Accessibility Testing Symposium Mobile Sub-Committee. Members include: Gian Wild (Co-Chair), Peter McNally (Co-Chair), Brent Davis, Corbb O'Connor, Karen Herr, Kathryn Weber-Hottelmann, Kathy Eng, Laura Renfro, Megha Rajopadhye, Mona Rekhi, Morgan Lee Kestner, Rafal Charlampowicz, Ryan Pugh, Steve Sawczyn, Sunish Gupta, Tom Lawton and Chris Law.

Copyright notice



The 2020 ICT Accessibility Testing Symposium proceedings are distributed under the Creative Commons license: Attribution-NonCommercial-NoDerivatives 4.0 International (CC BY-NC-ND 4.0, <https://creativecommons.org/licenses/by-nc-nd/4.0/>).

You are free to: Share — copy and redistribute the material in any medium or format.

Under the following terms: Attribution — You must give appropriate credit, provide a link to the license, and indicate if changes were made. You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use. NonCommercial — You may not use the material for commercial purposes. NoDerivatives — If you remix, transform, or build upon the material, you may not distribute the modified material. No additional restrictions — You may not apply legal terms or technological measures that legally restrict others from doing anything the license permits.

Workshop: Understanding ARIA 1.2 and the ARIA Authoring Practices Guide

Jon Gunderson, Ph.D., CWAP

Disability Resources and Education Services
University of Illinois at Urbana/Champaign
College of Applied Health Sciences
1207 S. Oak Street, Champaign, IL 61820
E-mail: jongund@illinois.edu

Abstract

Understanding the how the roles, properties and states defined in the W3C Accessible Rich Internet Accessibility (ARIA) 1.2 specifications are used by assistive technologies (e.g. screen readers) is critical for accessibility professionals to evaluate and provide guidance on how to identify and remediate web accessibility issues of online resources. ARIA technology is often not well understood by designers and developers as evidenced by high percentage of misused ARIA on the web. The first principle of using ARIA is no ARIA is better than bad ARIA. Misused or incomplete use of ARIA diminishes, rather than enhances the accessibility of a web resource. The purpose of this workshop is to help participants understand how ARIA is designed to support the creation of accessible web resources that are work with a wide range of browsers and assistive technologies.

Communicating Accessible Design Patterns

When accessibility issues are identified, or inappropriate use ARIA markup is found the person making the evaluation needs a way to communicate the correct information to the designers and developers. An important resource in communicating proper accessibility techniques to designers and developers is the W3C ARIA Authoring Practices Guide (APG). The APG is a reviewed reference for supporting the requirements of the W3C Accessible Rich Internet Application 1.2 specification and the use of HTML5 native semantics to meet W3C WCAG requirements. The APG provides the technical information need by interaction designers, developers and quality assurance personnel to design and test compliance with for WCAG requirements using the ARIA and HTML5 standards. The APG provides detailed information on the proper use of ARIA properties, states and roles to represent the interactive web resources to users of assistive technologies. The APG includes examples for landmarks and widget roles, and design information to support keyboard navigation between and within widgets, and how to describe relationships between content in web resources using HTML5 tags and ARIA markup. The APG examples have been extensively reviewed and tested with assistive technologies to help people understand and test the accessibility of ARIA enabled widgets.

Workshop Objectives

- Keyboard interaction models for role semantics
- Understand the concepts of role, properties and states
- Native semantics of HTML elements
- Removing HTML native semantics with role “none”
- Design patterns for widget roles
- Defining accessible names and descriptions
- Keyboard focus styling for high contrast support
- How screen readers use ARIA markup
- Mobile accessibility issues

Online Experience

All the slides used in the course are in HTML and will be publicly available during and after the workshop. During asynchronous sessions, the presenter will be available to answer questions through zoom. The workshop will take place over 2 days, with both synchronous and asynchronous sessions. Day 1 will include 1.5 hours of synchronous sessions and 2.5 hours of asynchronous sessions. Day 2 will be 1.5 hours of synchronous sessions. The asynchronous sessions allow people to go through the examples and test with assistive technologies at their own pace.

Part 1: Introduction and Basics of ARIA (Day 1, 60 minutes Synchronous)

- Workshop participants will introduce themselves at the beginning of the workshop, so people get to know who else is in the workshop.
- Overview of ARIA resources that will be used in the course:
 - ARIA specification
 - ARIA Authoring practices
 - Accessible Name specification
 - Core Accessible API Mapping specification
- Browser versus Assistive Technology support for ARIA
- ARIA landmark versus widget roles
- Widget design principles
 - Keyboard support
 - Roles, properties and states

- Focus styling
- High contrast support
- Mobile support
- Workshop Examples
 - Checkbox Example
 - Menu Button Example
- Using Webedit for testing workshop examples
 - Setting up account
 - Finding examples
- Testing examples:
 - Visual focus indicator
 - Windows/macOS high contrast modes
 - JAWS for Windows
 - NVDA for Windows
 - VoiceOver for macOS

Part 2: ARIA enabled custom Checkbox (Day 1, 45-minute Asynchronous Video with Assignments)

- Keyboard interaction model for checkbox pattern
- Roles, properties and states for checkbox pattern
- Accessible name
- Testing with a screen reader
- Focus styling
- High contrast support

Part 3: Review Custom Checkbox Widget (Day 1, 30 minutes Synchronous Session)

- Assistive technology experience of checkbox examples
- Focus styling
- High contrast support

Part 4: Menu Button Design Pattern (Day 2, 45 minutes Asynchronous Video)

- Keyboard interaction model for menu button pattern
- Roles, properties and states for menu button pattern
- Accessible name
- Testing with a screen reader
- Focus styling
- High contrast support
- Roving tabindex versus aria-activedescendant

Part 5: Review Menu Button Review (Day 2, 30 minutes Synchronous)

- Assistive technology experience of menu button examples
- Differences in using roving tabindex versus using aria-activedescendant
- Focus styling
- High contrast support
- Mobile support

Part 6: New Features in ARIA 1.2 and Questions (Day 2, 60 minutes Synchronous)

- HTML harmonization in ARIA 1.2
- Combobox 1.2 pattern
- ARIA attributes as DOM properties
- Questions and discussion

Copyright notice



The 2020 ICT Accessibility Testing Symposium proceedings are distributed under the Creative Commons license: Attribution-NonCommercial-NoDerivatives 4.0 International (CC BY-NC-ND 4.0, <https://creativecommons.org/licenses/by-nc-nd/4.0/>).

You are free to: Share — copy and redistribute the material in any medium or format.

Under the following terms: Attribution — You must give appropriate credit, provide a link to the license, and indicate if changes were made. You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use. NonCommercial — You may not use the material for commercial purposes. NoDerivatives — If you remix, transform, or build upon the material, you may not distribute the modified material. No additional restrictions — You may not apply legal terms or technological measures that legally restrict others from doing anything the license permits.

Panel: Accessibility Overlays

Chris M. Law (Chair)

Accessibility Track
4520 39th St N, Arlington, VA, 22207, USA
chrismlaw@accessibilitytrack.com

Karl Groves

Tenon.io
309 Ferndale Road Glen Burnie, MD 21061,
USA
karl@tenon.io

Al Hoffman

Deque Systems, Inc.
381 Elden St Suite 2000, Herndon, VA
20170, USA
allen.hoffman@deque.com

Anil Lewis

National Federation of the Blind
200 E Wells St, Baltimore, MD 21230, USA
alewis@nfb.org

David O'Neill

The Paciello Group
17757 US 19 North, Suite 560, Clearwater,
FL 33764, USA
doneill@paciellogroup.com

Cyndi Rowland

WebAIM
Center for Persons with Disabilities,
Utah State University
6807 Old Main Hill, Logan, UT, 84322,
USA
cyndi.rowland@usu.edu

Introduction

‘Accessibility overlays are applications and coding that “sits” between your website (or other digital content) and assistive technology. Their goal is to allow business owners and compliance managers the ability to make their website more accessible without having to change the underlying source code, thus providing accessibility at a fraction of the time and cost, comparatively speaking.’ Henry, B. (2020)

For decades, accessibility testing professionals have been advocating that to get to an accessible website, you need to fix the underlying code. If you want to make change lasting, you need to fix coding practices that lead to inaccessible code, through training and implementation of governance procedures in development. Those who promote and sell accessibility overlay technologies have espoused an alternative approach, whereby you can “use just one line of code” (Groves, 2018) to pull in their tool, effectively absolving the customer of the need to fix their code and their programming practices.

Over the years, the ICT Accessibility Testing Symposium has featured many papers and presentations advocating testing and development processes centered on fixing the underlying code. The subject of overlays has not received much attention at the conference, with only occasional discussions regarding their existence, and things like how well they work and the consequences of the incompatibility of the two approaches.

In 2020, two key things spurred the formation of this Panel discussion:

Overlays have been in existence for many years, and many overlay producers have made claims of ‘full’ or 100% conformance with accessibility standards through their use. However, this year, the proliferation of such claims and the growing number of lawsuits on digital accessibility has prompted many in the accessibility field to publish on the fallacy of such grandiose claims (e.g., Springer, 2020a, Feingold, 2020, Faulkner, 2020).

Overlays have been found to ‘spooft’ automated checkers (Roselli, 2020). Roselli notes that he found this when using an automated website checker tool (he gives the example of WebAIM’s WAVE: wave.webaim.org). The overlay tool was detecting use of the WAVE checker and then modifying the behavior of the website to provide more favorable results to the checker that do not represent the default presentation of the page.

It is clear that overlays are impacting accessibility testing, and we need to explore the impact of the proliferation of overlays on accessibility testing practice.

Notes on related technologies:

To keep this panel focused on one issue, we will be keeping to discussion focused on overlays. However, there are related technologies that often fall into the discussion of website accessibility add-ons:

Accessibility Widgets are add-on menus that give users a range of accessibility options and settings for visiting a site (with this technology, users would have to set their preferences for every site they visit). Most often, accessibility widgets and overlays are packaged together in one solution (Springer, 2020b).

Quality Assurance Checkers/Monitors provide a ‘dashboard’ view to site administrators of what is happening by automatically testing a certain subset of accessibility metrics or their organization’s website (Law, 2016). These tools are used to provide developers a numerical score of compliance and links to errors that need addressing. These tools do not test all features, and so are best used as QA tools on an existing accessible site whose content is being updated by many authors (e.g., news sites).

Questions for the panelists

In this panel, we will first briefly introduce the concept of overlays, what they do and how they generally work. We will then pose the following questions to begin the discussion:

- Can claims such as ‘100% accessibility conformance’ be substantiated with respect to overlays?
- When testing a website for accessibility, and an overlay is present, should the site be tested with or without the overlay?
- What should accessibility testers do when it is known that overlay coding is modifying test results?

- Could/should there be a collective statement on overlays, and/or guidance from members of the professional accessibility testing community to website developers, the public, and other concerned citizens.

References

Faulkner, S. (2020). Bolt-on Accessibility – 5 gears in reverse. 13 May, 2020. Available: <https://developer.paciellogroup.com/blog/2020/05/bolt-on-accessibility-5-gears-in-reverse/>

Feingold, L. (2020) **Honor the ADA: Avoid Web Accessibility Quick-Fix Overlays**. 10 August, 2020. Available: <https://www.lflegal.com/2020/08/quick-fix/>

Groves, K. (2018). **Automated Lies, with one line of code**. 25 May, 2018. Available: <https://karlgroves.com/2018/05/25/automated-lies-with-one-line-of-code>

Henry, B. (2020) **Accessibility Overlays in Digital Content**. 13 May, 2020. Available: <https://www.paciellogroup.com/accessibility-overlays-in-digital-content/>

Law, C.M. (2016) **QA Enabled Testing as the Hook for Organizational Change**. The ICT Accessibility Testing Symposium, Baltimore, MD, November 2-3, 2016. pp.47-54.

Roselli, A. (2020) **#accessiBe Will Get You Sued**. June 29, 2020. Available: <https://adrianroselli.com/2020/06/accessibe-will-get-you-sued.html>

Springer, T. (2020a). **Is Deploying an Accessibility Overlay or Widget Better Than Doing Nothing?** 14 July, 2020. Available: <https://www.linkedin.com/pulse/deploying-accessibility-overlay-widget-better-than-doing-springer/>

Springer, T. (2020b). **Lies, Damned Lies, Overlays and Widgets**. 25 June, 2020. Available: <https://www.linkedin.com/pulse/lies-damned-overlays-widgets-timothy-springer/>

Copyright notice



The 2020 ICT Accessibility Testing Symposium proceedings are distributed under the Creative Commons license: Attribution-NonCommercial-NoDerivatives 4.0 International (CC BY-NC-ND 4.0, <https://creativecommons.org/licenses/by-nc-nd/4.0/>).

You are free to: Share — copy and redistribute the material in any medium or format.

Under the following terms: Attribution — You must give appropriate credit, provide a link to the license, and indicate if changes were made. You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use. NonCommercial — You may not use the material for commercial purposes. NoDerivatives — If you remix, transform, or build upon the material, you may not distribute the modified material. No additional restrictions — You may not apply legal terms or technological measures that legally restrict others from doing anything the license permits.

Panel: COVID-19: The Great Accelerator (Part 1)

Mark Miller (Chair)

Director Sales, Emerging Accounts & Platform
The Paciello Group
17757 US 19 North, Suite 560, Clearwater, FL 33764, USA
mmiller@paciellogroup.com

Kathy Wahlbin

General Manager & VP Enterprise Compliance
kwahlbin@paciellogroup.com

Todd Waites

Director, Business Development
twaites@paciellogroup.com

Laura Miller

Corporate Business Development Manager
lmiller@vispero.com

David Sloan

UX Research Lead
dsloan@paciellogroup.com

Introduction

In this panel session we examine COVID's acceleration and forced adoption of digital communication channels for different roles and industries. In future years, COVID-19 may be known as the "Great Accelerator" as a result of its tremendous impact on everything from remote working to global supply chains. You would be hard-pressed to find an industry or an organization that has not experienced significant shock and forced into adopting new operational procedures that may have never been previously considered.

A global pandemic is an extraordinary situation, one that requires quick pivots and rapid advancement in order to survive. Without the luxury of bureaucratic processes slowing the adoption of new ideas, firms are forced to bravely grasp at whatever vines may be available to them to swing over the gaping ravine of the rapidly evolving situation.

Accessibility and the COVID-19 Response

Part of this new adoption has been a newfound reliance on digital technology and communication channels. Their employees forced into quarantine and living under shelter-in-place guidelines, firms with the option to do so have had little choice but to utilize organization-wide communication platforms like Slack or Trello, and video-conferencing applications like Zoom to keep their organizations afloat.

Whole industries have undergone remarkable revolutions in a startlingly short time period. Restaurants, starved of the patrons that fill their seats, started offering curbside pickup and food delivery almost immediately. While many of them opted to partner with an existing food delivery service like Uber Eats or Grubhub, many others elected to set up food ordering on their own websites. Grocery stores, already half-heartedly embracing delivery and curbside pickup for convenience, were scrambling to extend their ability to accommodate and overflow of demand as customers shied away from in-person visits. Schools may have been hit most dramatically of all. Their struggle to accommodate students of all types of abilities has been fraught with pitfalls and accessibility barriers. Not only do they need video conferencing technology, but they also require an operational model for receiving homework, giving and grading tests, and a repository for all the normal paper detritus that accumulates over the length of a school year.

These are just a few well-documented examples of how COVID-19 has forced industries and companies to turn on a dime to frantically resume pre-COVID productivity. Yet with all the madness that has ensued in this post-COVID world, accessibility becomes an even more critical consideration. Individuals with disabilities are often the ones hardest hit by global turmoil, as they are already a more vulnerable portion of the population. Not to mention the legal implications of inaccessible digital content. The ADA already prohibits discrimination on the basis of ability, and this has been de facto extended to digital content.

Adapting Roles

This panel will discuss the experiences from a variety of roles on how COVID-19 has forced them to modify their normal routines and embrace digital technology as a replacement for in-person communication.

Goals

- Highlight how COVID has changed the way specific roles rely on digital technology and what considerations need to be in place to accommodate people with disabilities
- Discuss different industry impacts (healthcare, education, ecommerce) and what changes we've already seen taking place
- Emphasize the necessity of accessibility when considering new technology
- Discuss legal implications of inaccessibility

Panelists

This panel is presented by members of The Paciello Group. The roles that will be addressed in session are:

- Legal perspective: *Kathy Wahlbin, General Manager & VP Enterprise Compliance*
- Business Development: *Todd Waites, Director, Business Development*
- Healthcare/Hospitality Kiosk Accessibility: *Laura Miller, Corporate Business Development Manager*
- UX Research: *David Sloan, UX Research Lead*

Copyright notice



The 2020 ICT Accessibility Testing Symposium proceedings are distributed under the Creative Commons license: Attribution-NonCommercial-NoDerivatives 4.0 International (CC BY-NC-ND 4.0, <https://creativecommons.org/licenses/by-nc-nd/4.0/>).

You are free to: Share — copy and redistribute the material in any medium or format.

Under the following terms: Attribution — You must give appropriate credit, provide a link to the license, and indicate if changes were made. You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use. NonCommercial — You may not use the material for commercial purposes. NoDerivatives — If you remix, transform, or build upon the material, you may not distribute the modified material. No additional restrictions — You may not apply legal terms or technological measures that legally restrict others from doing anything the license permits.

Panel: COVID-19: The Great Accelerator (Part 2)

Matt Feldman (Chair)

The Paciello Group
17757 US 19 North, Suite 560, Clearwater, FL 33764, USA

Corbb O'Connor

Siteimprove
7807 CreekrIDGE Circle, Minneapolis, MN 55439, USA
cooc@siteimprove.com

Markku Häkkinen

Educational Testing Service (ETS)
Rosedale Road, Princeton, NJ, 08541, USA
mhakkinen@ets.org

Mike Paciello

Peterborough, NH, USA
+1 603.566.7713
mikepaciello@gmail.com

Madeleine Rothberg

WGBH National Center for Accessible Media
WGBH Educational Foundation,
1 Guest Street, Boston, MA 02135, USA
madeleine_rothberg@wgbh.org

Introduction

In the Part 1 Panel, the roles of various personnel were addressed in the context of the response to the Covid-19 pandemic with respect to accessibility generally, and accessibility testing in particular. In Part 2, we look to the near-term future and what we changes we might reasonably anticipate in the practice of accessibility testing.

For example, what is the future of remote work, commerce, education, customer support, etc., and how will it affect accessibility testing programs? What can we do as a professional field? Where should we be focusing our energies?

Panelists will be asked to directly address the issues raised in the Part 1 Panel, as well as provide their commentary on the papers presented at this year's symposium that have relevance to the near-term future.

Past ICT Closing Panels

“The future ain't what it used to be.”

—attributed to various authors (see O'Toole (2012) for an interesting explanation)

In past years, the ICT Accessibility Testing Symposium committees have closed the conference with a Panel looking toward the near-term, medium-term, and long-term future for our profession. (For participants of past closing plenary panels on ‘Accessibility Testing: The Present, The Future’, see the archived programs of previous ICT symposia at www.ictaccessibilitytesting.org/programs-and-proceedings/.)

Of course, none of us knew in 2019 what would be in store for 2020.

In this strange year the whole world was wrapped up in a massive social change necessitating a huge increase in online distance learning, work, and social contact. This has in turn had a huge impact on the accessibility testing field.

Can we really predict what will happen in 2021, 2022? Probably not with great accuracy because predictions in the wider world have the same problem we do: not enough hard data. But, we can make educated guesses. So, for this year, we are going to corral the panel conversation to focus mainly on the short-term future, and how accessibility professionals can best place themselves to be ready, adaptable, (and yes... agile!) as we weather the Covid-19 ‘storm’ together.

We all hope and trust that normal service returns next year, with our regular annual peek into the future of accessibility testing.

Reference

O'Toole, G. (2012) **The Future Is Not What It Used To Be: Yogi Berra? Paul Valéry? Laura Riding? Robert Graves? Anonymous?** Quote Investigator website. Available: <https://quoteinvestigator.com/2012/12/06/future-not-used/>

Copyright notice



The 2020 ICT Accessibility Testing Symposium proceedings are distributed under the Creative Commons license: Attribution-NonCommercial-NoDerivatives 4.0 International (CC BY-NC-ND 4.0, <https://creativecommons.org/licenses/by-nc-nd/4.0/>).

You are free to: Share — copy and redistribute the material in any medium or format.

Under the following terms: Attribution — You must give appropriate credit, provide a link to the license, and indicate if changes were made. You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use. NonCommercial — You may not use the material for commercial purposes. NoDerivatives — If you remix, transform, or build upon the material, you may not distribute the modified material. No additional restrictions — You may not apply legal terms or technological measures that legally restrict others from doing anything the license permits.

How to evaluate video conferencing tools for accessibility: Navigating new features and fixes every week

Claudio Luis Vera

Fort Lauderdale, FL, USA
modulist@gmail.com

Keywords: Videoconferencing tools, work from home, remote testing

Track/Theme: Testing times

Abstract

What should someone be looking for when evaluating different videoconferencing options for accessibility? Captions, screen reader and keyboard accessibility, and advanced features are just a few of the factors to consider. With us all suddenly working from home in the COVID era, most employers have had to make some quick choices from a huge variety of video conferencing and collaboration tools. Often, accessibility is an afterthought and the organization is bound to a solution that excludes people with disabilities. With the right solution, working from home is a blessing for those with a disability. It eliminates transportation issues and allows employees to attend to medical needs in ways that might be disruptive in an office, like using a ventilator or an IV. With the *wrong* tools, a person with a severe disability may actually experience *new* barriers. They could be left out of meetings and other types of collaboration, and leave the employer vulnerable to a discrimination lawsuit. A common mistake when setting up online meetings or events is to assume that every attendee is fully able to participate: Unless the organizer knows everyone's ability level, they should ask if someone needs accommodations before setting up the meeting. Otherwise, they should assume that at least one of the attendees may have difficulty seeing, hearing, or working a mouse.

Criteria for evaluation

There are several factors that should be considered when evaluating different video conferencing platforms:

1. Does the platform offer built-in automated captions? What is the accuracy of these captions?
2. Do they offer the ability to use third party captioning?
3. Which of the platform's functions can be performed using a keyboard?
4. How much of the platform's functionality is accessible to a screen reader?
5. Does the platform's interface allow for magnification and zooming of the user interface? What about content?
6. Are all the platform's features, such as chat, polling, and Q&A fully accessible?
7. Can people with disabilities participate in ways that go beyond merely attending? Can they schedule, host, or present in meetings?
8. Has the provider expressed a commitment to inclusion in a public accessibility statement? Do they have a VPAT for their product?

The core roles

Almost all video conferencing apps have three different roles that a user can play: host, presenter, and participant. The names of these roles may vary with the app, but the functions are fairly standard: A *host* schedules the meeting, works as the master of ceremonies, and manages the permissions; a *presenter* or *panelist* is a speaker that will often share their screen. A participant or attendee will see the content and discuss it without presenting themselves. Of these roles, playing host has the greatest functionality and requires the greatest knowledge to perform. It's also the most difficult role to support accessibly.

Different needs for different impairments

While reviewing user research for this paper, it was surprising to see that people with certain disabilities would prefer a platform which could be complete unusable to a user with different set of disabilities. For example, users with low vision used Webex more frequently all platforms, while blind users would find critical bugs that would prevent them from using it entirely.

Zoom: the favorite for accessibility

While Zoom is generally favored by the accessibility community, its reputation comes mostly from shared opinions and anecdotes, as opposed to the findings from a formal industry study. In April 2020, Fable Tech Labs conducted an informal survey on a small population of people with a range of disabilities. In that sample, 80% of respondents listed Zoom as their tool of choice, with a few calling out the ability to join a meeting from an email invitation in a single click. In

contrast, other platforms (like Webex) require an attendee to navigate modal dialogs, which can present barriers.

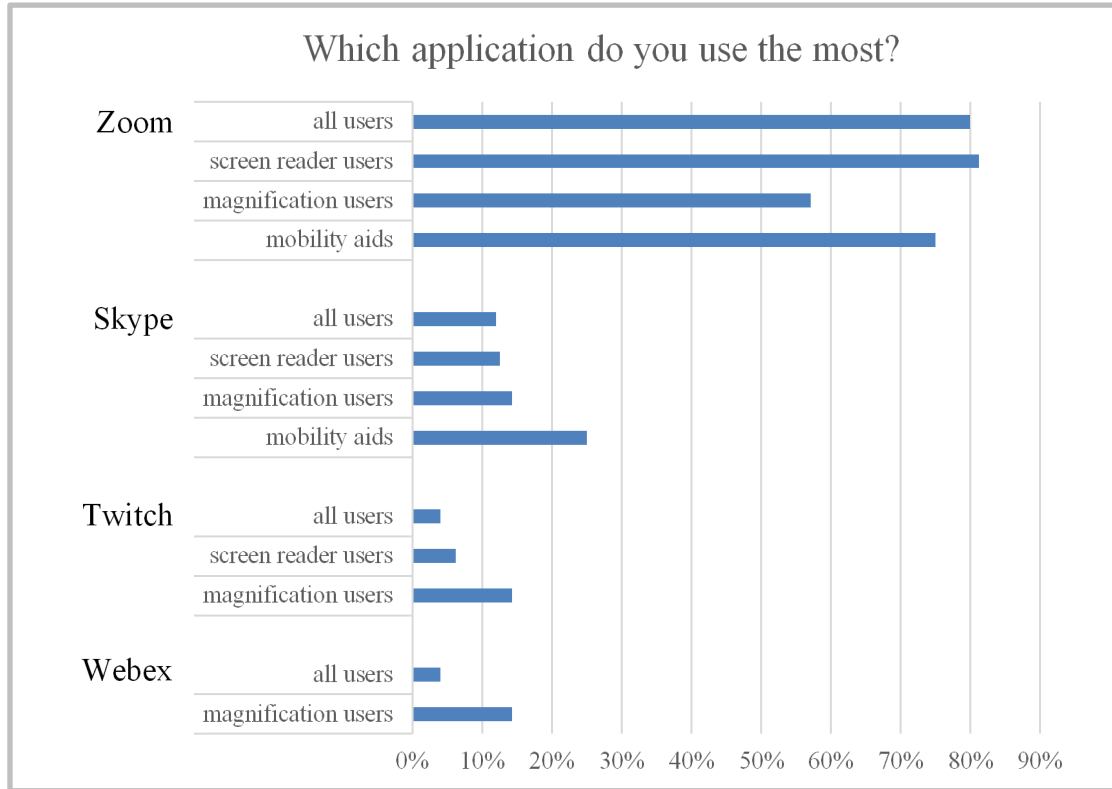


Figure 1: Results from the April 2020 Fable Survey

Zoom’s approach to products favors universal design over separate applications or modalities for different user types. The result is an easy-to-learn app for first-time use, combined with layer upon layer of additional features. In the audiobook *Meet Me Accessibly* (<https://mosen.org/zoom/>), podcaster Jonathan Mosen provides tutorials that teach easy to advanced tasks to fellow blind Zoom users. However, many of the more complex features in Zoom like Q&A or Polls are not accessible to all disabilities. Hosts should stick to the more vanilla features in larger meetings and be ready to provide alternate formats for interaction.

Use of captions

Captions are a lifeline for the deaf and the hard of hearing, and quite helpful to hearing users in a variety of other situations as well.

For everyday business use where the host is not certain that all attendees can hear well, meetings should have real-time captions that are automatically generated and can be toggled on and off by each attendee. For official meetings or larger webinars, live captions should be typed in real time by a human captioner using Communication Access Realtime Translation (CART).

The accuracy of captions varies greatly, depending on the method being used. Live captioning by a human using CART can come close to 100% accuracy. Automatically generated captions depend largely on the speaker's diction and accent, as well as the quality of the audio equipment. As a native speaker of American English without a heavy dialect automated captions, I regularly experience a rate of about 93% accuracy in recorded presentations. One of my colleagues, who is hard-of-hearing and has slightly atypical speech, typically has an accuracy rate of about 85% to 88%.

When the accuracy of captions falls below 90%, captions can become distracting or a serious annoyance, which is why they should be able to be turned on or off by the attendees. Inaccurate captions also require at least double the effort when creating transcripts.

A presenter should avoid generating automatic captions in the presentation itself – as opposed to within the meeting platform. For example, both PowerPoint and Google Slides provide automated captions if the presenter activates them within the app. While this feature is very useful for a live in-person presentation, the captions appear as part of the video stream and can't be turned off by the end user. In a recorded meeting, the captions from the presentation become an indelible part of the recorded video which can't be removed or edited.

Google Meet and Microsoft Teams both provide automated closed captions which are independent of the meeting's video. Both can also be set up to leverage the participants' login information to identify the speaker in the captions.

Judging apps by their accessibility pages

One good way to gauge a company's commitment to accessibility is to look for an accessibility page on their website. A company that is at least *aware* of accessibility will have a posted accessibility policy. If the platform doesn't have an accessibility policy, it's more than likely that any accessibility is accidental and not the product of a systematic approach.

A handful of companies post Voluntary Product Accessibility Template (VPAT) statements to meet Section 508 requirements. These are a goldmine of accessibility compliance information and should be reviewed by anyone in charge of making a purchasing decision.

VPATs classify the level of support for users with disabilities and assistive technologies into four categories:

1. Supports
2. Supports with Exceptions
3. Does not Support
4. Not Applicable.

Of these options, anything other than category "1. Supports" should be treated as a potential problem area. Companies often label their inaccessible features as "Supports with Exceptions".

If an app has advanced features or functionality that's not directly connected to video conferencing, then support can be iffy, as mentioned earlier with Zoom. A good rule of thumb would be to stick to basic features when accessibility is a concern.

Legal activity

Webex's spotty support for screen readers has led to a lawsuit against one of its customers: in October 2019, a blind plaintiff in Massachusetts sued her employer for not providing accessible alternatives like Zoom for Webex meetings. (see <https://www.nfb.org/about-us/press-room/blind-employee-sues-beacon-health-options>) Beacon Health Options, the employer, was using Webex for virtual conferences and mandatory employee training in which Amy Ruell, the plaintiff, was unable to participate. The case is still ongoing at the time of this writing.

Recommendations

This area lacks formal studies performed by a well-funded source, like an business analyst (e.g. Gartner, Forrester) or a major accessibility consulting firm. Large organizations should plan to conduct their own research as part of their procurement process and assemble their own documentation and training materials.

The field is constantly changing, as each provider rolls out major new features monthly. Anyone that uses *any* video conferencing software should be diligent about performing updates as frequently as possible — or enabling auto-updates on every device. Often, a serious accessibility issue may be resolved with a simple software update.

Unless a meeting host knows every attendee's ability level — as with immediate co-workers — they should ask if someone needs accommodations. If that's not feasible, then they should assume that at least one attendee will have difficulty seeing, hearing, or working a mouse — and plan accordingly.

Copyright notice



The 2020 ICT Accessibility Testing Symposium proceedings are distributed under the Creative Commons license: Attribution-NonCommercial-NoDerivatives 4.0 International (CC BY-NC-ND 4.0, <https://creativecommons.org/licenses/by-nc-nd/4.0/>).

You are free to: Share — copy and redistribute the material in any medium or format.

Under the following terms: Attribution — You must give appropriate credit, provide a link to the license, and indicate if changes were made. You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use. NonCommercial — You may not use the material for commercial purposes. NoDerivatives — If you remix, transform, or build upon the material, you may not distribute the modified material. No additional restrictions — You may not apply legal terms or technological measures that legally restrict others from doing anything the license permits.

Broadening the definition of ‘interaction’ for accessibility testing

Dr. Chris M. Law

Accessibility Track
4520 39th St N, Arlington, VA, 22207
chrismlaw@accessibilitytrack.com

Anil Lewis

National Federation of the Blind
200 E Wells St, Baltimore, MD 21230
alewis@nfb.org

Keywords: Accessibility Guidance; Mobile Interfaces; Smart Homes; Internet of Things (IoT); Remote Support.

Track/Theme: Accessibility and providing online customer and technical support, Remote ICT testing, Mobile Websites, Mobile Applications, Closed Products (Kiosks, ATMs etc.), Other (Home Electronic Devices).

Abstract

A study was conducted on the use of consumer products by people who are blind. We aimed to assess the suitability and utility of current consumer guidance developed as part of the National Federation of the Blind’s Accessibility Switchboard Information Portal. Three key findings from the study for accessibility developers and testers are that (1) there is a massive interaction gap between personal devices (smartphones, PCs) and other public and home technology products; (2) the accessibility testing field’s traditional focus on direct interaction should be broadened to include new forms of indirect interaction; and (3) ‘interaction’ should also account for engagement with third parties, including customer support, friends, family, co-workers and even helpful strangers/bystanders. We propose a number of new guides to take account of the broader context of interaction, and put forward our case for the broader perspective to be more widely employed the accessibility testing field.

Introduction

A study was conducted as part of the Accessibility Switchboard project, on the use of consumer products by people who are blind. The Accessibility Switchboard Information Portal is a project “to generate new, freely available resources—guides, and other articles—for readers who want to bring about changes to their situation, and changes to the organizations they work in.” The Switchboard has been under development since 2016 (Law & Lewis, 2016) and contains a number of guidance articles for consumers. In this paper, we extrapolate from the study of consumers, and present key findings for technology developers and accessibility testers. These findings specifically relate to the definition of ‘interaction’ as it relates to accessibility (and usability) testing. We argue that the definition of ‘interaction’ should be considered with a broader scope than is currently represented by publications on accessibility testing research and practice.

Methods

NFB has gathered a volunteer user group for periodically testing products, and collecting related feedback. Nineteen members of the Blind Users Innovating and Leading Design (“BUILD”) group participated in a survey, and ten of those participated in one of two follow-up focus groups that were conducted online via the Zoom meeting platform.

The average age of participants was 38 (high = 64; low = 19). Twelve participants had no functional vision; Three could see the outlines of small items but could not read large print, and four could read large print in high contrast only. Ten participants had been blind since birth, seven for over 10 years, and two for over 5 years.

An initial survey assessed general opinions and usage of consumer products. We then sent a follow-up survey that required participants to read five Accessibility Switchboard pages in order:

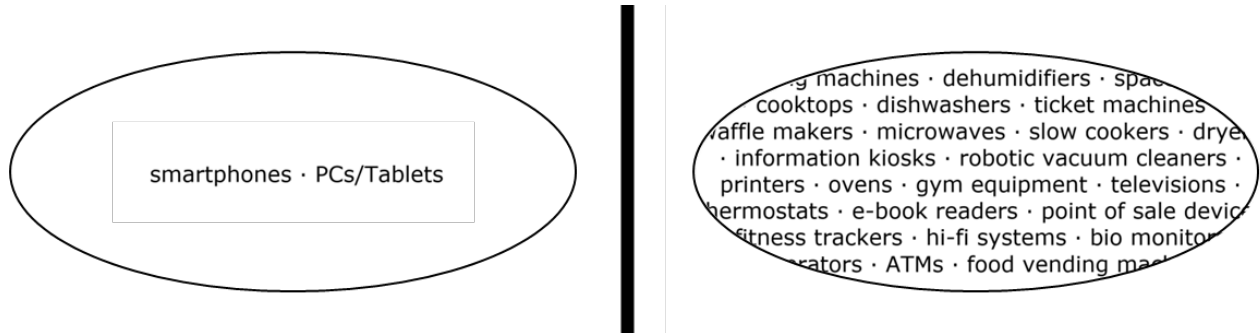
5. The home page of the site (<https://accessibilityswitchboard.org/>)
6. A consumer guide entitled “Documenting a day in your life: demonstrating the level of accessibility of the technologies you interact with, and those you cannot interact with” (<https://accessibilityswitchboard.org/documenting>)
7. A consumer guide entitled “How to approach change makers: what are the pros and cons of various options?” (<https://accessibilityswitchboard.org/changemakers>)
8. A consumer ‘Question and Answer’ article entitled “Where do I get good / accurate information on ICT (web, software, mobile, electronic document) accessibility?” (<https://accessibilityswitchboard.org/information>)
9. The site’s contact form (<https://accessibilityswitchboard.org/contactnfb>)

Subsequent to the follow-up survey, participants were asked to join an online focus group to discuss their use of consumer products, as well as the Switchboard guidance.

Participants were provided a small ‘thank-you’ gift card for each stage of participation.

Findings and Discussion

Finding #1: In terms of interaction, there is a massive gap between personal devices and other public/home technology products



“I can use phones and computers independently...”: 19/19 (**100%**)

versus...

“I can easily use home technology products and services independently and without any modification”: 1/19 (**5%**)

“I never have to ask for assistance from sighted people in order to use public technology products and services”: 3/19 (**16%**)

We asked participants if they had difficulty using personal devices such as smartphones and personal computers independently. These types of device typically have built-in accessibility features that can be activated, or such features can be added using freely available software. All responded in the affirmative, that they can use these devices independently.

Compare this to the very wide variety of other types of public and home use electronic products. This list includes things like washing machines and microwave ovens, ATMs and food vending machines. Specifically on the use of home products:

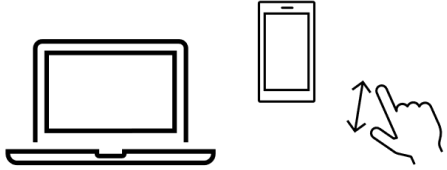
- Only one participant said that they can easily and independently use home technology products (5%).
- Four said they can use home devices independently, but with difficulty
- Thirteen said they add braille labels or tactile markers to home devices (giving access to control names, but not device displays)
- One said that they seldom use home devices independently

Many public use technologies are subject to accessibility laws, regulations and guidelines requiring them to be made independently usable by people with disabilities, including use without vision (e.g., under the Americans with Disabilities Act). Some commonly used devices have managed to stay exempt from such regulations despite their ubiquity, notably food and beverage vending machines. We did not make this distinction between regulated and unregulated devices in our questions, but we did ask whether participants could independently use public devices. Only three participants said that they never have to ask for assistance in using public devices (16%).

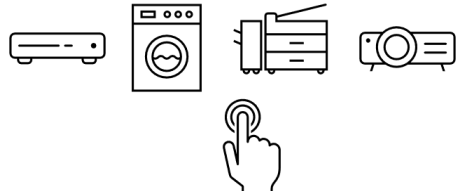
The implications of this finding are clear: For accessibility practitioners, there is a lot of fertile ground to cover in the development and testing of technology devices that are not smartphones and PCs. But... *what about controlling home and public use devices with your smartphone...?*

Finding #2: ‘Interaction’ for accessibility testing should be broader than the user and the device

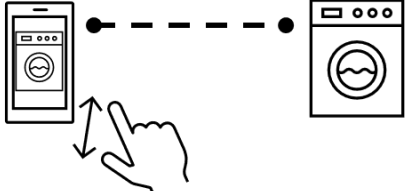
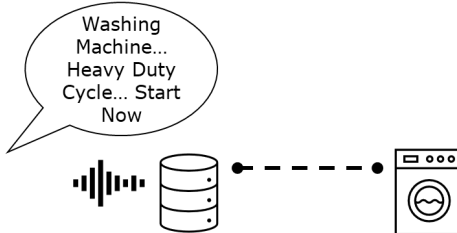
When accessibility and technology interface design was in its early days (i.e., through the last quarter of the 20th Century), direct interaction between the user and the device was almost exclusively the norm. Even with assistive technology (AT) being an add-on in the early days of the information age, the interaction with AT could still be considered ‘direct’. New forms of indirect and assisted interaction have emerged in the last decade. The lion’s share of development and testing in the accessibility field remains focused on direct interaction using smartphones and computers. Consequently, solutions are widely available—and the ‘100% independent use’ by our participants is testament to the successes of those who have pursued accessibility improvements in this form of interaction. Many problems still exist in terms of the ubiquity of solutions (e.g., Law, 2020), but they are available:

<p>User-Machine Direct Interaction (smartphones and PCs/Tablets):</p> 	<p>Where the vast majority of accessibility testing work is focused Many problems still exists, but in general independent nonvisual use is common</p>
--	---

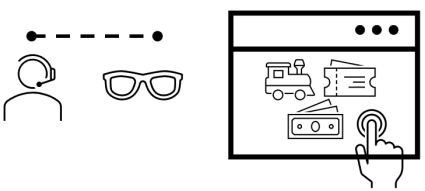
As we stated above, the majority of home-use technologies remain directly inaccessible to people who are blind, marking a massive interaction gap. This gap is also reflected in practice and reporting on accessibility testing over the years. For example, home technologies can be categorized under ‘hardware’ for testing purposes, and in the first four years of the ICT accessibility testing symposium, there has been only one paper presented on hardware testing (Ogami & Renfro, 2018). So, we unfortunately have a situation where few devices are directly accessible ‘out-of-the-box’ and very few papers (compared to web and mobile) on direct accessibility of such devices are published:

<p>User-Machine Direct Interaction (home/office and public use devices):</p> 	<p>An area where relatively little accessibility testing work has been focused Independent nonvisual use is problematic on a widespread basis When devices are made to be directly accessible (which is rare), affording to be able to replace existing devices isn't a realistic option for many, especially if they are more expensive than standard (inaccessible) versions</p>
---	--

Two other new form of interaction options have recently emerged in the marketplace. The first is using a smart phone application to indirectly control devices. The second is indirect use with virtual artificial intelligence (AI) assistants (e.g., [Alexa](https://developer.amazon.com/alexa), <https://developer.amazon.com/alexa>), where users speak their commands to a central device that connects to and controls other devices on a home network. These interaction options were not widely used by the study participants. Although they had experienced using such features, several prohibitive factors were mentioned, including the cost to change existing home technologies, privacy of data, security from hackers, variability in phone interface design, inaccessible phone applications, and reliance on internet services that could be discontinued:

<p>User-Smartphone-Machine Indirect Interaction:</p> 	<p>An area where some accessibility testing work is focused (the design and development of accessible mobile applications) Many manufacturers develop these types of interface without regard for App accessibility Again, when devices are made to be accessible, affording replacements for current devices is a problem for many Additionally, a dependence on the availability of App web services to control home devices, along with privacy concerns, means many were concerned about such systems Many study participants had tried these out, but did not commonly use them</p>
<p>User-Virtual Assistant AI-Machine Indirect Interaction:</p> 	<p>An area where very little accessibility testing work is currently focused Again, when devices are made to be usable through virtual assistants, affording replacements for current devices is a problem for many The same concerns as above were expressed on the availability of web services and privacy Many study participants had tried these out, but did not commonly use them</p>

To these relatively new indirect options we can add a third new form of interaction, that of Remote Support. The steady improvement of wireless internet services has bought the bandwidth capability for high quality video. This has enabled the emergence, in the last few years, paid (e.g., [AIRA](https://aira.io/), <https://aira.io/>) and volunteer remote support services (e.g., [Be My Eyes](https://www.bemyeyes.com/), <https://www.bemyeyes.com/>). The nonvisual user either holds up their smartphone camera, or wears a camera mounted to a pair of glasses, which the remote operator uses to provide the user spoken support. Such services had been experienced by 42% of our participants, but in the focus group discussions it was evident that none of the participants used such services on a regular basis. They cited cost and quality issues as prohibitive factors, in that the paid services are not widespread enough to be easily affordable; and the free/volunteer services are so variable in their quality that it was rarely regarded as a ‘go-to’ solution:

<p>User-Machine Direct Interaction (with sighted remote indirect support):</p> 	<p>Another area where very little accessibility testing work is currently focused</p> <p>Paid 24/7 services with well-trained operators work well</p> <p>Unpaid services with novice operators can be hit-or-miss on quality and availability</p> <p>Many study participants had tried these, but due to cost and quality issues did not commonly use them</p>
---	--

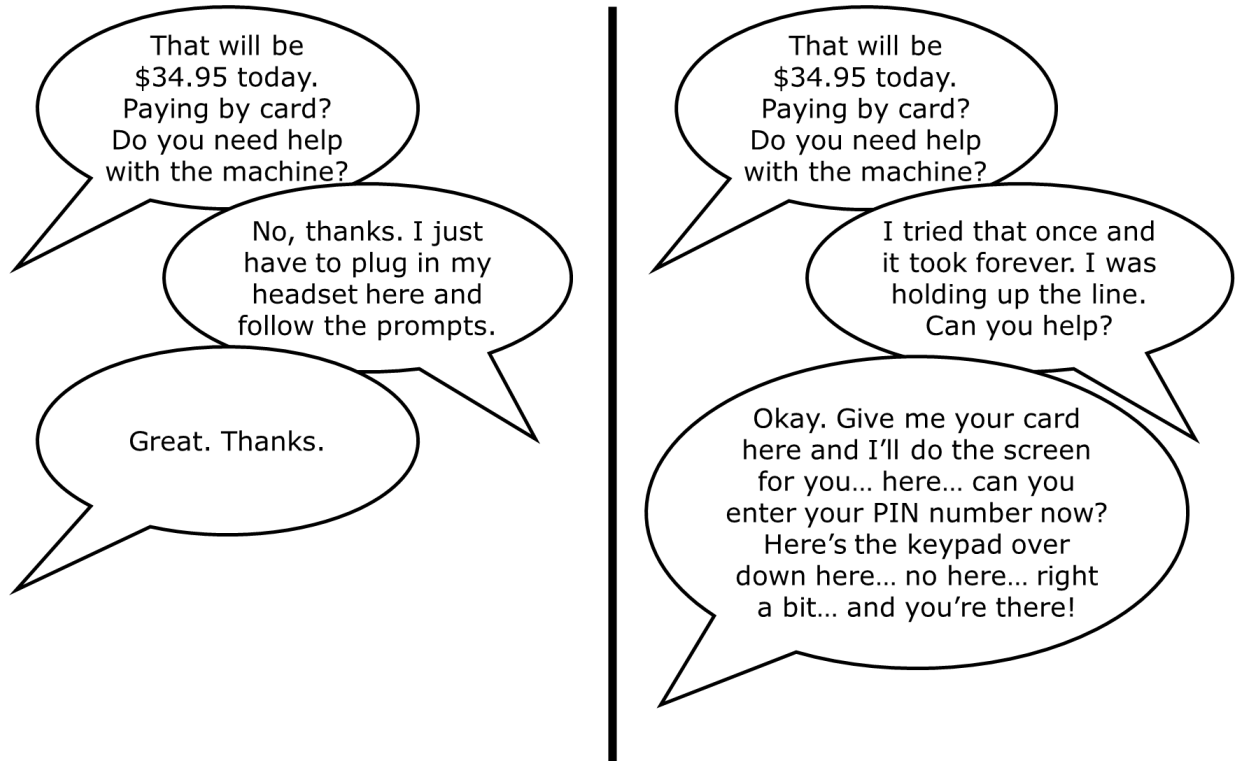
In terms of development and testing, these new forms of indirect and supported interactions have been of interest to those working in the accessibility testing field, but in terms of systematic studies, methodologies and promotion of concepts... we are yet to see any groundswell of activity. When we consider the basic function of interaction, that of using a technology to achieve a task goal, new and emerging forms of interaction such as those cited by our participants remain very low-use when compared to direct interaction using smartphones and PCs.

Sighted support services such as Aira and Be My Eyes have emerged specifically to support the needs of people with visual disabilities. While this has generated interest among the accessibility community, for example at technology accessibility conferences, we have not seen the application of accessibility testing methodologies that take into account the use of remote services as part of the interaction. (Note: Methods for remote accessibility testing of interfaces in traditional forms of direct interaction have been developed and reported on.) Remote support is not limited to accessibility, however. There are industries devoted to remote inspection and testing (in the building industry, petrochemical industry, aviation, and so forth). The 2020 global Covid-19 pandemic has accelerated the adoption of remote inspections and use of remote support.

And, while remote support using technology is of interest, we also noted from our study that the role of other people providing support has also been somewhat overlooked in the accessibility testing field...

Finding #3: ‘Interaction’ testing should also include the role of support provided to users by other people

If we imagine that we are a sighted person in a checkout line at the supermarket, and the customer in front of us in line happens to be blind. We might overhear either of the following conversations between the cashier and the customer.



Over half of the participants (53%) said that they would ask random strangers for sighted assistance when using public devices such as ATMs and Point of Sale devices. This includes help requests for financial transactions with machines that are supposedly ‘accessible’ as required by US laws (Law, 2020.). In the focus groups we learned that this was usually a need for expediency (most common for checkout lines where others were waiting), or as a result of experiences with bad interaction design. For example, one focus group participant told of a fare-vending transit machine that was “accessible” but not functionally usable, in that with the voice guidance it took him over 6 minutes to buy a ticket. This was compared to around 10 seconds for a sighted user to complete the same kind of transaction.

In terms of sources of human support, besides random strangers (reported by 53%), study participants also reported the following when responding to the survey question “If you do seek assistance from sighted people, who do you enlist?”

- Friends and family: 74%
- Customer support staff (in person or by phone or via email or internet chat): 74%
- Remote assistance services (Aira, Be My Eyes, etc.): 42%
- Work Colleagues : 26%
- Social acquaintances: 16%

In a separate question, we asked “Regarding use of public use technologies (Automated Teller Machines, Voting Machines, etc.) which statement do you most agree with?”. Only three of 19 (16%) said “I never have to ask for assistance from sighted people in order to use public technology products and services”. The remaining 16 of 19 (84%) said that they have to ask for sighted assistance:

- Sometimes 6/19
- Often: 6/19
- Almost Always: 4/19

We make note here that in the responses to this question the sighted support could come from people other than random strangers, such as customer support staff, friends or work colleagues. However, we also need to take note of the figures of 84% of participants stating that they require sighted support for (supposedly) accessible devices; and over 20% stating that they need sighted assistance *almost always*.

In terms of interface testing, the line between accessibility and usability has been debated and discussed for a long time (e.g., Henry, 2007, p.109). For example, we might ask whether a device passes accessibility requirements on paper, but fails in a practical usability test in a contextual setting. To our knowledge, usability and accessibility tests of interaction rarely include more than the one user. But, as we see here, for many nonvisual users their practical everyday use of home and public technologies involves additional users. We could (we must!) argue that devices be independently accessible and usable without relying on user vision. That over-arching goal is not in question in this finding. However, we do have to recognize that expanding our definition and consideration of interaction to include other persons involved in the process would be helpful.

In our ‘speech bubble’ example above, when help is offered there is an awkward interaction where the cashier is trying to be helpful when requested: “*Okay. Give me your card here and I’ll do the screen for you... here... can you enter your PIN number now? Here’s the keypad over down here... no here... right a bit... and you’re there!*”. What could be done to improve the design of this type of supported interaction?

Conclusions and Future Work

The context of ‘interaction’ from a developer and usability/accessibility tester perspective, has traditionally been considered as one user and one device (with the device usually being a smart phone application or website, or a PC/tablet application or website). However, we argue that—based on the findings reported in this paper—that the term “interaction”, and therefore what to

test to assess usability/accessibility needs to be broadened. For example, when assessing a technology, we should consider, when appropriate:

- User-machine direct interaction (with or without accessibility features or assistive technologies), *plus*
- User-smartphone-machine indirect interaction
- User-virtual assistant AI-machine indirect interaction
- Interaction with the support of nondisabled persons (either remote or in-person)

The last item should be ideally be considered a last resort, in that the design goal should, of course, be an independently usable device. However, the prevalence of the need for sighted help should not be ignored. It happens in the home, and it happens in public, even with machines that have been attested to be independently usable by people who are visually impaired or blind. The context of real-life use, as is the case at the retail check-out, needs to be considered in order to usability-test and accessibility-test real-life situations. At best, a smooth transaction can take place with assistance; at worst (although rare), people with disabilities are at risk of criminal activity during such transactions (Meehan, 2018).

At the time of writing, the global Covid-19 pandemic has caused a major switch to online work, education, and support (Accessibility Switchboard Project, 2020). This year's activities have resulted in many accessibility testing practitioners focusing on remote conformance and usability testing (as evidenced by the presentation lineup for the 2020 ICT Accessibility Testing Symposium). As we have noted, online support can also form part of *technology interaction*, and should be tested as such. Therefore, with this and the other findings reported herein in mind, we propose that a broader definition of 'interaction' be considered by accessibility testing practitioners.

For our part, we plan to expand our horizons further by developing guidance for the Accessibility Switchboard Information Portal's users:

- One of the needs expressed by our consumer study participants was for guidance on how blind consumers can better utilize the assistance of sighted individuals (either in person or using remote services).
- In our guidance for developers, we have focused on the traditional user-machine direct interaction, and this study has highlighted the need for us to go beyond that: We need to address, for developers and testers, the pros and cons of the newer forms of indirect interaction.
- We also need to address in guidance for developers and testers the ethical and practical considerations of the (unfortunately) common need for blind consumers to seek sighted assistance from friends and family to use household devices, and even complete strangers for help with sensitive financial transactions conducted with public devices.

Acknowledgments

The Accessibility Switchboard (www.accessibilityswitchboard.org) is a project of the National Federation of the Blind (NFB, <https://www.nfb.org/>) with support from the Maryland Department on Disabilities (<http://mdod.maryland.gov/>).

The authors would like to thank the study participants, as well as the members of the [Switchboard Community of Practice](https://accessibilityswitchboard.org/ascoplist) (<https://accessibilityswitchboard.org/ascoplist>) who helped with the planning and review of the website and guidance used in this study.

References

The Accessibility Switchboard Project (2020) Accessibility and keeping employees and students connected from home. March, 2020, Version 1.0. National Federation of the Blind Jernigan Institute. Available: <http://switchboard.nfb.org/>

Henry, S.L. (2007) Just Ask: Integrating Accessibility Throughout Design. Lulu.com

Law, C.M. & Lewis, A. (2016) The Accessibility Switchboard: A new resource and community of practice for tackling accessibility at the organizational level. The ICT Accessibility Testing Symposium: Section 508, WCAG, and beyond, Baltimore, MD, November 2-3, 2016.

Law, C.M. (Ed.) (2020) The Digital Accessibility Legal Digest Volume I: Practice Updates, Guidance and Resources. Available: <https://accessibility.legal/Book.html>

Meehan, S. (2018) Blind Marylanders sue Walmart, saying self-serve checkouts violate ADA. The Baltimore Sun Newspaper (Online), Oct 29, 2018. Available: <https://www.baltimoresun.com/maryland/baltimore-county/bs-md-walmart-blind-lawsuit-20181026-story.html>

Ogami, S. & Renfro, L. (2018) Evaluating hardware accessibility. The ICT Accessibility Testing Symposium: Mobile Testing, 508 Revision, and Beyond, Arlington, VA, November 7-8, 2018.

Copyright notice



The 2020 ICT Accessibility Testing Symposium proceedings are distributed under the Creative Commons license: Attribution-NonCommercial-NoDerivatives 4.0 International (CC BY-NC-ND 4.0, <https://creativecommons.org/licenses/by-nc-nd/4.0/>).

You are free to: Share — copy and redistribute the material in any medium or format.

Under the following terms: Attribution — You must give appropriate credit, provide a link to the license, and indicate if changes were made. You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use. NonCommercial — You may not use the material for commercial purposes. NoDerivatives — If you remix, transform, or build upon the material, you may not distribute the modified material. No additional restrictions — You may not apply legal terms or technological measures that legally restrict others from doing anything the license permits.

Introducing the Section 508 ICT Testing Baseline Alignment Framework

Andrew Nielson

General Services Administration
Office of Government-wide Policy
Government-wide IT Accessibility Program
1800 F St. NW, Washington, D.C. 20006
andrew.nielson@gsa.gov

Michael Horton

General Services Administration
Office of Government-wide Policy
Government-wide IT Accessibility Program
1800 F St. NW, Washington, D.C. 20006
michael.horton@gsa.gov

Katherine Eng

US Access Board
1331 F St NW #1000, Washington, DC 20004
eng@access-board.gov

Keywords: Guidance, Accessibility Testing, ICT Testing Baseline, Standardization, Manual and Automated Testing

Track/Theme: Testing Methods; Other - Creating a Framework to Validate Alignment of Test Processes and Tools

Abstract

Section 508 conformance of Information and Communication Technology (ICT) is currently subjective and dependent on who (or what) is doing the conformance testing. While the standards are defined, interpretation of the standards vary. The Section 508 ICT Testing Baseline Portfolio (aka “the Baseline” or “the Baseline Portfolio”) establishes the minimum requirements of accessibility testing to successfully validate conformance to Section 508. The Baseline Portfolio currently consists only of a testing Baseline for Web but will eventually include other testing Baselines for other ICT (e.g., software, hardware, etc.). The Baseline can be used to create Section 508 test processes as well as determine how well test processes cover Section 508 requirements. In addition to the Baseline document, a supporting framework of test cases, validation processes, change control procedures, and other governance mechanisms (the ICT Testing Baseline Alignment Framework) will assist existing test processes and tools to demonstrate alignment to the Baseline.

Creating a Framework to Validate Section 508 Test Processes

Federal agencies use a wide range of disparate testing methods to validate conformance to IT accessibility requirements. This difference in testing leads to:

- Duplication of effort across agencies,
- Inability to share and rely on validation testing results from other agencies, and
- Confusion among IT product vendors when test results from agencies conflict with one another.

A baseline for test requirements will allow agencies to share a common interpretation of Section 508 requirements. To properly implement the Baseline, a common framework of test cases, validation processes, change control procedures, and other governance mechanisms is proposed below.

ICT Testing Baselines for Revised Section 508

In January of 2017, the US Access Board released the final rule to revise the original Section 508 standards, which had first become effective in 2001. The Revised Section 508 Standards reference the World Wide Web Consortium's (W3C) Web Content Accessibility Guidelines (WCAG) 2.0, Level A and AA Success Criteria and conformance requirements as the "new" Section 508 standards for Web content. To update the [DHS Trusted Tester test process](#) for the new requirements, the ICT Testing Baseline for Web was developed as the foundation to build the test process and establish accuracy of utilized testing tools. After completing the Trusted Tester process and training, the ICT Testing Baseline Working Group (ITBWG) resumed efforts to finalize the Baseline for Web and continue promoting the Baseline Portfolio as a unified foundation for conformance test processes to improve Section 508 testing across government.

The [ICT Testing Baseline for Web](#), which specifies what web content must be evaluated for Section 508 conformance, was written to create a comprehensive test process. However, federal agencies that have a Section 508 test process can also use the Baseline for Web to determine how well their test process accurately tests for all Section 508 requirements.

The ICT Testing Baseline for Web is the first of many baselines to be developed to aid in evaluating Section 508 conformance of ICT. Plans for additional testing baselines include ones for software, hardware, mobile, and other types of ICT that have applicable Section 508 requirements.

The ICT Testing Baseline for Web is a compilation of testing requirements specific to web content. With the ICT Testing Baseline for Web completed at the end of September 2020, additional components are in development to support how agencies can demonstrate baseline alignment of their test processes. In particular, the [ICT Testing Baseline Alignment Framework](#) provides a framework and guidance for test process owners to evaluate whether their process aligns with the ICT Testing Baseline Portfolio. With agency test processes aligned to the Baseline Portfolio, agencies can trust each other's test results and reduce redundant accessibility

testing efforts. A consistent test foundation among agencies will provide clearer conformance results for IT vendors as well.

Elements of the ICT Testing Baseline Alignment Framework

The ICT Testing Baseline for Web has garnered some interest from federal agencies, industry, and academia, and various parties have made efforts to align their testing tools and test processes to the Baseline for Web. Such efforts support its primary purpose: to provide a consistent and accurate test foundation for accessibility test processes. However, current claims of alignment unfortunately lack a method to validate those claims. The ITBWG, therefore, created the ICT Testing Baseline Alignment Framework to provide test process owners the ability to verify and document alignment. It also provides federal agencies and other parties interested in adopting a particular accessibility test process the ability to evaluate test process alignment prior to adoption. The proposed ICT Testing Baseline Framework consists of test cases, validation processes, change control procedures, and other governance mechanisms.

The predominant component of the ICT Testing Baseline Alignment Framework consists of a number of test cases, each of which includes a test file (or code sample) and the steps that must be completed to verify that a test process meets the requirements defined in the ICT Testing Baseline for Web. Each test case describes the expected result for each code sample if a test process actually follows the Web Baseline requirements. The idea behind the test cases is that a test process produces the same result that the ICT Testing Baseline for Web would predict for each code sample, thus validating alignment.

The Test Cases are supported by a validation process describing,

- Who is actually involved in validating alignment,
- What information is necessary to validate,
- What steps are necessary, and
- An overall governance process to review and adjudicate proposed changes to elements of the both the ICT Testing Baseline and the Alignment Framework.

These elements are further described in the following three sections.

Note: The ITBWG has not yet finalized the ICT Testing Baseline Alignment Framework. Until it is finalized and approved, any or all of the components of the Alignment Framework are subject to change.

Test Cases for Technical Verification of Alignment

The ICT Baseline Working Group is still in the process of compiling code samples and test cases to facilitate validation of alignment to the ICT Baseline for Web. The test cases are available for review in the [Baseline Alignment Framework repository](#).

The test cases present the ICT Testing Baseline tests as the minimum requirements that an accessibility testing process must satisfy in order to validate Section 508 conformance. It also

lays out the steps necessary to verify that the test process in question meets the baseline requirements. Each test case is organized with the following content sections:

- **Test Case Summary:** provides test case identifier information, a general description of the test case and its purpose, and the applicable ICT Testing Baseline requirements
- **Test Case Detail:**
 - **Expected Baseline Result:** for the code in question, designates the expected test result and describes the reasoning and justification for the result
 - **Test Data:** provides the actual code to be tested for accessibility
 - **Test Steps:** details the steps necessary to verify that a test process meets the ICT Testing Baseline requirements

When documenting accessibility test process alignment with the ICT Testing Baseline for Web, process owners should compare their actual results against the Expected Baseline Result for each test step. The Baseline Alignment Framework repository will provide a standard report template for test process owners and stakeholders to document alignment as they progress through the test cases.

Alignment Validation Process

As the ICT Testing Baseline Alignment Framework and all of its components are available in a GitHub repository and as a published website, anyone may use it to evaluate whether a particular test process aligns with the ICT Testing Baseline for Web. The Alignment Framework also provides some resources to help facilitate evaluation including:

- **ICT Testing Baseline - Test Case Traceability Matrix** -- the traceability matrix maps each test case to a specific step or set of steps in the ICT Testing Baseline
- **Alignment Validation Report Template** -- the alignment validation report provides a format for documenting results against each test case
- **Test cases in JavaScript Object Notation (JSON)** -- the JSON file provides information about the test cases that could be particularly useful for evaluating automated testing tools; it includes a URL for each test file as well as test case alignment to WCAG Success Criteria and ICT Testing Baseline requirements

While anyone can conduct their own evaluation of alignment using the above resources, an official determination of alignment requires independent validation. The ITBWG is still in the process of evaluating what resources and/or methods the government will employ to conduct its independent validation of alignment claims. Options could include validation by a federal employee or representative, validation by a third-party accrediting authority, self-assessment and attestation by the process owner, or some combination of all three. The validation method may also evolve over time, based on interest and the evolving needs of the government. Each option could imply both cost and resource commitments to the government, to process owners, or to process consumers. The ITBWG is carefully considering such implications to evaluate the most cost effective and efficient solution while ensuring the impartiality and integrity of the process.

Regardless of what type of independent validation method the ITBWG elects to use, the overall validation process will include the following steps:

1. Use the test cases to evaluate your test process and document your results using the Alignment Validation Report Template
2. Submit an independent validation request to the ITBWG
 - Provide self-assessment of alignment (using the Alignment Validation Report Template)
 - Provide your test process documentation and access to tools employed by the test process; Communication with the independent tester will not be available during evaluation, so submitters must ensure documentation is sufficient for execution of the test process
3. Independent Verification & Validation - who will actually conduct is still TBD (government representative or 3rd-party accreditation authority)
 - IV&V will evaluate request and self-assessment; then use tool and process to validate alignment against each test case
 - Dispute resolution process will be created and enforced for contested results

Proposed Governance and Change Control

While test cases are still pending finalization for the Baseline Alignment Framework, the final, revised version of the ICT Testing Baseline for Web was released at the end of September 2020. Any future revision past the initial release, including changes to the ICT Baseline itself, to individual Baseline Alignment Framework test cases, to the alignment validation process, and even simple edits to fix typos will require ongoing governance and change control. For each change, the government must consider possible implications for stakeholders, including costs related to government oversight and/or test process owners for potential re-validation of test process alignment. For any future revision, the government will provide notification and description of any change(s).

Baseline Governance Board Membership

For the purposes of governance and change control, the ITBWG has proposed the establishment of the Baseline Governance Board (BGB). The BGB will consist of the following roles and membership:

Role	Membership
BGB Authorizing Officials	<ul style="list-style-type: none"> • US Access Board • General Services Administration
BGB Technical Advisory Representatives	<ul style="list-style-type: none"> • US Access Board • General Services Administration • Department of Homeland Security • Office of Personnel Management

Baseline Governance Board Roles and Responsibilities

The table below outlines the responsibilities generally assumed for each BGB role.

Role	Responsibilities
BGB Authorizing Officials	<ul style="list-style-type: none"> • Designate Technical Advisory Representatives with the appropriate technical expertise to support and inform decision-making • Review and adjudicate proposed ICT Testing Baseline and/or Baseline Alignment Framework changes • Evaluate impact of proposed changes on stakeholders, including implications for previously certified processes/tools; ensure changes are impartial • Provide notification of changes and determine any requirements and grace period allowances for re-alignment and/or re-validation
BGB Technical Advisory Representatives	<ul style="list-style-type: none"> • Review input and feedback on the ICT Testing Baseline and the Baseline Alignment Framework; determine appropriate methods to address or incorporate feedback • Monitor technology trends (e.g., code specification changes, accessibility API updates, testing advancements) that may affect or influence the ICT Testing Baseline and/or the Baseline Alignment Framework • Propose changes to the ICT Testing Baseline and/or the Baseline Alignment Framework to address identified issues and/or feedback from stakeholders • Advise BGB Authorizing Officials with regard to impact of proposed changes, re-alignment and re-validation requirements, and notification to stakeholders

Feedback and Contributions

Like the [Section 508 ICT Testing Baseline for Web](#), the [ICT Baseline Alignment Framework](#) is hosted on GitHub as a platform to facilitate feedback and contributions to the alignment framework and test cases.

Provide Feedback

The ICT Testing Baseline Working Group (ITBWG) welcomes feedback from government and industry web development and accessibility testing communities to ensure the maintenance of the Baseline in accordance with established web-based technologies. Individuals interested in providing feedback should contact the ITBWG by:

- Sending an email to ictbaseline@gsa.gov, with feedback which:
 - Includes your full name, email address, phone number, and agency or business
 - Specifies the baseline test(s) being discussed in your feedback
 - Sufficiently describes the issue(s), with any recommendation(s) and reference(s)
 - Creating an “issue” in the GitHub repository (for the ICT Testing Baseline or for the Baseline Alignment Framework)
- Creating your own fork(s) of the ICT Testing Baseline or Baseline Alignment Framework GitHub repository and submit pull requests to propose changes directly

Become a Contributor

Participation in the ICT Testing Baseline Working Group (ITBWG) is open to government employees and representatives who specialize in the development and testing of web-based products. As a member of this all-volunteer governing body, individuals will support the continued development, maintenance and promotion of the Baseline.

Reflecting the same set of skills and expertise necessary for implementing Section 508 across an agency’s IT portfolio, the ITBWG welcomes the support of contributors in various areas of focus including, but not limited to:

- Test case developers
- Requirements officials
- Change management officials
- Technical writing
- Administrative support
- Section 508 SMEs
- IT testing specialists

Eligible individuals interested in becoming a contributor should contact the ITBWG at ictbaseline@gsa.gov.

Framework Rollout

Rollout Milestones

The following table represents a high-level timeline of key milestones of the Baseline rollout.

Milestone	Timeline
Finalization of the Section 508 ICT Testing Baseline	Phase 1
Formal adoption of the ICT Baseline by OMB and/or the Federal Chief Information Officers' Council	Phase 1
Designation and allocation of resources to support ongoing governance	Phase 2
Start date for accepting recommendations and proposals for validation of test processes and tools from federal agencies	Phase 2
Finalization of the ICT Baseline Alignment Validation process and test cases	Phase 2
Anticipated turnaround time for alignment validation of individual processes and tools	Phase 3
Pilot the Validation process with two-three federal agencies	Phase 3

What Organizations Can Do to Prepare for and Adopt the ICT Baseline

While practical adoption of the Baseline will necessitate use of the validation framework, agencies may be interested in an initial review of their products for alignment to the Baseline. Organizations may find some benefit to reviewing the Baseline in order to gain some insight into any possible adjustments that may be necessary for successful validation. Such a preliminary review might include:

- Evaluating existing test processes for possible adoption and implementation in an organization's system development processes
- Evaluating testing tools for possible incorporation in test processes
- Modifying existing processes to align to the ICT Baseline
- Combining automated and manual testing processes in system development processes

Copyright notice



The 2020 ICT Accessibility Testing Symposium proceedings are distributed under the Creative Commons license: Attribution-NonCommercial-NoDerivatives 4.0 International (CC BY-NC-ND 4.0, <https://creativecommons.org/licenses/by-nc-nd/4.0/>).

You are free to: Share — copy and redistribute the material in any medium or format.

Under the following terms: Attribution — You must give appropriate credit, provide a link to the license, and indicate if changes were made. You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use. NonCommercial — You may not use the material for commercial purposes. No Derivatives — If you remix, transform, or build upon the material, you may not distribute the modified material. No additional restrictions — You may not apply legal terms or technological measures that legally restrict others from doing anything the license permits.

Leveraging Trusted Tester for Web with Test Automation

Ann Marie Davis

New Editions Consulting, Inc, supporting
U.S. Department of Homeland Security
Office of Accessible Systems and
Technology
annmarie.davis@associates.hq.dhs.gov

Halima Diallo

New Editions Consulting, Inc, supporting
U.S. Department of Homeland Security
Office of Accessible Systems and
Technology
halimatou.diallo@associates.hq.dhs.gov

Alan King

New Editions Consulting, Inc, supporting
U.S. Department of Homeland Security
Office of Accessible Systems and
Technology
alan.king@associates.hq.dhs.gov

Kristen Smith-O'Connor

New Editions Consulting, Inc
103 W. Broad St., Suite 400, Falls Church,
VA 22046
ksmith@neweditions.net

Dominique Wheeler

New Editions Consulting, Inc, supporting
U.S. Department of Homeland Security
Office of Accessible Systems and
Technology
dominique.wheeler@associates.hq.dhs.gov

Vince Prentice

New Editions Consulting, Inc, supporting
U.S. Department of Homeland Security
Office of Accessible Systems and
Technology
vincent.prentice@associates.hq.dhs.gov

Keywords: Manual testing, Trusted Tester, Automated Testing, Section 508.

Track/Theme: Automated Testing, Manual Testing Tools, WCAG 2.0, 2.1, and beyond, Section 508 Refresh / Revision, Provision of training to testers

Abstract

The Trusted Tester (TT) process for Web continues to evolve. Efforts have been made to streamline and clarify the test process for manual testing, improve the student training experience in the TT process, and meet the growing demands of test automation by incorporating viable test tools to support the integration of TT accessibility into the test automation process. Historically, the TT program's overall mission was to create a manual, accurate, and repeatable Section 508 Web testing process that could be implemented by any group wanting to perform accessibility testing and remediation. New opportunities in the TT process focus our attention on key improvements that multilaterally provide guidance to students through a series of instructional resources, clarify and simplify the test process, and introduce a vetted ruleset process that consists of automated tool test results when using the manual Test Process for Web v5.0 (TTv5.0) criteria.

Trusted Tester

Overview of Trusted Tester Training Program

The TT process was launched by the Department of Homeland Security (DHS) Office of Accessible Systems and Technology (OAST) in 2008 to create a reliable, repeatable, and verifiable Section 508 web testing process that could be used by any group wanting to implement accessibility testing and remediation. The goal was to create a testing process that would educate testers and validate their understanding and proficiency of Section 508 conformance testing using free accessibility testing tools. The TT program slowly evolved to meet the growing demands of numerous Developers, Quality Assurance Analyst, Independent Verification & Validation (IV&V) Testers, and accessibility enthusiasts, while aligning with the revised Section 508 Refresh Standards. The TT Training Program, launched in 2010, has experienced tremendous changes since its inception. These changes provide opportunities for growth, improvement, innovation, and expansion to incorporate test automation.

The TT Training Program

The Trusted Tester Web course is a free online certification available to anyone who wants to learn accessibility testing. Students are trained to apply the DHS TT Section 508 Conformance Test Process for Web v5.0 (TTv5.0) using the approved testing tools. The test process uses a manual test approach that aligns to the Information and Communication Technology (ICT) Testing Baseline and relies on understanding WCAG 2.0 and Section 508. Although there are no pre-requisites to enroll in the program, it is recommended that individuals have a basic understanding of HTML, CSS, JavaScript, and ARIA.

The TT Web Course is an enrollment-based, self-paced online training course that provides detailed information and instructions, including practice exercises, to prepare students in testing and evaluating ICT products pursuant to WCAG 2.0 (Level A and AA), ICT Baseline, and the revised Section 508 Standards. The course provides:

- Test tools, methods and references;
- Course-specific Knowledge Checks;
- Identifying Content for each WCAG Success Criterion;
- ‘How to Test’ and ‘Evaluation’ procedures for each WCAG Success Criterion;
- PASS, FAIL and Does Not Apply examples of each course topic;
- Support through Questions and Answers (Q&A) Boards and Frequently Asked Questions (FAQs) where students can pose questions and get responses from instructors or review the FAQs to see if their question(s) have been already addressed;
- Summary of exam results; and
- Online and printable course resources to aid students as they go through the exams, such as the *Trusted Tester Process for Web* document and the *AT and Service-Based User Guide*.

The DHS TT Training Program is recognized within the federal government and across multiple industries as an established Section 508 conformance testing process. To become certified, individuals must demonstrate their proficiency in using the test process to evaluate web content for Section 508 conformance, accurately interpret test results, and pass the DHS Section 508 TT Certification Exam. Since v5.0 of the test process was launched in August 2019, over 1,000 testers have been certified. Once a TT Web certification is granted, individuals possess the required skill-sets to perform code-based inspection of ICT web products and are considered to be subject-matter-experts in understanding and interpreting web accessibility. Having gone through the TT Training Program and acquiring a TT web certification, individuals are able to:

- Interpret test results produce by manual and automated accessibility testing methodologies;
- Provide guidance to System Owners, IT Program Managers (PM) and Developers in planning and design phase in the software development life-cycle;
- Help developers with remediation efforts;
- Create and generate testing results using the Accessibility Compliance Reporting Tool (ACRT); and
- Possess the needed skills to review vendor Voluntary Product Accessibility Template (VPAT)/ Accessibility Compliance Report (ACR) in determining ICT product accessibility.

Improvements to the Trusted Tester Course

Over the last year, several key improvements have been made to the TT Web certification program, including:

- Adding guidance for incorrect test results in the Incremental Exams;
- Incorporating instructional testing videos on how to test using the Social Security Administration’s Accessibility Name and Description Inspector (ANDI) testing tool; and
- Providing Lesson Plans to aid students preparing to retake the certification exam.

The newest development to the TT course is a series of online “How to test using ANDI” tutorial videos. These videos will be embedded within specific course topics to highlight the How to Test steps in the Test Process. They provide a visual demonstration of the test process for each lesson to better accommodate different learning styles.

The tutorial videos are paired with the course materials to increase students understanding and retention levels, as well as, provide a method for students to be more engaged with the course material. The videos:

- Provide a breakdown of complex course content, using animation and simple visuals;
- Are available for students to watch or re-watch at any time;
- Assist students in understanding the extensive, technical text material; and
- Accommodate different learning styles and provide a more varied learning experience.

Proposed Updates to the Trusted Tester Process

Minor revisions are being proposed for version 5.1 of the DHS Trusted Tester Section 508 Conformance Test Process for Web. Changes are reviewed by the TT Process Working Group which includes members from the Federal Access Board, General Services Administration, and DHS OAST. This group has proposed changes to clarify how to apply the test process and to simplify testing steps. This section describes some of the changes being considered.

Keyboard Testing

Testing version 5.0 instructed users to identify interactive elements using the mouse. Recognizing there are other ways to identify interactive elements, version 5.1 also provides for the use of touch screen, voice commands, and other documented functionalities (such as shortcut keys). It further clarifies that shortcut keys need to be discoverable by users, typically on the page in Help documentation.

Keyboard testing was performed in eight separate test steps, but version 5.1 eliminates separate test results for 4.G, 2.4.3-focus-order-reveal and 4.H, 2.4.3-focus-order-return. Test 4.G has been combined with testing for 4.F, 2.4.3-focus-order-meaning by adding a step “to activate trigger controls that reveal hidden content (e.g., menus, dialogs, modal dialogs, expandable tree list) to check the focus order to, from, and within the revealed content.” This change made the test for 4.H, 2.4.3-focus-order-return redundant.

Instructions related to backwards focus order were added to 4.F, 2.4.3-focus-order-meaning in version 5.1.

- Keyboard focus is expected to remain in a modal dialog box until it is closed. Version 5.1 clarifies this includes both forward and backward focus.
- A new test step clarifies “Backward focus order does not have to mirror the forward focus order. However, it must preserve the meaning and operability of the page.”

Content Structure

Headings provide important organizational and navigational structure to web page content. To achieve this, they need to be visually apparent and properly identified programmatically. Version 5.1 of the test process clarifies that the visual and programmatic heading must match. New step 3 in test 10.B, 1.3.1-heading-determinable directs testers to: “Review the ANDI Output for each heading to determine if it matches the visual heading. If they do not match, then the heading is not properly defined programmatically.”

Non-Interference of Non-Conforming Content

Sometimes developers determine that an alternate version of content is needed. Tests are performed on the “accessible” version such as ensuring it has the same functionality as the non-conforming version. Tests performed on the content of non-conforming version must meet Conformance Requirement 5 from Section 508 to ensure no interference is introduced for users of the accessible version. For example, a keyboard trap in the non-conforming version may pose an accessibility problem for users. Test 1.E for non-interference requires these key tests on the non-conforming version either PASS or DO NOT APPLY.

Flashing content poses a challenge to testers with photosensitive epilepsy. The TT result for flashing content is NOT TESTED. Version 5.1 has a new note that clarifies a test result of NOT TESTED for 3.A, 2.3.1-flashing does not meet test condition for 1.E, non-interference.

Buttons and Links

Testing for Buttons was moved from the Links section to Form Elements. By clarifying that form elements include buttons, version 5.0 test 6.B, 4.1.2-button-purpose was removed. The forms tests verbiage was modified from form “fields” to form “components” or form “controls.” Test 6.C, 4.1.2-change-notify-links is now 6.B, 4.1.2-change-notify-links.

Responding to Baseline Updates

The Baseline Working Group combined Baseline 2 for Focus Visible and Baseline 3 for Focus Order into a new Baseline 2 for Focus. To avoid renumbering the entire baseline and minimize the impact to ongoing testing updates, Baseline 25 for Non-Interference is now Baseline 3 for Non-Interference. Version 5.1 of the Accessibility Test Process reflects these changes.

Other Small Clarifications

Other edits were made to the test document to provide clarity:

- The identifying content for 2.A, 1.4.2-audio-control as “Automatically moves, blinks, or scrolls without user activation” now contains the text “(including scrolling text, videos, and multimedia)”.
- An inaccurate reference to “moving/blinking/scrolling” content in the Evaluation Results for 2.C, 2.2.2-auto-updating was updated to correctly reference “auto-updating” content.

Other items under consideration

Version 5.1 of the test process is still in draft form and further changes are still in discussion. Among other changes, the TT Process Working Group is considering whether:

- it is considered a browser setting that is not tested if color alone indicates a link has been visited, or should be failed for 13.A, 1.4.1-color-meaning.
- contrast requirements apply to the alternate text version of a CAPTCHA images, color changes to visited links, selected menu options, etc.

Test Automation

When testing for accessibility conformance, it is generally accepted that manual testing methods like TTv5.0 will produce a more complete and accurate result. Many WCAG success criteria simply cannot be reliably tested using test automation alone. Nonetheless, there is still strategic value in integrating test automation techniques. A better decision is made by considering the “big picture” rather than being limited solely to the “manual” vs. “automated” test result question.

Enhancing Overall Software Development and Delivery

Business and government are constantly looking for ways to build a better mouse trap in terms of developing and delivering ICT. In today's technology, a force to be reckoned with in terms of a better mouse trap is a phenomenon called DevOps.

DevOps, aligning the goals of Development and Operations, is collectively a combination of tools, techniques, and technical culture enhancements that bring value to business and customers alike in the improved creation and delivery of ICT. Although DevOps has several components, automation is the key underlying principle to realizing its promise.

Therefore, the resulting challenge to accessibility testing, historically a manual process, is how to best participate in this new paradigm to retain the integrity of the WCAG Success Criteria while not appearing as a hindrance to the enhanced life cycle processes of today's project teams.

Selecting Accessibility Test Automation Tools

There are viable test automation tools – both licensed and open source – that will support the integration of accessibility success criteria into the test automation process at various levels.

In terms of minimal criteria, tools should be chosen that:

1. Integrate well into the organization's technical environment;
2. Provide technically accurate test results; and
3. Allow the tester to customize or choose the pass/fail criteria (rulesets) used for testing.

The Importance of Vetting Rulesets

Vetting the underlying pass/fail criteria of the automated tool (AKA rulesets) against TTV5.0 criteria is critical to having that tool's test results align favorably with those criteria.

Consequently, choosing a test automation tool that allows the tester to choose the individual rules used for testing is important to tool choice when the goal is to align as closely as possible with TTV5.0.

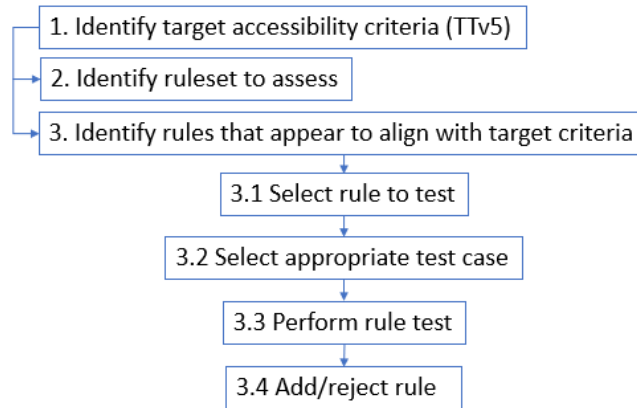


Figure 1-Ruleset Vetting Process Diagram

The ruleset vetting process consists of selecting rules most likely to align, based on their descriptions, and then testing those rules against relevant test case examples to demonstrate the degree to which the automated tool’s test results agree with test results achieved when using the manual TTV5.0 criteria against the same test case.

Example of Ruleset Vetting

Automated Rulesets

The following example examines two different fictitious automated rules (rule imgLabel and rule imgH37) from two different vendors (vendor “x” and vendor “y”) who both describe the functionality of their rules as follows:

Image elements must have alternate text. This rule looks for implementations where image elements do not have programmatically associated labels.

Example Test Cases

To confirm the logic in the automated rule, it should be tested against one or more relevant test cases. Test cases should be constructed to expose the degree to which a rule agrees with the testing outcome of a specific TTV5.0 Test ID regarding what would fail as well as what would pass for conformance. Below are examples of “passing” and “failing” scenarios for the following TTV5.0 Test ID:

- 7.A: The accessible name and accessible description for a meaningful image provides an equivalent description of the image:

Failing test case: The following code snippet is a test case that, under TTV5.0, will result in a failing result (missing equivalent text alternative via accessible name or description):

```

```

Passing test case: The following code snippet is a test case that, under TTV5.0, will result in a passing result (proper equivalent text alternative provided by the aria-label attribute):

```

```

Ruleset Vetting Result

The vetting process results are as follows:

Rule `ImgLabel` (vendor “x”) was a match for both the failing and the passing test case. The rule logic identified the failing test case as having a missing equivalent text alternative, as well as identifying the passing case as having a proper equivalent text alternative via the `aria-label` attribute. Therefore, the decision would be to include this rule in automated testing.

Rule `ImgH37` (vendor “y”) was a match for the failing case, but was not a match for the passing test case. Specifically, the rule logic for `ImgH37` correctly identified the missing equivalent text alternative for the failing case, but the logic in this rule only accepts the `alt` attribute as a proper way to implement an equivalent text alternative for an `` element and considers any other remedy, such as `aria`, as failing compliance. This logic conflicts with TTV5.0 methodology. The high chance of false negatives that could result from this logic conflict suggests that this rule should be rejected in favor of the `ImgLabel` rule to align more closely with TTV5.0.

Examples of Accessibility Test Automation Integration

Having established a “preferred” list of rules against which to test, testers can then integrate this custom accessibility criteria into various levels of automated developer unit testing that include but are not limited to the following integration types and example tools:

- Command Line Interface (CLI) testing
 - [Axe-core](#)
 - [Google Lighthouse](#)
 - [Pally](#)
- Automated Scripting
 - [Axe-core](#)
 - [Pally](#)
- Testing Framework Integration
 - [Create React App](#) / [Jest](#) / [Jest-axe](#)
- Continuous Integration/Continuous Delivery (CI/CD) Pipeline
 - [Jenkins test automation server](#)

Ensuring the Full Scope of Accessibility Testing

To ensure the full scope of applicable test criteria are properly tested, manual testing must be performed on any Section 508 success criteria not fully testable by automation alone.



The 2020 ICT Accessibility Testing Symposium proceedings are distributed under the Creative Commons license: Attribution-NonCommercial-NoDerivatives 4.0 International (CC BY-NC-ND 4.0, <https://creativecommons.org/licenses/by-nc-nd/4.0/>).

You are free to: Share — copy and redistribute the material in any medium or format.

Under the following terms: Attribution — You must give appropriate credit, provide a link to the license, and indicate if changes were made. You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use. NonCommercial — You may not use the material for commercial purposes. NoDerivatives — If you remix, transform, or build upon the material, you may not distribute the modified material. No additional restrictions — You may not apply legal terms or technological measures that legally restrict others from doing anything the license permits.

Creating an Evaluation Checklist for Microsoft Office Users

Jonathan Whiting

WebAIM
6807 Old Main Hill
Logan UT, 84341
United States
jonathan.whiting@usu.edu

Keywords: Word, PowerPoint, Accessibility Checker

Track/Theme: Testing Approaches

Abstract

Microsoft Office contains a capable Accessibility Checker. However, like all automated accessibility evaluation tools, it offers an incomplete accessibility picture. Manual evaluation is still a necessary part of a complete accessibility evaluation. WebAIM has created a checklist that combines the automated reporting of the Microsoft Accessibility checker with manual testing required for a complete evaluation.

Introduction

Microsoft Office—especially Word—may be the starting place of more web content than any other tool, making it vital that content creators not only understand how to create accessible content, but also how to evaluate already created content. For years, Microsoft has included a built-in accessibility checker. This tool continues to improve with time, but it shares the same accessibility limitations as any other automated accessibility checker—it does not identify all elements.

This is something that Microsoft Acknowledges (Rules for the, n.d.). However, their documentation only outlines two Accessibility Checker limitations—information conveyed using color alone and videos that have open captions or do not need captions. It fails to outline potential issues that must be evaluated manually including:

- Text that serves as a heading in Word that is not given appropriate heading Style (Heading 1, Heading 2, etc.).
- Text that is styled as a heading in Word that does not function as a heading (i.e., it does not describe a section of content that follows).
- Images with alternative text that is not equivalent.
- Tables with row headers that are not identified correctly.

Alternative text is a good example of an accessibility technique where the Word Accessibility checker can make evaluation easier (especially with newer versions), but where manual evaluation is still necessary.

There are several possibilities for the alternative text of an image in an Office document: no alternative text, the image filename as alternative text (which was once a bug in Microsoft Office), the correct alternative text, and incorrect alternative text. Every version of the accessibility checker will identify images that do not have any alternative text. However, only the checker in Office 365 will flag alternative text that it thinks is an image filename.

Office 365 introduces two additional options: “Mark as decorative,” and “Generate a description for me.” “Mark as decorative” is an excellent and long overdue feature, but there is also the possibility that an image is marked as decorative even when it conveys content (possibly to get rid of an error in the accessibility checker). If an image is given alternative text using “Generate a description for me” (not recommended), the checker will provide a unique message: “Intelligent Services: Suggested alternative text.” However, neither of these new Office 365 features will be identified by the accessibility checker in Office 2016.

Of course, manual evaluation is also necessary to determine if the alternative text of a given image is equivalent.

WebAIM has created a checklist that combines the automated reporting of the Microsoft Accessibility Checker with manual testing required for a complete evaluation (Whiting, 2019).

Requirements for an Office Accessibility Checklist

Four design requirements guided the creation of the checklist:

1. A single version of the resource covers Office 2016 and 365.
2. The language is approachable for content creators with basic accessibility knowledge.
3. It is concise so that more people will want to use it.
4. It provides guidance for repair.

A Single Version for Office 2016 and 365

Currently, there is a single checklist for Office 2016 and 365 on Windows and Mac. Although this resource is available freely on the WebAIM site, it was designed to be a handout during live training events where participants share the same documents and resources but are often using different versions of Office and/or operating systems.

Creating and maintaining a single evaluation resource has sometimes been a challenge. The primary reason for this is the fact that there are many users of older versions of Office, especially Office 2016. There are several differences between Office for Mac and Windows, and between Office 2016 and 365. Much of this will be resolved in the future as people continue to move to subscription-based Office 365. There is much greater parity between operating systems (but the Windows and Mac interfaces are still not the same). Additionally, new accessibility features are typically rolled out on Office for Windows first, and it is sometimes months before the same features are available on Mac.

Approachable Language

While evaluators using this resource are surely knowledgeable in their respective areas, it is fair to assume that many, or even most, are not familiar with the more technical aspects of electronic document accessibility or the structure and terminology of WCAG 2. On the other hand, a person seeking out this resource will likely be familiar with basic accessibility principles and terminology, such as “alternative text” and “table header”, and if not, there are links to introductory Word and PowerPoint articles at the start of the resource.

Concise

Always present during the development of this checklist was the prediction that people were more likely to use the checklist if the length did not make it seem overwhelming, and that more people creating substantially accessible content would have a greater impact than fewer people creating documents that were WCAG 2.0 or 2.1 Level AA conformant.

Notably, Microsoft does not claim that use of their Accessibility Checker ensures WCAG-conformant documents (Improve accessibility with, n.d.).

With only three exceptions, all Errors, Warnings, and Alerts available in Microsoft’s Accessibility Checker are included in the checklist. The excluded three results—“Default section name” (Error), “Duplicate section name” (Tip), and “No image watermarks are used” (Tip, removed in Office 365)—were omitted because they were seen as adding more overhead than benefit.

A very small number of manual checks were omitted for the same reason. For example, Office permits a custom color for visited links. While it is possible for a document to contain a custom link color that does not meet WCAG 2.0 contrast requirements, checking for this uncommon scenario would be time consuming and may make it less likely that a checklist like this would be used regularly.

Guidance for Repair

For many websites, accessibility evaluation is a standalone process, with repair to follow. However, in an Office document, it is more natural to repair any issues while the document is being evaluated. For this reason, this checklist includes instructions for repair.

Checklist Excerpt: Tables

Below is an excerpt from the checklist related to tables. It demonstrates how the checklist combines automatically-detected issues (e.g., “No header row”) with steps for manual review (e.g., look for row headers), and even points out a place where a Warning within the Accessibility Checker—“Merged or split cells in a table”—may not always indicate an accessibility issue.

Tables

Principle	Review	Repair
The first row in a table contains correctly-defined column headers.	Error: No header row	Table Design tab > Check Header Row checkbox
When the first column of a table should contain headers, they are correctly identified as row headers.	Manual: Click within the Table Design tab > make sure the First Column checkboxes match the table header structure.	Check or uncheck the "First Column" checkbox to match the table header structure.
Table has a simple structure, avoiding merged cells or split cells.	Warning: Merged or split cells in table Use the Tab key to ensure the order of the cells in the table is logical.	Table Layout tab > Merge Cells or Split Cells
Tables should not use blank columns or rows for visual formatting.	Warning: Tables don't use blank cells for formatting (not in all versions, see Note) Manual: Visual Inspection	Right Click empty column or row > Delete Cells
Avoid tables that are used for layout when possible. If used, ensure table reading order is logical.	Warning: Check reading order (for tables with no borders or styles) – Use the Tab key to navigate the cells and ensure reading order is correct.	- If the content can be removed from the table: Click in the table > Table Layout tab > Convert to Text . -If this isn't possible, use the Tab key to navigate the cells and ensure reading order is correct.

Note: Empty table cells are not always an accessibility problem. For example, the top-left cell is often blank in a table.

References

Microsoft. (n.d.) Improve accessibility with the Accessibility Checker. Retrieved on September 4, 2020 from <https://support.microsoft.com/en-us/office/improve-accessibility-with-the-accessibility-checker-a16f6de0-2f39-4a2b-8bd8-5ad801426c7f>

Microsoft. (n.d.) *Rules for the Accessibility Checker*. Retrieved September 4, 2020, from <https://support.microsoft.com/en-us/office/rules-for-the-accessibility-checker-651e08f2-0fc3-4e10-aaca-74b4a67101c1>

W3C World Wide Web Consortium (2018, June 5). *Web Content Accessibility Guidelines 2.1*. <https://www.w3.org/TR/2018/REC-WCAG21-20180605/>

Whiting, J (2019, January 30). *Word and PowerPoint Accessibility Evaluation Checklist*. WebAIM. <https://webaim.org/resources/evaloffice/>

Copyright notice



The 2020 ICT Accessibility Testing Symposium proceedings are distributed under the Creative Commons license: Attribution-NonCommercial-NoDerivatives 4.0 International (CC BY-NC-ND 4.0, <https://creativecommons.org/licenses/by-nc-nd/4.0/>).

You are free to: Share — copy and redistribute the material in any medium or format.

Under the following terms: Attribution — You must give appropriate credit, provide a link to the license, and indicate if changes were made. You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use. NonCommercial — You may not use the material for commercial purposes. NoDerivatives — If you remix, transform, or build upon the material, you may not distribute the modified material. No additional restrictions — You may not apply legal terms or technological measures that legally restrict others from doing anything the license permits.

Demodocus: Automated Web Accessibility Evaluations

Trevor Bostic

The MITRE Corporation
7525 Colshire Drive
McLean, VA 22102-7539, USA
tbostic@mitre.org

Jeff Stanley

The MITRE Corporation
7525 Colshire Drive
McLean, VA 22102-7539, USA
jstanley@mitre.org

Daniel Chudnov

The MITRE Corporation
7525 Colshire Drive
McLean, VA 22102-7539, USA
dlchudnov@mitre.org

John Higgins

The MITRE Corporation
7525 Colshire Drive
McLean, VA 22102-7539, USA
jphiggins@mitre.org

Brittany Tracy

The MITRE Corporation
7525 Colshire Drive
McLean, VA 22102-7539, USA
btracy@mitre.org

Justin F. Brunelle

The MITRE Corporation
7525 Colshire Drive
McLean, VA 22102-7539, USA
jbrunelle@mitre.org

Keywords: web application testing; web crawling; user models; automated testing.

Track/Theme: Automated Testing; Government (Federal, State, Local)

Abstract

Government services are increasingly moving online. The need for accessible government web services was made glaringly apparent during the COVID-19 pandemic. While the adoption of JavaScript and other dynamic web technologies to develop web applications increases, the testing of web application accessibility remains reliant on manual efforts. We present *Demodocus*, a framework for automated evaluation of web applications without *a priori* training of evaluation tools. We evaluate the software prototype of Demodocus, showing that it evaluates accessibility 223% slower than a human evaluator but can run continuously. Demodocus discovered 1.7 times more violations with 68% precision compared to the human evaluator (and performed well on WCAG SC 2.5.5). We designed Demodocus to complement other accessibility tools to improve web application accessibility evaluation coverage. Demodocus is useful for initial triage of accessibility issues by experts and non experts, alike, and can help ensure consistent evaluations are performed by human evaluators.

Introduction

As government organizations shift their citizen services and functions from in-person to online, the accessibility of online content continues to grow more important. Inaccessible web content may prevent people with disabilities from using essential government services. Inaccessible web content may prevent many of the 20% of people in the US with disabilities from utilizing services (United States Census Bureau, 2012). Kanta (2018) shows that despite current efforts and legislation to promote the widespread adoption of accessibility practices, many government websites still suffer from a variety of accessibility issues.

The government is not unique in its need to make citizen services more accessible. The importance of and need for accessible web-based services has been highlighted by the COVID-19 crisis. With in-person services often infeasible or potentially unsafe, society is increasingly reliant on web-based services. Education is a prime example of a traditionally in-person service delivery that has moved online during the pandemic. Educators have noted the effects of COVID-19 on students and are adjusting to increase the accessibility of educational materials that have been rapidly shifted online (Taylor and Marois, 2020). The World Health Organization indicates people with disabilities will be affected by the movement of healthcare and other services to online delivery (World Health Organization, 2020).

Dynamic web applications (as opposed to non-interactive web pages) are frequently used as the mechanism to deliver online services. Websites are increasingly using JavaScript to introduce dynamism into web pages (making them web *applications*) as time progresses (Brunelle et al., 2016). Web applications introduce additional challenges to accessible service delivery due to their use of JavaScript and other dynamic content. Social media websites are examples of web applications that introduce many accessibility issues such as infinite scrolling (Wild, 2018).

One accessibility challenge introduced by web applications is the increased difficulty in testing applications for accessible content. Although many tools exist for web accessibility testing, they typically rely on *a priori* user training and scripting, are human-driven, or are limited to automated evaluation of only static content. For example, The General Services Administration uses *pally* (Team Pally, 2018) to crawl and assess the accessibility of both static HTML and the HTML that results from scripted user interactions. Tools like *pally* rely on predefined scripting to interact with the web automatically by rendering and executing the client-side code of a page and interacting with DOM elements as would a human user. Such tools generally lack the ability to discover and interact with the DOM without *a priori* knowledge or scripting.

An alternative is to use a manual testing approach, such as that used by Mankoff et al. (2005). Their approach focuses on assessing accessibility of web pages for blind users and recommends an optimal method of blindness accessibility evaluation using a screen reader. Weber-Hottelmann (2020) analyzes several web accessibility tools and makes recommendations on when to use each tool. However, these tools are either limited to static content, or require user interaction or *a priori* scripting to be successful. Automated accessibility testing can – and should – be integrated into the software delivery pipeline for web applications (Garrison, 2018; Bollinger, 2017); fully-automated testing tools are a prerequisite for large scale and low-cost accessibility evaluations to become standard practice.

To increase the speed, accuracy, and ability of government and other organizations to evaluate their web applications for accessibility violations, we present the *Demodocus* framework. Demodocus is based on current research from web science (e.g., JavaScript-driven web application crawling and testing (Dincturk, et al., 2014)), accessibility (e.g., tools such as pal1y), and user modeling to create an automated tool for identifying and assessing accessibility violations in any web application without prior training (Bostic, et al., 2019).

The Demodocus Framework

We designed Demodocus to crawl web applications using various user models to guide how it interacts with content. Demodocus uses a fully-featured web browser to detect the same interactive elements as a human user, including those driven by JavaScript or CSS styling. We design user models that simulate interactions of users with varying abilities; the user models trigger client-side events during crawls to identify potentially inaccessible content in the web application. For example, a keyboard-only user model will navigate and interact with the web application without executing mouse-based actions (e.g., `onClick`, `onMouseOver`, style changes using `:hover` or `:focus`) to identify potential accessibility violations for the user model. Other user models (e.g., low-vision) simulate a disability preventing a user from using the web application by disallowing interaction with violating content. Candidate accessibility violations are identified when content cannot be accessed by a user model simulating a user with a disability but can be accessed by a user model simulating a user without a disability.

After Demodocus crawls an application, the discovered candidate violations are presented via a dashboard to human subject matter experts for validation and correction. When paired with existing static analysis tools, we expect Demodocus to complement existing accessibility testing approaches and to improve the overall quality and coverage of web application evaluations.

Crawling an Application

Beginning with a URL, Demodocus creates a graph of the web application. The graph is the set of reachable states within the web application. The states in the graph represent unique, reachable client-side representations that a user can reach as a result of an action (e.g., button click, mouse over). The edges that connect the states represent actions that cause the transitions from one state to another. When creating the graph, Demodocus uses its *omniuser* model; this user can discover all states, perform all interactions, and may create omnipotent knowledge of the application's states. As Demodocus creates this *omnigraph*, it identifies and deduplicates equivalent states to reduce complexity and improve the speed of the evaluation. Duplicate states might also lead to skewed or unusable results of the accessibility evaluation.

User Models

Demodocus simulates users with and without disabilities through a series of user models. In these models, Demodocus adjusts the types of interactions available to simulate the user experience available to the specific user being modeled. We expect that these models will be adapted and expanded in the future. Currently, the prototype uses four user models along with the *omniuser*. The *VizKeyUser* models a user that is only capable of interacting with content that

meets WCAG AA contrast requirements and WCAG AAA size requirements. Additionally, the user is unable to execute mouse actions (e.g., `onClick`, `onMouseOver`, `:hover`), limiting the user to only keyboard actions. The *VizMouseKeyUser* expands on the *VizKeyUser* by allowing access to mouse actions. Similarly, the *SuperVizMouseKeyUser* relaxes the constraints on interactive elements to allow content that does not meet contrast or size requirements. In contrast, the *LowVizMouseKeyUser* model tightens contrast requirements to now require meeting WCAG AAA standards for content to be considered for interaction.

Identifying Violations

After generating the omnigraph, Demodocus adopts each of its defined user models to traverse the omnigraph. This creates a navigable replica (that we can consider a subgraph) of the omnigraph that is specific to the states reachable to each user based on its specific abilities, which may limit their available actions (i.e., state transitions). We define a state for a given user model to be *accessible* if there exists at least one incoming edge that is traversable by the user model. This traversal may be non-optimal and still satisfy our definition. Likewise, we define a state to be *inaccessible* for a given user model if it has no incoming edges that are traversable by the user model. A violation is then registered for **each** incoming edge to the state that could be altered to make the state accessible. Note that this means several violations could be fixed by simply modifying one edge. For example, a navbar dropdown may be activatable by the `onMouseOver` or `onClick` events. These would represent edges to the state with the dropdown open that are unusable by our *VizKeyUser* user model. Violations would then be generated for each of the edges as there is no keyboard equivalent to access the state.

We can determine what content would be unreachable to a user model by finding the delta between its graph of reachable states and the omnigraph. This is important as there could be content that is accessible by the definition above but unreachable due to inaccessibilities in prior states. We can further compare the difficulty of the user models' traversal to some end state as compared to the optimal path (e.g., if the omniuser has a shorter access path than another user model). In an optimally accessible web page, we expect the differential between the complete graph and the graphs of reachable states generated by the user models to be minimal. This provides weights to the actions (i.e. edges in the graph) that quantitatively represent the difficulty or effort of reaching a state for a user.

Experiment Methodology

We present an evaluation of the Demodocus framework's first software prototype. We used the Demodocus prototype to evaluate a small sample of web pages from [The WebAIM Million](https://webaim.org/projects/million/) (<https://webaim.org/projects/million/>). We randomly sampled 30 US government web pages (i.e., in the `.gov` top level domain (TLD)) and 30 non-government web pages (i.e., that are not in the `.gov` TLD) to create our *Government* and *Non-Government* datasets, respectively. The Demodocus prototype was able to successfully run on 20 of the 30 government websites and 18 of the 30 non-government web pages. The development team use the failed runs to address bugs in the prototype code. Each crawl was run using the same configuration and without *a priori* information about the site. Failures to complete a page crawl were often due to either bugs that caused the prototype to crash or the inability to recognize duplicate states, which lead to a

crawler trap. For our evaluation, we randomly selected 10 pages on which Demodocus successfully ran from each of the two datasets, creating a dataset of 20 evaluated pages to limit the human evaluator's load.

For each of these pages, the human evaluator began by running several automated tools (including WAVE, Total1y, aXe, WCAG Accessibility Audit, and ANDI) and merged and compared their results. The evaluator then evaluated each page using only a keyboard both with and without the JAWS screen reader enabled. Finally, the evaluator inspected the code of each page to ensure all violations were detected. We compare the performance of Demodocus and the human evaluator to determine the added level of work performed by Demodocus and to determine where each excels and where they overlap. We present the violations discovered by Demodocus and compare the coverage of the evaluation to that of the human evaluator. As a comparator, we compare the statistics of the Demodocus evaluations to those from the WebAIM dataset. The WebAIM dataset was generated from WAVE, a tool that performs a static analysis of a page and does not perform the dynamic page interactions Demodocus performs. By comparing the Demodocus evaluations to human and a static tool analysis of the dataset, we can compare the performance, accuracy, and completeness of the Demodocus prototype to a subject matter expert evaluation and a current static analysis tool evaluation.

Demodocus Prototype Evaluation Results

We compare the evaluations from our human evaluator, Demodocus, and WebAIM on the 10 government and 10 non-government websites. Table 1 shows the results of the Demodocus (*D*), Human (*H*), and WebAIM (*W*) evaluations for the 10 government sites. We compare the results of the 10 non-government evaluations in Table 2. Because WebAIM publishes only a summary of its accessibility violations, we are unable to directly compare Demodocus to the WAVE static analysis tool used in WebAIM. For both Table 1 and Table 2, note that the Page Elements metric is calculated differently for WebAIM and Demodocus. Because Demodocus is intended to work alongside a static analysis tool, WebAIM reports the number of DOM elements on the page (it evaluates all of them) and Demodocus only reports the number of interactive elements that it evaluates. Importantly, Demodocus highlights any unreachable states due to accessibility issues, a calculation unreported by static evaluation tools like WebAIM. Unreachable states in Tables 1 and 2 are reported per user model for each page. While human evaluators do not directly report this metric, it may be derived – through added human effort – from human evaluator notes.

In the government dataset, Demodocus takes an average of 1.2 hours to evaluate a page while the human evaluator takes an average of 0.87 hours per page. From this initial evaluation, Demodocus runs 133% slower than our human evaluator (Demodocus is able to run constantly and without a human in the loop while humans must take breaks). On average, Demodocus was also able to discover approximately as many violations (86.5) as the human (85.6). While WebAIM evaluated 856% more elements than Demodocus, Demodocus discovered 339% more violations than WebAIM.

In the non-government dataset, Demodocus evaluates 335% slower ($\bar{x} = 3.0$ hrs per page) than the human evaluator ($\bar{x} = 0.93$ hrs per page). On average, the human evaluator discovered 70.6% more violations than Demodocus. WebAIM evaluated 547% more elements than Demodocus but

Demodocus discovered 3.9 times more violations. Across both the government and non-government datasets, Demodocus discovered 439% more violations than WebAIM.

Table 1. The statistics of the Demodocus (*D*), human (*H*), and WebAIM (*W*) evaluations for the 10 pages in the Government dataset. The table includes mean (\bar{x}) and standard deviation (s_x). The WebAIM dataset only provides a summary of the violations for each page. We use “*NC*” to designate that a metric was not calculated by the evaluator.

	Crawl Time (hrs)		# States		# Violations		# Page Elements		# Unreachable States	
	\bar{x}	s_x	\bar{x}	s_x	\bar{x}	s_x	\bar{x}	s_x	\bar{x}	s_x
N=10										
D	1.2	1.4	57.1	50.3	86.5	102.3	77.2	99.1	39.2	48.0
H	0.87	0.25	<i>NC</i>	<i>NC</i>	85.6	79.7	<i>NC</i>	<i>NC</i>	<i>NC</i>	<i>NC</i>
W	<i>NC</i>	<i>NC</i>	<i>NC</i>	<i>NC</i>	30.2	30.2	895.7	851.1	<i>NC</i>	<i>NC</i>

Table 2. The statistics of the Demodocus (*D*), human (*H*), and WebAIM (*W*) evaluations for the 10 pages in the Non-government dataset. The table includes mean (\bar{x}) and standard deviation (s_x) of each metric. The WebAIM dataset only provides a summary of the violations for each page. We use “*NC*” to designate that a metric was not calculated by the evaluator.

	Crawl Time (hrs)		# States		# Violations		# Page Elements		# Unreachable States	
	\bar{x}	s_x	\bar{x}	s_x	\bar{x}	s_x	\bar{x}	s_x	\bar{x}	s_x
N=10										
D	3.0	4.5	54.3	40.4	124.7	99.2	77.4	39.9	33.9	37.6
H	0.93	0.22	<i>NC</i>	<i>NC</i>	176.6	149.9	<i>NC</i>	<i>NC</i>	<i>NC</i>	<i>NC</i>
W	<i>NC</i>	<i>NC</i>	<i>NC</i>	<i>NC</i>	28.4	15.8	423.0	168.8	<i>NC</i>	<i>NC</i>

The non-government dataset was predictably less accessible than the government dataset with an average of 1.4 times more violations per page. However, these violations lead to an average of 39.2 unreachable states in government websites as compared to 33.9 unreachable states for non-government websites.

Table 3. The violations identified by each evaluator across both datasets. The current Demodocus (*D*) prototype only recognizes a portion of SCs but can be expanded. WebAIM (*W*) does not identify specific numbers of SC violations identified; we only represent whether the violations were identified (*C*) in the WebAIM data or not identified (*NC*). The metrics for Human (*H*) and Demodocus are per page and include mean (\bar{x}) and standard deviation (s_x).

	S.C. 1.4.3		S.C. 2.1.1		S.C. 2.4.3		S.C. 2.4.7		S.C. 2.5.5	
	\bar{x}	s_x	\bar{x}	s_x	\bar{x}	s_x	\bar{x}	s_x	\bar{x}	s_x
N=20										
D	31.5	36.2	3.75	3.5	1	1.8	19.9	19.8	53.3	52.8
H	44.3	84.7	2.1	2.1	0.35	0.6	10.5	16.4	2.0	3.8
W	<i>NC</i>	<i>NC</i>	<i>NC</i>	<i>NC</i>	<i>NC</i>	<i>NC</i>	<i>C</i>	<i>C</i>	<i>C</i>	<i>C</i>

Low-contrast text was the most frequently occurring error in the WebAIM data with 92% of the 20 pages having that error. Demodocus discovers that SC 2.5.5: Target Size was by far the most discovered violation (Table 3); this is due to the volume of menu items and footer links not being

at least 44px tall and wide. SC 2.5.5 was also largely underreported by the human evaluator, most likely because this is a highly restrictive (Level AAA) and relatively new SC with a less solidified testing procedure. The ability of Demodocus to recognize SC 2.5.5 demonstrates the value of the framework to complement humans and static tools to increase evaluation coverage.

Evaluating Demodocus Performance

We compare the accuracy of Demodocus to the human evaluator in Table 4. While the human evaluator outperformed Demodocus for SC 1.4.3, 2.1.1, and 2.4.3, Demodocus discovered 22.6 time more SC 2.5.5 violations. The human evaluator discovered 2.2 times more SC 1.4.3 violations, presumably because the violation is easily detected by static tools and due to Demodocus’s focus on only interactive content that static analysis tools are unable to evaluate. Overall, Demodocus has a precision of 68% (i.e., 68% of all violations discovered are confirmed violations, or true positives). Without SC 2.5.5, Demodocus would have an overall precision of 54% and discover only 826 violations to the human’s 1080. Further, we include duplicate violations recorded by Demodocus (i.e., the same violation detected multiple times for the same element across multiple states) as a false positive; this decision accounts for ~250 false positives. If duplicates are removed from the false positives, Demodocus’s precision increases to 77%.

Table 4. Demodocus has varying precision for various SCs with 2.5.5 having the highest precision. The volume of SC 2.5.5 True Positives weighs heavily on the Total Precision metric.

	S.C. 1.4.3	S.C. 2.1.1	S.C. 2.4.3	S.C. 2.4.7	S.C. 2.5.5	Total
True Positives	234	27	3	181	858	1303
False Positives	173	54	94	60	240	621
Demodocus Total	407	81	97	241	1098	1924
Human Total	885	41	7	147	38	1118
Demodocus Precision	0.57	0.33	0.03	0.75	0.78	0.68

We designed Demodocus to present all *potential* violations to a human evaluator for correction. While the precision is lower than desired (and poor for SC 2.1.1 and 2.4.3), Demodocus shows that it identified 1/3 of the violations discovered by a human during testing; we determined that Demodocus and the Human found 370 of the same violations (33% of the human-discovered evaluations). Demodocus can present these violations to content owners ahead of a human-driven analysis and reduce the time and effort taken for accessibility evaluations.

Conclusions

We present *Demodocus*, a framework for automated assessment of web application accessibility without *a priori* customization. Demodocus crawls web applications using its omniuser model to create a canonical *omnigraph* of states of a web application. Using this omnigraph, Demodocus simulates users with varying abilities (e.g., inability to use a keyboard) to identify the portions of the graph that are inaccessible and unreachable to users with those abilities. Demodocus presents potential accessibility violations to a content owner or human evaluator to be addressed.

In this initial evaluation with 20 evaluated pages, we show that our current Demodocus prototype discovers 1.7 times more violations (1.2 times more true positives) than the human evaluator at a speed 223% slower than the human evaluator with 68% accuracy. Demodocus also discovered 439% more violations than the WebAIM dataset despite WebAIM evaluating 853% more page elements. Demodocus found an average of 36.5 unreachable states per page. Demodocus can automatically report unreachable states to alleviate human-driven efforts.

From this evaluation, we conclude that an approach such as that implemented in the Demodocus framework will provide best value when paired with a static accessibility tool (e.g., WAVE) and when the results of the discovered *potential* accessibility violations are reviewed and addressed by a human evaluator. We expect the Demodocus framework to reduce the workload on human experts evaluating web application accessibility, reserving their time and expertise for addressing the violations. We envision Demodocus being incorporated into the testing and evaluation stages of DevOps pipelines before releasing a website, and specifically used in online government service delivery. The ability of the crawler to execute autonomously and on multiple sites without the need for prior training or scripting will allow human experts to focus their efforts on prioritization and remediation, should decrease the cost and time of accessibility evaluations, and is expected to increase the accuracy and completeness of web application accessibility testing.

We plan to release the Demodocus prototype as open source software, work to improve efficiency with multi-threading, and improve coverage and accuracy of accessibility violation tests. Future prototype versions will include features that resolve issues illuminated in our testing. We expect to add protection against crawler traps (i.e., cycles in the graphs) such as infinite scrolling pages. We will work to handle iframes, improve coverage of client- and server-side state, and add user models with improved client-side state coverage. We will also add compatibility with modern programming patterns and web design paradigms that were omitted from the evaluated prototype. We also recognize that accessibility violations vary in severity; future Demodocus prototypes will provide emphasis or severity ratings (e.g., Level A) for discovered violations. We anticipate a more comprehensive evaluation with a broader sample size with a future prototype. Simulating a user's interaction with forms is a research challenge that falls out of scope of our research due to its heavy reliance on natural language processing; such a challenge is currently better left to scripted evaluation tools such as pal ly.

Government citizen services will continue to migrate to web applications and the importance of ensuring services are accessible will continue to increase. As a result, fully-automated testing tools – such as Demodocus – are likely a requirement for low-cost, large-scale accessibility evaluations to become standard practice.

References

- D. H. Bollinger, “Using a Component-First Approach in Front End Development and Accessibility Testing”, *The 2018 ICT Accessibility Testing Symposium: Mobile Testing, 508 Revision, and Beyond*. Pages 83-91
- T. Bostic, J. Stanley, J. Higgins, D. Chudnov, R. L. Bradley Montgomery, and J. F. Brunelle, “Exploring the Intersections of Web Science and Accessibility”, *Proceedings of the 2nd International Conference on Human Systems Engineering and Design: Future Trends and Applications*. Pages 483-488, 2019.
- J. F. Brunelle, M. Kelly, M. C. Weigle, M. L. Nelson. The impact of JavaScript on Archivability. *International Journal on Digital Libraries* 17, 95–117 (2016).
- J. F. Brunelle. Scripts in a Frame: A Framework for Archiving Deferred Representations. *PhD thesis, Old Dominion University*, 2016.
- M. E. Dincturk, G.-V. Jourdan, G. V. Bochmann, and I. V. Onut. A Model-Based Approach for Crawling Rich Internet Applications. *ACM Transactions on the Web*, 8(3):19:1–19:39, July 2014.
- A. Garrison, “Continuous Accessibility Inspection & Testing”, *The 2017 ICT Accessibility Testing Symposium: Automated & Manual Testing, WCAG 2.1, and Beyond*. Pages 57-65
- S. Kanta, “Insights with PowerMapper and R: An exploratory data analysis of U.S. Government website accessibility scans” *The 2018 ICT Accessibility Testing Symposium: Mobile Testing, 508 Revision, and Beyond*. Pages 65-72
- J. Mankoff, H. Fait, and T. Tran. Is your web page accessible?: A comparative study of methods for assessing web page accessibility for the blind. *SIGCHI Conference on Human Factors in Computing Systems*, pages 41–50, 2005.
- Z. W. Taylor and M. Marois, Ensuring ADA Compliance During COVID-19: Communication for People with Disabilities (*Presentation Slides*) (April 7, 2020). <http://dx.doi.org/10.2139/ssrn.3570904>
- K. Weber-Hottleman, “Comparing Free Automated Accessibility Testing Tools”, *The 2019 ICT Accessibility Testing Symposium: Perfecting Traditional Methods, Tackling Emerging Interfaces, and Beyond*. Pages 51-57
- G. Wild, “Testing social media for WCAG2 compliance”, *The 2018 ICT Accessibility Testing Symposium: Mobile Testing, 508 Revision, and Beyond*. Pages 57-64
- Team Pa1ly. Pa1ly. <https://github.com/pa1ly/pa1ly>, 2018.
- United States Census Bureau. Nearly 1 in 5 People Have a Disability in the U.S., Census Bureau Reports. <https://www.census.gov/newsroom/releases/archives/miscellaneous/cb12-134.html>, 2012.
- World Health Organization, Disability considerations during the COVID-19 outbreak. <https://apps.who.int/iris/bitstream/handle/10665/332015/WHO-2019-nCov-Disability-2020.1-eng.pdf> January 2020.

Copyright notice

© 2020 THE MITRE CORPORATION. ALL RIGHTS RESERVED. Approved for public release. Distribution unlimited. Case number: 20-2317.



The 2020 ICT Accessibility Testing Symposium proceedings are distributed under the Creative Commons license: Attribution-NonCommercial-NoDerivatives 4.0 International (CC BY-NC-ND 4.0, <https://creativecommons.org/licenses/by-nc-nd/4.0/>).

You are free to: Share — copy and redistribute the material in any medium or format.

Under the following terms: Attribution — You must give appropriate credit, provide a link to the license, and indicate if changes were made. You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use. NonCommercial — You may not use the material for commercial purposes. NoDerivatives — If you remix, transform, or build upon the material, you may not distribute the modified material. No additional restrictions — You may not apply legal terms or technological measures that legally restrict others from doing anything the license permits.

Approaches to Remote Testing Using People with Disabilities to Achieve Inclusion

Sue Ann Rodriquez

Director of Accessibility Services
WeCo Accessibility Services
323 Washington Avenue North, Second Floor
Minneapolis, MN 55401
sueann@theweco.com

Kelli Ryan

Director of Operations
WeCo Accessibility Services
323 Washington Avenue North, Second Floor
Minneapolis, MN 55401
kelli@theweco.com

Keywords: Remote testing, usability, accessibility, inclusion

Track/Theme: ITC remote testing, provisions of training to testers, and management of testing teams within website and application testing domains.

Abstract

Due to COVID-19, people around the world have transitioned to conducting their lives through online venues and remote digital systems. With the dramatic and quick digital shift, non-profits, government entities, educational institutions, and companies are facing challenges when it comes to achieving inclusion for all. Usability testing performed by people with disabilities can be beneficial to those who want to create an accessible and optimal user experience for all. Conducting remote usability testing can easily be done without the additional considerations and challenges that come with conducting in-person testing. Sue Ann Rodriquez, the author of this paper who lives with a disability and is the Director of Accessibility Services at WeCo (a consulting firm made up of digital technologists and testers who are required to live with disabilities as part of their remote usability testing work) will discuss the approaches to consider for training, testing, and managing test teams.

Introduction

The COVID-19 pandemic has forced people worldwide to go online to work, attend classes, shop, and socialize. Types of online activities that were once just considered as an alternative option. One of the biggest objectives resulting from COVID-19 has been the drive towards inclusion, a topic typically associated with people living with disabilities, but that has now become a necessary for people without disabilities as well. Now that people without disabilities require the same type of inclusion as those with disabilities, focusing on accessibility and usability has a far greater impact than ever before.

Usability testing performed by people living with disabilities can be beneficial to those who want to create an accessible and positive user experience for all. According to UTT (2020), as people age, they often develop vision, hearing, dexterity, mobility, and cognitive issues. Any usability testing done by people living with disabilities will benefit aging users as well (UTT, 2020). Usability testing focuses on testing for learnability, efficiency, satisfaction, etc. (Nielsen, 2012). While accessibility testing focuses on testing to determine the ability to access and interact with content and features (such as forms) by people with disabilities who may or may not use assistive technology.

Accessibility Doesn't Equal Usability

In a time when many people are limited to shopping exclusively online, it is more essential than ever to ensure that online venues (such as websites and apps) are not only accessible, but will also allow users to locate and perform various functions successfully.

For example, if a user searches for “Bayer Aspirin,” but the search results listed are cold medicine products, the search feature is not helpful for users attempting to locate their desired product. Also, if purchasing a product requires a user to wade through several product options because there is no filtering feature or the feature doesn't work properly, users must spend extra time and energy in locating their desired product. If a text field requires a zip code to be entered, but the label for the field is “Location,” users will not know what information is to be entered into the field because the label is not clear. These items may be technically accessible, but the user interface design can cause users to have a negative user experience, as they may not be able to locate a needed item, or worse, may not be able to complete their purchase. This can cause users to become frustrated, and ultimately leave without making a purchase.

A paper by Sundby, Herr, & Howle, titled "How WCAG 2.1 Benefits People with Cognitive Impairment" (from the 2019 ICT Accessibility Symposium) remarks how it is important that people are provided with clear, descriptive, and sufficient information, so they can succeed at completing a task the first time. The impact of errors can increase anxiety, contribute to low self-confidence, or trigger depression.

The shift to online solutions for remote learning is another area in which accessibility and a user's experience is vital to avoid leaving students behind. For example, school districts and teachers are now tasked with ensuring that their classroom online interfaces and textbooks are

accessible to students with a wide variety of disabilities. Not all have the luxury of a department devoted to digital accessibility, to accomplish this.

This means that a learning resource may be technically accessible, but the user interface design may make a screen reader user take so long to navigate around that they give up before achieving or even discovering the learning objective.

Recorded online classes also present accessibility issues. Captioning for students who are deaf or hearing-impaired is crucial, and correct captioning is key to foster learning. Sundby et al. (2019) states how having accurate captions with 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 a cognitive impairment. When the captions are poorly edited with misspellings, missing capitalization, and missing or incorrect punctuation, it becomes far more difficult for these students to learn.

With inclusion on the forefront of people's minds today, the question is: how does one know if their online venue (such as a website, app, etc.) is accessible and able to provide a positive user experience? The answer: usability testing, executed by testers who live with the disabilities we are seeking to include.

Disability-focused usability testing can verify not only if all aspects of websites and electronic documents work for users living with disabilities, but also if they function as the user anticipates, providing a useful, comfortable, and satisfying experience.

A few examples include determining if a search text box or a filter drop-down box has clear labels and functions in a predictable manner. Another example is verifying if a label for a text field, such as the zip code example discussed above, makes sense to users, eliminating the need for users to guess what information they need to enter. Usability testing can also determine if a learning resource handout is easy for people using screen readers, for example, to navigate effectively and efficiently.

Which type of usability testing is most suitable for all people with disabilities?

Websites and apps that are technically accessible might still be unusable. Conducting usability testing using testers with disabilities can help organizations obtain feedback with first-hand experience of how people with a wide array of disabilities interact with websites, apps, and other digital venues, due to their grasp of digital needs from their own perspective. But which usability testing type should be implemented when conducting usability tests with people with disabilities?

In-Person Testing

Traditional usability testing has typically been conducted in person. This type of testing involves a facilitator (also referred to as the moderator) to interact with the participant by asking the participant to perform tasks and respond to questions (Moran, 2019). Usually, there will also be one or two additional observers in the testing session to document the participants' actions and spoken dialog during the session. Conducting in-person usability testing has various benefits, such as:

- Moderator and other observers can analyze, experience, and document participants' body language, as they can observe how the participants' facial expressions correlate with their actions and words.
- A moderator can guide participants through the test, ask clarifying questions, have conversations around participants' confusion, etc.
- The testing session can be recorded for further analysis after the session has concluded.

There are also various challenges that must be taken into consideration when conducting in-person usability testing with people living with disabilities, such as:

- If the facility and the testing room will need to be accessible (such as having wide, clear aisles to accommodate wheelchairs for participants with mobility disabilities, or service dogs for participants who are blind or low vision.)
- Participants who require assistive technology may require additional time to complete the test, which could extend beyond the scheduled testing session. This could apply to participants who are blind or low vision and use screen reader or magnification software.
 - UTT (2020) explains how assistive technology (such as screen readers, speech recognition software, and magnifier software) add a level of complexity to the use of a website or application. This means more time is likely needed to perform tasks (UTT, 2020). As a rule of thumb, UTT (2020) suggests allowing twice the time for task performance than is allocated to non-disabled participants.
- It may be difficult for participants to travel to the testing facility due to their disability.
 - Participants with disabilities which impact physical and mental energy may become fatigued by the effort it would take just to get to the facility. It may be necessary to pare the task list down to fewer or simpler tasks for these participants, who may become more easily fatigued during the scheduled testing session.
- Participants with cognitive disabilities, such as learning impairments and attention deficit disorders, may need to have the test questions written in a simpler, easier to read format. They may also require that the questions be read out loud and/or explained to them for comprehension and to keep them engaged.

A WeCo Tester who has Multiple Sclerosis (a disability that affects cognition, energy, and mobility) explains how participating in an in-person usability test would be difficult for her for various reasons. "Before I even got into the usability testing room, I would already be fatigued

because the effort that it would take me to travel to the facility, park, and get into the facility would wear me out. And during the session, I would have a difficult time in concentrating on the task that I'm trying to perform as well as needing to verbalize my thoughts due to my disability.”

Thus, conducting in-person usability testing can be difficult to execute, and not the most suitable option.

Remote Testing

Remote usability testing has become a popular option, especially during these times when virtual interaction has become an important way to manage public health. There are two types of remote testing: moderated and unmoderated.

Moderated Remote Testing

Moderated remote testing involves a facilitator (or moderator) and participant being in the same “virtual” space at the same time (Schade, 2013). The moderator watches the usability test session remotely as it happens, and communicates directly with the participant (Schade, 2013).

Conducting moderated remote testing has various benefits, such as:

- Moderator and other observers analyze, experience, and document participants’ body language, as they can observe how the participants’ facial expressions correlate with their actions and words. (This is dependent on the camera and recording tool capabilities.)
- A moderator guides participants through the test, asking clarifying questions, and addressing any issues or confusion the participants may encounter.
- The testing session can be recorded for further analysis after the session has concluded.

There are various challenges that must be taken into consideration when conducting moderated remote testing with people living with disabilities, such as:

- Participants may not have the system requirements to have adequate technology (e.g., camera, video conference tool, etc.) to complete the test or be able to install/use the technology due to it being inaccessible and/or due to their disability.
- Some participants (with a cognitive disability, for example) may become anxious about being on camera, and for those with an attention deficit disorder may find the camera distracting, thus affecting their results.
- Screen recording capturing software may not always be 100 percent secure.
- Connection speed requirements may limit participants residing in certain geographic areas, from participating in tests.

A WeCo employee who has attention-deficit/hyperactivity disorder (ADHD), a neurodevelopmental disorder that includes inattention, hyperactivity, and impulsivity (American Psychiatric Association, 2020) and low vision describes how participating in a moderated remote usability test would be difficult for him. “Due to my ADHD, it is difficult for me to focus and be able to focus for long periods of time. Having someone interrupt my focus, by asking me a question and reminding me to verbalize my thoughts as I’m attempting to complete tasks, would force me to exert extra energy to regain my focus. This would eventually result in me not being

able to focus 100% because I'm using more energy to get my focus back to the task that I'm attempting to complete every time my focus is interrupted.”

These potential challenges make moderated remote usability testing difficult to accomplish, therefore not the most suitable option.

Unmoderated Remote Testing

In unmoderated remote testing, the participant completes the usability test session alone and on their own schedule without a facilitator. (Schade, 2013). Benefits of unmoderated remote testing include:

- Allows participants to complete testing work at their own pace. The advantages are described here by a WeCo Tester who lives with a blindness and a mobility disability, “I don't feel rushed to complete a usability test, as I can take breaks and go back to finish it when I'm well rested.”
- Testing conducted in a participant's home is more accurate, realistic (reflect the environment of the real end-user) and allows participants to feel more comfortable.
- Allows participants to attempt to complete more tasks, because they do not need to be completed in the same limited time frame.

There are various challenges that must be taken into consideration when conducting unmoderated remote testing with people living with disabilities, such as:

- There is no moderator, or any other observers, present to be able to analyze, experience, and document participants' body language. This means that participants must document their own experiences and convey them remotely.
- There is no moderator present to guide participants through the test, such as answering questions and assisting with issues. This means that participants will be responsible for reaching out for assistance when issues arise.

Based upon WeCo's experience with successfully conducting unmoderated remote testing, this type of testing is deemed as the most suitable option. This is because unmoderated remote testing provides multiple benefits and the least amount of challenges to obtain valuable and helpful results from people living with disabilities.

Provision of Training for Testers

Accommodations

Through the planning and implementation phases for providing training to testers living with disabilities, keep accessibility and accommodations in the forefront of everyone's mind.

1. When determining the venue to deliver training, avoid having onsite or in-person training as the only available option, since this can limit the participation of some people living with disabilities. Offering a virtual option ensures that everyone can participate.

2. When communicating the training to participants, include verbiage asking them to indicate any accommodations they will require to attend and participate during the training. Another option would be to create a registration form where participants can list any accommodations they will require. Wufoo, a tool with the ability to customize online forms used to collect data, is accessible to users of assistive technology.
 - a. It is not necessary to include accommodations, such as closed captioning if no participants identify it as a need. However, if the training is intended to be recorded to be used in the future, or will be put on an online platform, such as YouTube, have it verbatim captioned. (Since YouTube captioning, as of this writing, is not accurate, hence the importance of uploading a verbatim script to accompany the video.)
3. Conduct training using an accessible online video conferencing platform, such as Zoom. (For WeCo, Zoom has proven to be accessible for participants living with disabilities.)
4. Prior to the training, request that participants login to the virtual conference room to ensure they can gain access to the room without issues and to familiarize themselves with the layout and features.

Accessibility

1. All materials that are created for your test preparation and implementation (email notifications, accompanying handouts, surveys related to the training presentation, etc.) need to be in an accessible format.
2. Provide all materials to participants prior to training so that they can review them beforehand, as well as be able to follow along during the training.
3. Use visuals in the training presentation slides and in the supplemental training materials. The simplification of written language with the use of supporting images aids in the ability to take in information properly (Sundby et al., 2019).
4. Consider what might inhibit certain disability groups from accessing and understanding the information provided during the training.
 - a. For example, having too much content on a presentation slide may be overwhelming or distracting to individuals with a cognitive disability.
 - b. Another example is that any visual components on the presentation slides will need to be described for attendees who are unable to view them.
5. Do not rely on visual cues alone to get important information across. Offering visual and verbal cues will help attendees of all abilities to learn and retain the information that is being provided.

Training should include tips on how attendees can ask questions, as some features (such as “raise hand”) may not be accessible to all assistive technologies. When asking for questions, comments, or feedback, provide multiple options of producing them:

1. Chat box within the video conferencing tool,
2. Offer an email address of the presenter or moderator, and
3. Offer ample time for attendees to verbally ask their questions or give their comments and feedback.

Offer multiple opportunities for attendees to participate interactively in the training, such as asking questions and offering comments or feedback, to ensure that they are absorbing the information that you are presenting.

Management of Test Teams

Attempt to hire at least one participant representing the four major disability classification groups (sight, hearing, mobility, and cognitive) recognized by the Department of Human Services. This will ensure your test group has a broad disability user focus. It is legally feasible to solicit, recruit and hire testing participants who live with specific types of disabilities to match the goals of the usability test. Reference the protections of Bona Fide Occupational Qualifications (*Office of the Revisor of Statutes 363A.08*).

While some participants may be hired as contractors, companies can still require mandatory training to assist the participant's ability in providing feedback in a manner that will meet the goals of all stakeholders involved. Be sure to provide specifications and expectations of participants regarding completion of usability testing work. Allow the participants the opportunity to become familiar with them and to ask questions if anything is unclear to them.

Summary

With the many consequences associated with a global pandemic, there is a great demand for ensuring that everyone can complete purchases, access vital information, be able to communicate, and socialize. Failure to address accessibility and usability will result in missed opportunities for achieving inclusion, which increases morale and productivity for all users. Usability testing implemented by people who live with disabilities, is an excellent opportunity to foster better accessibility and inclusion practices, across global organizational structures. The results of which may remain in place beyond the present pandemic and serve as a major leap forward to including a wider portion of users living with disabilities in meaningful access to information, services, employment and much more.

References

- American Psychiatric Association. (2020 Aug. 24). What Is ADHD? Retrieved from <https://www.psychiatry.org/patients-families/adhd/what-is-adhd>.
- Moran, K. (2019, Dec. 1). Usability Testing 101. Retrieved from <https://www.nngroup.com/articles/usability-testing-101/>.
- Parthasarathy, S. (2020, May 21). Why Digital Accessibility Is Non-Negotiable While We Deal with COVID-19. Retrieved from news.microsoft.com/en-in/digital-accessibility-covid-19-shriram-parthasarathy.
- Schade, A. (2013 Oct. 12). Remote Usability Tests: Moderated and Unmoderated. Retrieved from <https://www.nngroup.com/articles/remote-usability-tests/>.
- Sundby, V., Herr, K., & Howle, V. (2019, Sept. 30). *How WCAG 2.1 Benefits People with Cognitive Impairment*. Retrieved from <https://www.ictaccessibilitytesting.org/wp-content/uploads/2019/09/Proceedings-of-the-2019-ICT-Accessibility-Testing-Symposium.pdf>.
- UTT, M. H. (2020, Aug. 24). Usability Testing by People with Disabilities: Some Guerrilla Tactics. Retrieved from https://uxpamagazine.org/guerrilla_tactics/.
- Office of the Revisor of Statutes (2019) 2019 Minnesota Statutes:363A.08 Unfair Discriminatory Practices Relating to Employment or unfair Employment Practice Subd.4. <https://www.revisor.mn.gov/statutes/cite/363A.08>

Copyright notice



The 2020 ICT Accessibility Testing Symposium proceedings are distributed under the Creative Commons license: Attribution-NonCommercial-NoDerivatives 4.0 International (CC BY-NC-ND 4.0, <https://creativecommons.org/licenses/by-nc-nd/4.0/>).

You are free to: Share — copy and redistribute the material in any medium or format.

Under the following terms: Attribution — You must give appropriate credit, provide a link to the license, and indicate if changes were made. You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use. NonCommercial — You may not use the material for commercial purposes. NoDerivatives — If you remix, transform, or build upon the material, you may not distribute the modified material. No additional restrictions — You may not apply legal terms or technological measures that legally restrict others from doing anything the license permits.

Developing and Testing Accessible eLearning Courses

Lisa M. Mayo

ICF Next
9300 Lee Hwy, Fairfax, VA 22031
Lisa.Mayo@icfnext.com

Jolie Dobre

ICF Next
9300 Lee Hwy, Fairfax, VA 22031
Jolie.Dobre@icfnext.com

Aruna Kedarshetty

ICF Next
9300 Lee Hwy, Fairfax, VA 22031
Aruna.Kedarshetty@icfnext.com

Keywords: e-learning, remote learning, online learning, online course

Track/Theme: Delivering online instruction, moving course materials to electronic formats

Abstract

The term eLearning refers to using electronic technologies to access educational curriculum outside a traditional classroom. Organizations create and publish eLearning content using sophisticated authoring software, Learning Management Systems, and a network. This type of remote learning is becoming even more popular in the age of COVID-19; however, it can be challenging for instructional designers to know how to make eLearning content accessible. Proprietary players, multimedia, animation, and interactive quizzes all provide engaging content but present a variety of accessibility difficulties. This paper will provide an overview of key accessibility considerations and technical solutions for both eLearning designers and accessibility testers. In addition, the paper will offer guidance on incorporating the accessibility testing team into the designing process and capturing remediation guidance for future eLearning projects.

Popularity of eLearning in the Age of COVID-19

In December 2019, the website Research and Markets (Renub Research, 2019) reported that the global online education market will reach \$350 billion (U.S. dollars) by 2025, due to the introduction of flexible learning technologies in the corporate and education sectors. This impressive amount of growth was predicted before the COVID-19 pandemic, and the health emergency has only increased the demand for eLearning. Research and Markets reports that the United States and China are the top two countries in the global online education market, and as more U.S. organizations move toward online learning out of health and safety concerns, the need for producing accessible eLearning courses has never been greater.

Organizations wishing to publish eLearning courses use a software authoring program that enables them to create and arrange content for delivery in several multimedia formats. The most common output is the SCORM (Shareable Content Object Reference Model) format, which can then be uploaded to a learning management system, which is a software application that offers administration, tracking, reporting, and delivery of online educational courses. Adobe Captivate and Articulate Storyline are two eLearning authoring programs that are widely used. Our team of instructional designers and Certified Trusted Testers have noticed that many of our federal government clients have used one of these programs to produce their eLearning courses. The discussion that follows will address the accessible authoring considerations for programs such as Captivate and Storyline, although our recommendations can be used with a variety of eLearning authoring programs.

The Challenges

Organizations look to the World Wide Web Consortium's Web Content Accessibility Guidelines (WCAG) (World Wide Web Consortium, 2008) for guidance in making online content accessible, but WCAG does not specifically speak to the eLearning environment and its unique challenges, as WCAG is technology neutral. WCAG offers an important foundation for developing accessible eLearning, but other factors come into play, such as the inherent limitations and unique features of certain eLearning software programs, the limited number of instructional designers with accessibility experience, and the limited number of accessibility testers with knowledge about the methods and techniques for creating and testing eLearning courses.

As we began accessibility testing for courses created by these programs, our first important observation was that the accessibility testing team (ATT) should be involved in the early design and development decisions – or what is commonly referred to in the software development world as “shift left.” By shifting the ATT's role to the left, the team could take the lead in investigating the built-in accessibility features of the chosen authoring program and advise the team on making use of these important features, which could save a great deal of time and money. In addition, shifting the team left would allow them to weigh in on important design considerations, such as compliant color palettes, button and menu location, question types, and special coding techniques, such as adding “opens a new window” to the accessible names for external links.

Producing and Testing Accessible eLearning Courses

Our testing experiences have helped us formulate a framework for improving the accessibility of online training while reducing rework and shortening our testing cycles. We have found a correlation between earlier ATT involvement and fewer issues identified during testing. We recommend that an ATT be involved in all phases of design, from planning through maintenance. (See Appendix A: A Framework for Accessibility Testing Team (ATT) Involvement in the Production and Testing of Accessible eLearning Courses.)

In an ideal world, the ATT will contribute to the selection of the eLearning authoring program that the design team will use to produce its online courses. By conducting research – such as reading the program’s Voluntary Product Accessibility Template and testing demo versions – both teams can determine the most accessible product to purchase. Ideal products have several features, such as the ability to customize training slide templates to meet accessibility standards, keyboard accessibility for navigation and selection, and accessible color palettes, which not only ensure compliance but reduce design time.

In reality, the organization often acquires the program before the ATT is consulted, but regardless of how the product is selected, it is important to have the ATT research the program’s product website and support documentation to discover and document the available accessibility features. For example, in January 2020, Articulate made major accessibility updates to its Storyline 360 program that included requiring screen reader users to use the Up/Down arrow keys to move through all text and interactive objects on the course slides, while using the Tab/Shift+Tab keys to jump between interactive objects. The goal of these updates was to reduce keyboard strokes for keyboard-only users (Articulate, 2020, July 27). Storyline 360 uses the program’s “tab order” feature to control the reading order as well as the navigation order of slide objects. It is crucial for both the designers and ATTs to understand and make use of these built-in features in the production and testing of accessible course content.

During the planning and analysis phases, the ATT can evaluate the design team’s accessibility competency and can coach inexperienced teams. The ATT can also provide a “pre-flight” accessibility checklist for designers so they can test their own slides and reduce the number of minor errors, like missing alternative text and illogical tab order. We have found that providing an overview of standards, conducting empathy training, providing checklists, and supporting the development of user personas and user stories that feature users with hearing and vision impairments and mobility limitations results in better design and more productive solutioning of accessibility violations.

For those who are contracted to perform testing, an assessment of the client’s approach to accessibility is critical. We sometimes find that clients have internal accessibility governance teams responsible for enterprise accessibility but are not always aware of development initiatives. They may be required to review test results and may have requirements beyond WCAG and Section 508. Understanding their expectations early ensures all requirements are tested and timelines are adjusted for collaboration. Our team uses Confluence, a knowledge management tool, to capture information about our testing activities so we can learn from prior testing engagements, provide consistent experiences to our clients, and store resources for our ATT members. (See Appendix B: A Template for Knowledge Capture.)

We recommend that the ATT review Confluence when assigned to a project, update it throughout the engagement as needed, and then conduct a final update, after testing concludes, with lessons learned and links to deliverables such as test reports, job aids, and training presentations. During the design phase, the ATT can support template development, confirm that color selections are compliant, and test base templates (with sample text and images) for compliance issues related to navigation and keyboard access. When possible, our team encourages the use of storyboards. Designers often use storyboards to capture the intended slide text, captions, audio descriptions, images, image alternative text, slide branching, and navigation. The ATT can review storyboards before design begins and then use the storyboard as a resource during testing. (See Appendix C: Storyboard Template for a look at our template.)

In the following section, we will discuss important considerations in designing and testing accessible course content, basing our accessibility evaluation on the Section 508/WCAG 2.0 Level A/AA standards.

Keyboard Accessibility

When designing and testing an eLearning course, make sure that every activity, object, and element in the course that conveys meaning is keyboard accessible and that no keyboard traps exist. Also, ensure that the focus indicator is visible when each item has the current keyboard focus. Avoid individual keystrokes that require specific timing to complete an action, as this can be challenging for keyboard users and those with cognitive disabilities. In addition, investigate whether the authoring program offers certain keys (or custom keyboard shortcuts) to navigate and interact with content, and document these keys for the user.

Some authoring programs allow the designer to slowly fade in content on the slide. Avoid showing/hiding elements (such as time-delayed text), as this type of display can cause screen reader users to miss important content as they navigate through the slide with the keyboard.

To ensure that all keyboard/screen reader users understand how to navigate the course, consider beginning the course with a “Navigating the Module” section where the course instructs the user on the location of common interface controls and common slide elements and the keys that should be used to interact with them. By adopting this approach, users are not left to guess at how they should interact with an unfamiliar program.

Standards: WCAG 2.1.1 Keyboard, 2.1.2. No Keyboard Trap, 2.4.7 Focus Visible

Color

Color is a useful and engaging tool in eLearning courses; however, designers must ensure that their color palette is accessible to color-blind and low-vision users. When designing course content, ensure that color is not the only means to convey information, and select colors that meet color contrast requirements as described under WCAG 1.4.1 Use of Color.

Articulate Storyline, Adobe Captivate, and other eLearning authoring programs offer built-in templates, but their predefined color themes may not have compliant colors for all player controls, buttons, and link states. Explore whether the chosen program allows designers to

customize a theme color or create a new custom theme. In addition, some programs may also allow the designer to edit the output files and adjust styles after publication.

Using colored states for objects such as buttons, tabs, and menu items can be problematic for color-blind and low-vision users, so verify that all states pass the color contrast ratio, or consider eliminating states that are not required to convey important information. Another option is to use symbols or other visual indicators to convey the same information. For example, in Storyline 360, a colored checkmark symbol can be added to a completed section to indicate that the user has finished the content.

Standards: WCAG 1.4.1 Use of Color, 1.4.3 Contrast (Minimum)

Descriptive Text

Write good alternative text (accessible names) for all relevant objects in each slide of the course. Screen reader users rely on this text to understand the meaning and purpose of the object. Be sure your alternative text conveys the same information as the object or image and does not duplicate other slide text. The alternative text does not need “image” or “graphic” in the name and should not include abbreviations or unnecessary punctuation (Articulate, 2015). If the image or graphic is purely decorative, it does not need alternative text and it can be left out of the tab/reading order.

If the course includes an avatar character that guides or engages the user, evaluate whether the user needs to understand what the character is doing or saying, or if the character is providing meaning that is not being captured elsewhere on the slide. If the character’s presence is meaningful, then give it alternative text; otherwise, the character can be considered decorative (Articulate, 2015).

When a slide has an image and text element that are meant to be interactive and they are both triggering the same function, consider grouping them into one object, removing the alternative text from each, and then giving the group an accessible name. The same technique can be applied to complex graphics with multiple images.

If the course includes an animation, consider offering a text-based alternative. The animation slide can include a link to a static, text-based layer with the same information for screen reader users (Articulate, 2020, August 7).

For link text that opens a new window and takes the user away from the course, be sure to indicate this in the accessible name assigned to the link. For slides that contain a collection of external links (such as a Resources or Citations slide), include the text “links open in a new window” in the introductory paragraph, so the message is concisely communicated for all the links.

Standards: WCAG 1.1.1 Non-text Content, 1.3.1 Info and Relationships, 2.4.4 Link Purpose (In Context), 3.2 Predictable

Reading Order

Reading order and tab order are important to keyboard/screen reader users who navigate text, images, interactive content, and player controls using the keyboard. When the reading order is poorly designed, it can be confusing for the user. For example, if a slide walks the user through four steps to prepare a meal, but the reading order is set to step 1, step 4, step 2, step 3, the user will not find the content helpful.

Follow these guidelines when creating the reading order and tab order:

- Organize content left to right and top to bottom when possible. For complex slides, customize the reading order in a way that is meaningful for the users.
- Maintain navigation consistency across all slides. Ensure video player controls are in a consistent location on each slide. If using Previous and Next buttons, locate them in the bottom right on each slide, so they are read last by screen readers (Articulate, 2015).
- For quizzes, ensure the full question is read first, followed by the response options sequentially. When using buttons such as Reset Response, Cancel, Submit, or Next, use them consistently throughout the course and in the same order.
- For form fields, consider adding the field name or instruction text before the input field, as screen readers will not read alternative text for input fields (Articulate, 2015).
- When using text and graphics on the same slide, consider locating the text before the graphics, so the user hears the descriptive content read first. If using image captions, write alternative text that includes the image description and any caption, and hide the caption from screen reader users (Articulate, 2015).

Standard: WCAG 1.3.2 Meaningful Sequence

Navigation

Navigation in an eLearning course often includes an interactive menu that presents a list of the slides in the course and a set of Previous and Next navigation buttons for progressing through the slides (see Figure 1). Navigation aids benefit all users, but screen reader users are negatively impacted when they are forced to engage with repetitive navigation on every slide. Give users the ability to show/hide or turn off persistent navigation. For example, in Storyline 360, the menu tab panel allows the user to show or hide the panel by clicking a link.

Use a Skip Link button to bypass repetitive content and place the skip link in the same order on each slide (at the end of the slide content). For example, in Storyline 360, when users get to the end of the slide and press the Tab key, a Skip Link button appears and is announced to the screen reader user. The user can press the Enter or Spacebar key to return to the top of the slide, thus enabling the user to skip the player controls.

When writing slide titles that will appear in the menu panel and be read out to a screen reader user, create unique titles that help the user discern between different slides. For example, if the

course includes multiple “Knowledge Check” slides, number the slide titles (e.g., “Knowledge Check 1”), so each title is unique in the menu.

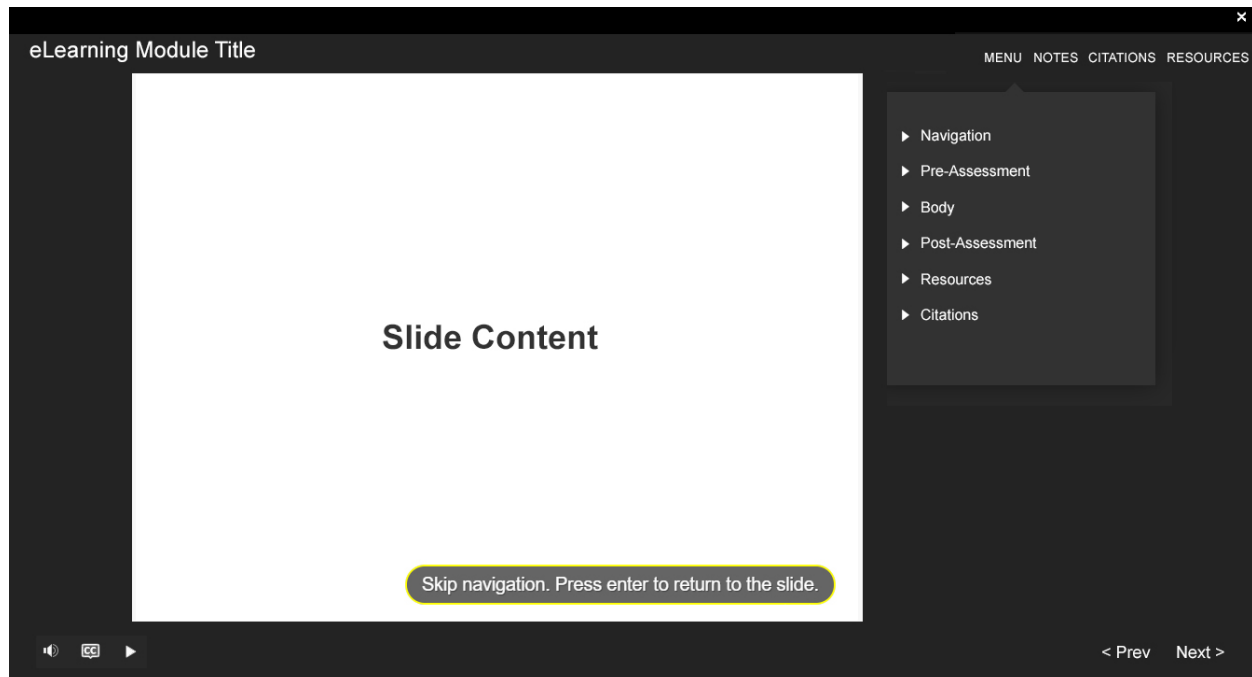


Figure 1. Sample interface

Standards: WCAG 2.4.1 Bypass Blocks, 3.2.3 Consistent Navigation, 2.4.2 Page Title

Audio and Video Content

Many eLearning courses improve engagement by using audio and video in the course presentation; however, multimedia can present difficulties for those with auditory, visual, and cognitive impairments.

Follow these guidelines when creating audio and video content:

- Provide captions for deaf users and ensure the content is equivalent to the audio/video track. Add the closed captioning button to the video player, so users can toggle it on or off. Ensure the button is in the same location on each slide.
- Include a transcript of audio and video. Transcripts are helpful to those using assistive technology, such as refreshable Braille. The transcript can be displayed in a text box or contained in a special section in the navigation bar. For example, in Storyline 360, the Notes Panel in the navigation bar can display the transcript for each slide (Articulate, 2015).
- If the video includes visual content that is not described via the audio, include an audio description for blind users. This information can be included in the transcript.

- Avoid playing videos automatically, which can confuse deaf and blind users and people with cognitive disabilities. Give the user control of the multimedia, but if possible, offer a toggle to turn off manual play if the user so desires.
- If audio plays automatically, provide a mechanism to pause audio using the keyboard. As a best practice, the control should be within the first few elements in the tab order.

Standards: WCAG 1.2.1 Audio-only and Video-only (Prerecorded), 1.2.2 Captions (Prerecorded), 1.2.3 Audio Description or Media Alternative (Prerecorded), 1.2.5 Audio Description (Prerecorded)

Question Types

Many eLearning courses include quizzes to test the user's knowledge. Drag-and-drop interactions, mouse/hover reveals, or any other quiz interactions that require mouse control or vision are difficult to make accessible and should be avoided. The following question types offer a more accessible alternative for keyboard/screen reader users by making use of text fields and standard keyboard-accessible form elements (Articulate, 2015)

- Multiple choice
- True/false
- Fill in the blank
- Numeric
- Short answer
- Multiple response
- Pick one, pick many
- How many
- Matching drop-down
- Ranking drop-down
- Essay question

When presenting complex questions that are spread across multiple slides (e.g., a scenario question where the text is displayed on two slides due to space constraints), consider combining the entire text into one chunk of text (via the alternative text/accessible name) for the screen reader user, rather than asking them to use Previous and Next buttons to navigate between slides to answer the question.

Standard: WCAG 2.1.1 Keyboard

Conclusion

With the growing need to reach distant consumers, eLearning platforms will continue to be the preferred method to quickly bring new training to market. To ensure eLearning products are compliant and brought to market quickly, ATTs need to be involved early in the process. Early involvement ensures that accessibility is designed into the final product through thoughtful development of compliant content and the use of compliant templates that preclude reliance on color, poor contrast, lack of keyboard accessibility, or use of less accessible question types. Early engagement not only improves the final product but also supports and educates design teams, improving quality and cadence.

References

Articulate. (2015). *6 best practices for designing accessible e-learning*.

<https://community.articulate.com/e-books/6-best-practices-for-designing-accessible-e-learning>

Articulate. (2020, July 27). Storyline 360: Slide content is more accessible.

<https://articulate.com/support/article/Storyline-360-Accessible-Slide-Content>

Articulate. (2020, August 7). Storyline 3: How to design an accessible course.

<https://articulate.com/support/article/Storyline-3-How-to-Design-an-Accessible-Course>

Renub Research. (2019). *Online education market & global forecast, by end user, learning mode (self-paced, instructor led), technology, country, company*. Research and Markets.

<https://www.researchandmarkets.com/reports/4876815/online-education-market-and-global-forecast-by>

World Wide Web Consortium. (2008). *Web content accessibility guidelines 2.0*.

<https://www.w3.org/TR/WCAG20>

Appendix A: A Framework for Accessibility Testing Team (ATT) Involvement in the Production and Testing of Accessible eLearning Courses

Our testing experiences have helped us formulate a framework for improving the accessibility of online training while reducing rework and shortening our testing cycles. We have found a correlation between earlier ATT involvement and fewer issues identified during testing. We recommend that an ATT be involved in all phases of development to reduce the number of accessibility issues found during testing. The image below illustrates the six phases of the Agile Development Lifecycle for software applications and the activities of the ATT at each phase.

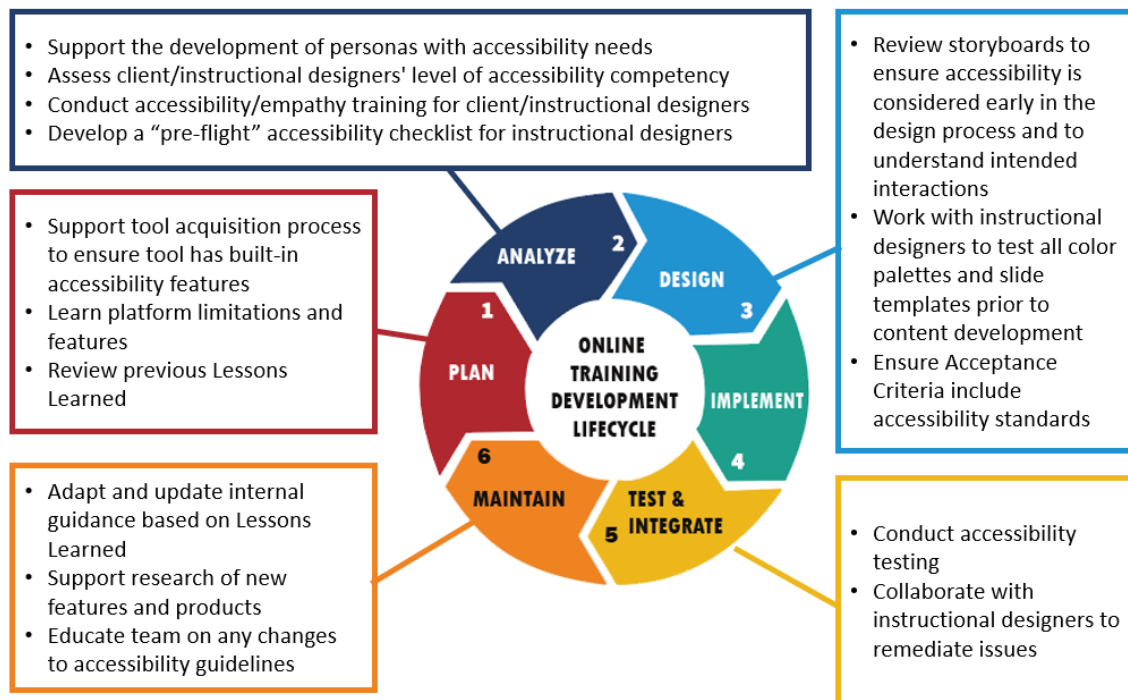


Figure 2. Agile development lifecycle

1. During the planning phase, the ATT supports the tool acquisition process to ensure the tool has built-in accessibility features, learns platform limitations and features, and reviews lessons learned for the client or project.
2. During the analyze phase, the ATT supports the development of personas with accessibility needs, assesses client/designer level of accessibility competency, conducts accessibility/empathy training for client/designers, and develops a "pre-flight" accessibility checklist for designers.
3. During the design/implementation phases, the ATT reviews storyboards to ensure accessibility is considered, works with designers to test all color palettes and slide templates, and ensures Acceptance Criteria include accessibility standards.
4. During the testing phase, the ATT conducts accessibility testing and works with designers to remediate issues.

Appendix B: A Template for Knowledge Capture

As a contracted testing team, our team uses Confluence, a knowledge management tool, to capture information about our testing activities for our clients so we can learn from prior testing engagements, provide consistent experiences to our clients, and store resources for our ATT members. We recommend that ATT members review Confluence when assigned a project, update it throughout the engagement as needed, and conduct a final update with lessons learned and links to deliverables such as test reports, job aids, and training presentations. For in-house testers, a similar template could be used to track project information or simply aggregate your organization's accessibility resources.

CLIENT NAME

Description

General information about the client and their mission, link to their website, and any background information that will help others on your team become familiar with the client.

Client Accessibility Resources

Descriptions and links to any accessibility resources within the client's organization. This might include an accessibility department, checklists, custom standards, or other guidance.

Client Accessibility Contacts

Provide names and information about any accessibility contacts or stakeholders within the client organization.

- Document their preferred contact methods and their information priorities.
- Note if they will be reviewing your test results so team members are aware of the need to collaborate.

Platform Information

Document the e-learning platform(s) the client uses.

- Known accessibility weaknesses (and strengths)
- Links to help documentation
- Links to training materials
- Technically proficient contacts with the client, your organization, or platform designer/developer

Previous Testing

Document your testing interactions as they occur.

- Document the date of testing
- Provide tester name and contact information
- Highlight major issues encountered
- Document lessons learned
- Link to resources developed (test reports, templates, training, presentations, etc.)

Appendix C: Storyboard Template

Designers often use storyboards to capture the intended slide text, captions, audio descriptions, images, image alternative text, slide branching, and navigation. The ATT can review storyboards before design begins and use the storyboard as a resource during testing to ensure all slides and branches are tested.

Screen Name	Content Reference
C1L1_010	N/A

Screen Text

Enter slide text in this box.

Audio

Has background music/sound

Describe audio contents in this box.

Production Elements

File Name	Type	Description	Alt Tag
GFN	Graphic	Logo	
GFN	Graphic	Standard background image	
C1L1_010.wav	Audio	Narrator	
C1L1_010.srt	Captions	Caption file for accessibility	

Notes/Instructions

Provide specific instructions in this box. This area could also include accessibility directions and notes for the designers.

Branching

On	Action
Next	C1L1_020
Back	Menu

Figure 3. Storyboard Template

Copyright notice



The 2020 ICT Accessibility Testing Symposium proceedings are distributed under the Creative Commons license: Attribution-NonCommercial-NoDerivatives 4.0 International (CC BY-NC-ND 4.0, <https://creativecommons.org/licenses/by-nc-nd/4.0/>).

You are free to: Share — copy and redistribute the material in any medium or format.

Under the following terms: Attribution — You must give appropriate credit, provide a link to the license, and indicate if changes were made. You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use. NonCommercial — You may not use the material for commercial purposes. NoDerivatives — If you remix, transform, or build upon the material, you may not distribute the modified material. No additional restrictions — You may not apply legal terms or technological measures that legally restrict others from doing anything the license permits.

Focus first: a new front-end approach

Claudio Luis Vera

Florida, USA
modulist@gmail.com

Marcelo Paiva, Sr.

UKG
Miami, FL USA
mp@cross.team

Marcello Paiva

Cross.team
Miami, FL USA
mop@cross.team

Keywords: UX Design, Design Systems, Front-end development, Accessibility, Guidance; Focus states.

Track/Theme: Testing approaches

Abstract

Shouldn't the element in focus be the first thing you see when looking at a screen? When we interact with websites, mobile apps, or desktop applications there is always an element that is waiting to receive input from us. This is what is referred to as focus in programming and user experience (UX) design. Focus can be as simple as a blinking cursor in a document — or more complex, as with a custom control. No matter what the component is, users need to be aware at all times of where focus lies. For people who can't use a pointing device, it's critically important to have a focus indicator that's always visible. If there's any gap in that continuity, then the user is left with no idea what interaction is expected of them. In software that manages heavy machinery or aircraft, for example, a gap in focus continuity could have catastrophic results. For all users, focus should be the one of most visible elements on the screen at any given moment. This is particularly true in a high-distraction environment where the user may have to look or step away from a task and then start up later where they left off. It's a perceptual task that is like finding a needle in a visual haystack. To cope with this challenge, most of us use hacks like wiggling the cursor or typing Tab / Shift-Tab to find the element in focus. But this adaptive behavior is hardly a desirable feature.

Primacy of focus

The element in focus should be the first thing a user sees when looking at a screen.

If our aim is the primacy of focus, then the current design standards for focus states are far too subtle. This paper proposes a set of new design techniques to fill this gap.

Design Tokens and Design Systems

Most major brands use a **design system** as their source of all digital materials related to their brand. A design system typically includes style guides, brand standards, pattern libraries, and component libraries among other assets. A well-architected system benefits designers and engineers alike by eliminating redundancies and rework by abstracting design decisions through reusable variables and components. The abstraction of colors becomes an important step for white-labeling and theme management purposes, this practice is referred to as **design tokens**.

In Brad Frost's hierarchy of [Atomic Design](https://bradfrost.com/blog/post/extending-atomic-design/), (see <https://bradfrost.com/blog/post/extending-atomic-design/>) design tokens carry the “sub-atomic” values of widely used design decisions, such as:

- Colors variables
- Font families, sizes, and weights
- Spacing utilities and other complementary design elements

Teams can use tokens to handle different levels of background colors for a given theme (e.g. *light, dark, high-contrast, brand1, brand2*, etc.), gaining greater scalability and flexibility.

Shifting left

A focus state is at least as important in a design system as other states like *normal, hover, disabled*, and *active*. Designers will define styles for the hover state and neglect to define styles for the focus state, not realizing that the hover state is inaccessible to many users with disabilities. If the focus state is not defined up front by the designer, someone else will implement it later as an afterthought, often “breaking” a carefully planned design through arbitrary choices.

If a design system is to achieve primacy of focus, its focus states need to be planned from the very beginning – that is, shifting to the far left of the design process.

A poor convention

For years, the major browsers have been representing focus visually by using a blue rectangle or a single-pixel dotted rule around the element in focus. This practice is poorly implemented by the browsers, and one can see the flaws in the box model when the focus element has any roundness to it. As a result, many designers loathe the focus styles and suppress them entirely through CSS in their themes and design systems.

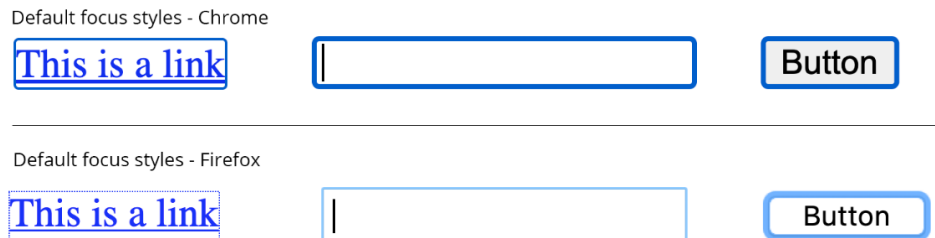


Figure 1 - Chrome and Firefox default focus styles for links, text fields, and buttons

Chromium is updating its default focus styles to a double black and white outline for increased visibility, which will likely frustrate designers further and fail to make the focus state the most visible element on a screen. However, there is no reason to settle for the browsers' default styles. In fact, the `:focus` pseudoclass should be able to draw from every available property in CSS.

Static design techniques

Focus color

Most design systems specify colors through design tokens for primary and secondary calls to action, error messages, alerts, and success states.

An ideal focus color should be one that works in harmony with the other brand colors — while remaining one of the brightest in the brand palette. For this reason, it's important to set aside at least one designated color for focus at the start of the planning process for any design token/system or art direction. A focus color that is chosen after the fact will have to be neutral or bland to avoid disrupting the brand palette's harmony.

With the upcoming [WCAG 2.2 guidelines](https://www.w3.org/TR/WCAG22/) (<https://www.w3.org/TR/WCAG22/>), the focus color will have to have at least 3:1 contrast with the adjoining colors, namely the background color and the element's background or border. If the background changes from white to a dark color, then a second value or token may be needed to maintain color contrast with the background.

To achieve primacy of focus, neon colors like lime green, safety orange, coral, or highlighter yellow are perfectly reasonable choices for focus indicators.

One technique for selecting focus color is to select the brightest existing color in the palette, and then select colors that vary by 60° on the color wheel (semi-triadic), or by 120° (triadic), or by 180° (complementary).

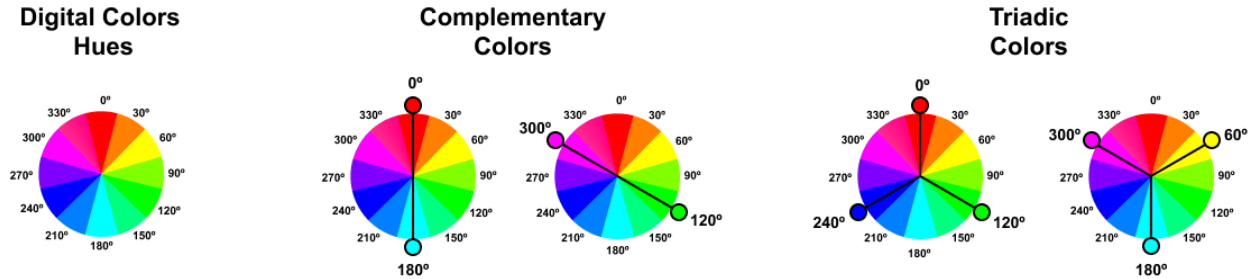


Figure 2: semi-triadic, triadic, and complementary colors

To avoid confusion, the focus color in a design system should not be the same as the color for the call to action or the error states in the user experience.

The focus color does not need to be limited in application to the `outline` property. It can also be applied to the element's `background-color`, to its `border`, to a `solid box-shadow`, or even text, just to name a few examples.

Focus width

If the design system uses an outline or border to convey focus, then it should have a of at least 4 CSS pixels (px), or enough to be the most visible element on the screen. To achieve primacy of focus, the width of the focus indicator should be greater when the focus color is less prominent.

By default, the web browsers' implementation of the box model displays the outline as a rectangle around the actionable element, regardless of any curvature that the element may have. This is especially visible with radio buttons and other elements that have a `border-radius` greater than zero. In these instances, the `box-shadow` property with a blur of zero should be used instead of `outline`, as it will follow the curvature of the element in focus.

Below is a table of recommended proportional widths in CSS for different actionable elements:

Small focus width elements

hyperlinks	.25 rem (4 px) <code>border-bottom</code>
radio buttons and checkboxes	minimum: .25 rem (4 px) <code>box-shadow</code> with 0 px blur ideal: at least $\frac{1}{4}$ the width of the element

Medium focus width elements

text fields, buttons, select	minimum: 4px <code>box-shadow</code> with 0 px blur ideal: $\frac{1}{4}$ of text field height
text areas	match the width used for the text fields

Large focus width elements

regions, landmarks, and other layout containers	minimum: .5 rem <code>box-shadow</code> with 0 px blur Ideal: .75 rem
cards	minimum: .5 rem <code>box-shadow</code> with 0 px blur Ideal: 1 rem

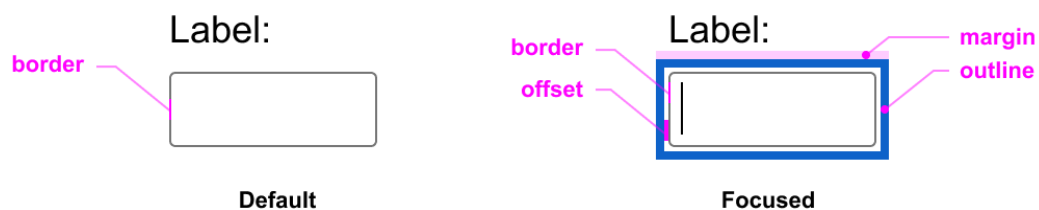


Figure 3: outlines, border, and offset in the box model

Offset

One technique to make the focus indicator more flexible is to include an offset of at least 2 px wide in a color that has the inverse of the focus color's brightness. This technique resembles the new Chromium double outline of black and white. The benefit is that a single focus style can be used for a light or dark background, or when the element lies over an image.

Introducing icons

“You can’t rely on color alone”, according to front-end accessibility guru Eric Bailey.

Because color-coding often fails for people with color vision deficiency, it needs to be combined with other ways of communicating information. One option may be to insert a dot or an icon signifying focus inside the element.

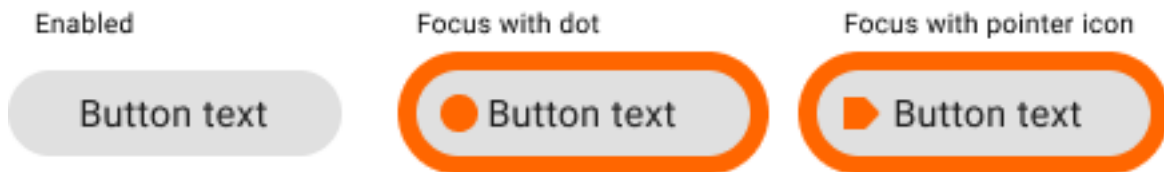


Figure 4 - Focus examples with dot and pointer icon.

Pointer shape or “sign here”

Occasionally, focus needs to be placed on the `<main>` element or on a block of text instead of on an interactive element. An example of this is after clicking on a “Skip to Content” link. In these cases, an outline may not be practical or visible on all sides. In these instances, the designer could insert a pointer icon before the first word. The icon could be a metaphor for the “sign here” stickers used on contracts in the physical world.

Motion design techniques

Our animal brains have difficulty finding objects when they’re still. It’s one thing to search for a needle in a haystack — it’s far easier if it actually moves. Like animal predators, we are naturally drawn to focus our attention on a small moving object in a static field, so motion design is a logical solution for focus states.

In motion design, microanimations and transitions are techniques that designers can use to direct the user’s eye to the element in focus. They can range from gentle repeating movements to more aggressive motions, depending on the audience’s tolerance and the brand’s personality.

In 1981, Disney animators Ollie Johnston and Frank Thomas published *The Illusion of Life: Disney Animation*, which detailed 12 basic principles of animation. Most of these principles can be applied to create microanimations as well: squash and stretch

- anticipation
- timing
- staging
- easing / slow in and slow outarc
- secondary action

- exaggeration and appeal
- follow through and overlapping action

(see https://medium.com/@ruthiran_b/disneys-motion-principles-in-designing-interface-animations-9ac7707a2b43)

Tying animations into events

There are three different points in which animation can be applied: when the element gains focus, when it loses focus, and after a certain amount of idle time. This can be mapped to two events and function as follows:

onFocus

This event applies when the element receives focus. This event can be used to guide the eye into a new style or context through animation.

onBlur

Can be a transition that is used when the element loses focus. It's best used as a transition that returns the element to its original state before it gained focus, and it can even be a reverse of the onFocus animation.

onIdle(time)

A timer function may be used to trigger an animation when the element in focus has been idle for a certain period of time, and where the idle time is passed as an argument. Idle animations should be very subtle, to avoid creating a needless sense of urgency for the user.

Timing and size of animations

The ideal length of an animation is between 200 and 500 milliseconds, according to UX expert [Val Head](https://valhead.com/2016/05/05/how-fast-should-your-ui-animations-be/), with a preference for being on the short side. (see <https://valhead.com/2016/05/05/how-fast-should-your-ui-animations-be/>) The human brain has difficulty perceiving animations shorter than 100 milliseconds as anything but instantaneous.

In his [in-depth article](https://uxdesign.cc/the-ultimate-guide-to-proper-use-of-animation-in-ux-10bd98614fa9) on the proper use of animation in UX, (<https://uxdesign.cc/the-ultimate-guide-to-proper-use-of-animation-in-ux-10bd98614fa9>) Taras Skytskyi explores other considerations: the relative size of the screen and the distance covered by the animation. As a rule of thumb, longer distances and larger screens require a little more time.

Consider vertigo

It's important to consider the amount of the screen that will be affected by a change of context: a full-screen zoom animation can trigger motion discomfort in people who have vestibular disorders. Apple had to retire zoom and parallax from iOS 7 for this very reason, and it introduced an OS-level setting to reduce animations in a later version of iOS.

A toggle to reduce or remove animations now exists in the iOS, Android, Mac OS, Windows, and Ubuntu. However, developers are largely unaware and don't make use of this setting.

In CSS, a media query exists for that should allow users to turn off unwanted animations — provided that developers have set up the animations to be conditional on that media query. Again, awareness and adoption in the developer community are very limited — in spite of widespread browser support.

Resources for animations

Fortunately, microanimations don't always have to be built from scratch. For example, [LottieFiles](https://lottiefiles.com) (<https://lottiefiles.com>) is a giant open-source library of contributed animations that work with the [Lottie](https://airbnb.design/lottie/) (<https://airbnb.design/lottie/>) React library developed by Airbnb Design. In this library, there are dozens of arrow and shape animations that would be suitable for focus.

The authors of this paper will also be releasing WTFocus, a focus-first web app that allows designers and developers to explore and try out focus states in CSS and javascript. Expect to see a beta release later in 2020.

Copyright notice



The 2020 ICT Accessibility Testing Symposium proceedings are distributed under the Creative Commons license: Attribution-NonCommercial-NoDerivatives 4.0 International (CC BY-NC-ND 4.0, <https://creativecommons.org/licenses/by-nc-nd/4.0/>).

You are free to: Share — copy and redistribute the material in any medium or format.

Under the following terms: Attribution — You must give appropriate credit, provide a link to the license, and indicate if changes were made. You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use. NonCommercial — You may not use the material for commercial purposes. NoDerivatives — If you remix, transform, or build upon the material, you may not distribute the modified material. No additional restrictions — You may not apply legal terms or technological measures that legally restrict others from doing anything the license permits.

The Importance of Switch Testing

Daman Wandke

Wandke Consulting
924 N. Garden St. #302 Bellingham, Washington 98225
daman@wandke.com

Keywords: Usability Testing; Functional Testing; Switch Technologies; Success Criteria; Switch Control; Switch Access; IT Accessibility; Limited Manual Dexterity; Equal Access.

Track/Theme: Testing Approaches.

Abstract

The importance of switch testing--people with limited manual dexterity fulfilling tasks while utilizing switch control--is often overlooked in the overall idea of usability testing. The Web Content Accessibility Guidelines (WCAG) focus on the operability of assistive technologies, where keyboard only accessibility is specifically mentioned, but WCAG does not include success criteria specific to switch technology use. This paper makes a case for incorporating switch technology testing into usability testing to ensure accessibility to everyone.

Introduction

Switch Control and Switch Access are metaphorical ramps to technology for individuals with limited manual dexterity. If technology is not compatible with switch platforms, it is rendered useless for these users. As switch technologies become more popular, conducting usability testing with switch users becomes more important; it exposes errors that keyboard testing may not. Switch Control and Switch Access testing can improve access and inclusion for people with limited speech or limited manual dexterity who cannot use a keyboard.

Background Information

Daman Wandke has Cerebral Palsy (CP); his CP affects his manual dexterity and speech, so he understands the importance of switch technology firsthand. He knows how switch technology allows himself, as well as others, to have equal access to technology. He also knows that compatibility of other technologies with switch technology is so important for that equal access to occur; without compatibility, the switch technology is rendered useless.

Daman also has over ten years of experience working in IT accessibility. He has been using switch technologies and conducting usability tests for clients since he found switch technology in 2014. Through conducting usability studies with switch devices and documenting remediation techniques for compatibility issues, he understands the importance of conducting usability testing with switch technology.

All About Switch

General Overview

Simply speaking, a switch device is how a person with limited manual dexterity can control technologies such as a power wheelchair, their phone, a computer, a smart home device, as well as many other technologies. There is a diverse range of ways that switch technologies can be implemented and different people have different preferences on how they utilize this technology.

The most common type of switch technology is a big round button that can be placed anywhere (e.g. near the user's hand, head, or foot, etc.) the user has the most control. The user can press this button to use the switch interface.

Some switch users can also use adaptive keyboards and/or voice commands. These technologies are more difficult to use when on the go. Large keyboards are larger and more cumbersome when using mobile devices. Voice control is difficult to use in some environments where talking isn't appropriate, privacy is needed or the environment is too loud. Switch also has the added capability of simulating a tap on the screen, which is not available via keyboard. As switch technology gains popularity on mobile, ensuring keyboard functionality may have less of an impact on the accessibility for people with limited manual dexterity.

How Switch Works

For an individual utilizing switch technology to navigate a screen, there is typically a focus indicator that moves throughout that screen showing where the user is located so that they can activate the task in which they would like to fulfill by selecting the lit-up focus indicator via the switch technology. There are also assistive technology devices that have more than one switch that can be programmed to do different tasks; for example, one switch navigates throughout the screen while the other switch selects the task in which the user wants to fulfill.

Types of Switch Technologies

Switch technologies have two components: hardware and software. The most basic hardware to interact with the switch interface is using your mobile devices' screen as your switch. Beyond using the device's screen, the next basic switch hardware is a one or two-button Bluetooth switch. The Bluetooth switch by AbleNet, which has two large buttons/switches, is an example of this. These large buttons are easier to interact with than a keyboard for users with limited manual dexterity. Each switch can be programmed to a switch function. For example, the most common functions are scan and tap. The scan function will scan through all the items on the screen using the switch interface's item mode. Once item mode places focus on the desired element, then the tap function will activate that element.



Figure 1: AbleNet Blue2 Bluetooth Switch

More complex switches are available depending on the user's needs. Someone that can only move their head can attach one or two switches to their wheelchair's headrest or use sip-and-puff technology to activate the switch. If the user can only move their leg, a switch device can be put by their knee. Stephen Hawking used a cheek switch to access his technology. There is a wide variety of switch devices available, making it highly customizable for users with significant physical disabilities.

Switch Interface Features

Switch hardware interacts with the switch interface built into an operating system. This interface is called Switch Control for iOS and MacOS and Switch Access for Android. Microsoft Windows does not have Switch Control but it has the Windows On-screen keyboard that is compatible with switch devices. Both Switch Control and Switch Access have two primary modes: item mode and point mode. Both platforms also have a scrolling feature. iOS also has more advanced gestures and allows program recipes to create automated tasks that are done repetitively.

The easiest way to use the switch interface across all platforms is item mode. Item mode scans through every element on the screen. The user can then activate the desired element. On iOS and MacOS, elements are grouped together, and the user can drill down to the desired elements. On Android, elements are not grouped. Item mode also scans through the keys on the on-screen keyboard, allowing users to type. When setting up your Bluetooth switch on Android, the on-screen keyboard option under Physical Keyboards needs to be enabled. Switch Access also works best with the Gboard on-screen keyboard.

When an element is not accessible via item mode, switch users can use point mode (iOS or MacOS) or point scan mode (Android) to simulate a tap on the screen. This feature allows the user to select the location on the screen they wish to tap using a coordinate-like system. The user first selects the horizontal location and then the vertical location that they wish to tap. Depending on the size of the element, selecting the location they wish to tap can be difficult and tedious for users with limited manual dexterity as utilizing the coordinate system requires more dexterity than item mode.

Even with optimal design, using switches is very tedious and for a switch user, many tasks take much longer to complete than when using a keyboard. Therefore, it is important to include switch users as a user group in usability testing to ensure that any barriers, especially tasks that are not accessible via item mode, are found and addressed for this user group.

Switch Testing

The Process

Using the screen as the switch can work great for some users because they do not have to carry around an external switch. When the screen is used as the switch, all the standard screen features are disabled. This makes using the screen as a switch difficult when performing accessibility testing. However, if you get to a barrier, you may want to use the screen to bypass the error. Using a Bluetooth keyboard is the easiest method to test using a switch. The tester can program specific keys on the keyboard to the desired switch functions (a key programmed as a switch controller function will not work as originally intended). However, for more realistic testing to represent a person who cannot utilize a keyboard, you can test with a one or two-button Bluetooth switch device.

Common Switch Accessibility Barriers

Keyboard-only navigation and switch technology are both used by people with limited manual dexterity. However, switch interfaces have marked differences. These differences create usability issues that may not happen in keyboard testing. One example that creates differences is when the elements that receive focus differ between using the switch versus the keyboard. A second example is the limitations that switch interfaces have with scrolling. A third example is that switch technologies use the operating system's Accessibility API, which creates differences when ARIA is used improperly.

The first example is when the elements that receive focus using switch technologies do not match keyboard focus. Some nonactionable elements receive focus to access the switch interface's menu for other functions (such as scrolling, point mode, etc.). The switch interface will place focus on what it determines to be significant elements (for example, div elements) even when these elements are not actionable via keyboard. This issue is common when a button is placed within a div element. One tab stop (the button) is actionable while the other tab stop (the div element) is not actionable. Changing the div elements to span elements will prevent the switch interface from placing two focus stops on these buttons. While the major differences between div and span elements are in their default display values, testing proved that placing "display: inline" on div elements will not prevent the element from receiving focus, while adjusting the HTML element would.

Another example has to do with scrolling. Switch's item mode scans through each element that is currently on screen. In order to scroll, the user must activate the scroll up or down feature. The amount of page the switch interface scrolls is consistent so users cannot attempt to use a combination of scrolling down, then up, to bring hidden items into view. Scrolling moves one screen at a time and the user cannot control the amount of screen that scrolls each time they activate the switch. This sometimes causes elements on a webpage to get stuck between two

scrolling panes and therefore, a user cannot access them. This issue is commonly found when floating menus are used. The switch interface is not aware that elements are hidden behind a floating menu.

A third example is that Switch Access uses the same Accessibility API as iOS VoiceOver and Android TalkBalk, which sometimes makes the switch behavior and keyboard functionality differ when ARIA is used improperly. For example, if a form field has the attribute `aria-hidden="true"`, a switch user will not be able to place focus in the form field because the form field is hidden from the Accessibility API. A sighted keyboard-only user would not have this issue.

The above examples are just a few of the issues that are not addressed by sighted-keyboard only testing standards. Before switch technologies, it made sense that there were only sighted-keyboard only standards; however, people with limited manual dexterity could only access their mobile device using a large keyboard which is difficult to transport. Now that switch technologies are available, a new standard for accessibility has been created. When only complying with sighted-keyboard only testing standards, barriers that would be encountered by a switch technology user will not be found. Switch testing is needed to ensure people who use these technologies have equal access to websites and applications through their mobile devices.

Conclusion

When it comes to accessibility and compliance with the law, people with disabilities have adopted the phrase, “nothing about us, without us!” as they desire to be part of the conversation. Society needs to understand that “[a]ccessibility is not only about meeting standards (Henry) as accessibility is not “one-size-fits-all”. Switch technologies are necessary to provide equal access to people with limited manual dexterity. However, switch technologies cannot provide access to websites and mobile applications that are not designed with switch compatibility in mind. Usability testing by users with limited manual dexterity ensures accessibility with switch technologies. This usability testing will not only benefit the disability community but organizations as a whole. Increased inclusivity increases the bottom line.

References

Henry, S. (2019). The Missing Link: Accessibility and Usability Working Together. In Proceedings of the 4th annual ICT Accessibility Testing Symposium, 3.

Sethfors, H. (2018). Assistive Technologies – The Switch. Axess Lab. Retrieved from <https://axesslab.com/switches/>

Copyright notice



The 2020 ICT Accessibility Testing Symposium proceedings are distributed under the Creative Commons license: Attribution-NonCommercial-NoDerivatives 4.0 International (CC BY-NC-ND 4.0, <https://creativecommons.org/licenses/by-nc-nd/4.0/>).

You are free to: Share — copy and redistribute the material in any medium or format.

Under the following terms: Attribution — You must give appropriate credit, provide a link to the license, and indicate if changes were made. You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use. NonCommercial — You may not use the material for commercial purposes. NoDerivatives — If you remix, transform, or build upon the material, you may not distribute the modified material. No additional restrictions — You may not apply legal terms or technological measures that legally restrict others from doing anything the license permits.

Using Personas for Accessible Design

Kathryn Weber-Hottleman

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

Keywords: Personas, scenarios, web development, web design

Track/Theme: Governance and implementation of testing in the systems development life cycle

Abstract

There are a few steps that developers and designers can take to incorporate accessibility from the beginning of a solution's design. One way of doing this is to create additional personas that are users with disabilities. Also, having a primary persona with a disability means that disability will never be far from developers' and designers' minds. Another way to keep accessibility present is to include it in scenario design. There are multiple benefits to including characteristics of users with different disabilities and their needs in persona and scenario development. First, it saves time and money that would otherwise be spent remediating accessibility issues. It makes the solution available to all demographics, not inadvertently losing the users with disabilities market. It also aligns with organizational values of diversity and inclusion. Finally, it minimizes legal risk.

What is UX and why is it important?

UX, or user experience, can be defined as how people feel when they use a product or service (Interaction Design Foundation, 2018). It is important to begin thinking about UX early in the development and design process, to shift UX as close to the beginning of the workflow as possible, because the desired user experience outcome can shape the project's direction. One aspect of UX is considering the end user's needs and finding solutions to meet those needs. Meeting the user's needs can influence the user's expectations of the project's results, which in turn shapes the project's trajectory and outcome.

Consider a pen. These devices have been in use for hundreds of years. They were created so that people could have a means of expressing themselves permanently, through writing and art. The user need, in this case, was to be able to set down thoughts in a lasting, consistent way. The user experience, however, has led to the pen's constant improvement, from quill pens which were dipped in ink, to fountain pens, to modern rollerball pens. Dissatisfaction with the user experience created a user need for an improved pen.

What is design thinking?

Innovation as a result of user need is the goal of design thinking. Design thinking is one framework for understanding the user experience and can be defined as:

...an iterative process in which we seek to understand the user, challenge assumptions, and redefine problems in an attempt to identify alternative strategies and solutions that might not be instantly apparent with our initial level of understanding. At the same time, design thinking provides a solution-based approach to solving problems. It is a way of thinking and working as well as a collection of hands-on methods. (Interaction Design Foundation, 2018)

The Hasso-Plattner Institute of Design at Stanford has developed five phases of design thinking, which include empathizing with users, defining users' needs and problems along with the developer's insights, ideation by challenging assumptions and creating innovative strategies, prototyping solutions, and testing solutions (Interaction Design Foundation, 2018). Iterating through these five phases incorporates user-oriented thinking early in the development and design phase of the project. In fact, the first three phases focus primarily on the user and his or her needs. It is not until late in phase three that strategies to meet the user's needs are considered. Tim Brown suggests that "design thinking is firmly based on generating a holistic and emphatic understanding of the problems that people face, and that it involves ambiguous or inherently subjective concepts such as emotions, needs, motivations, and drivers of behaviors" (Interaction Design Foundation, 2018). A key to design thinking is spending time getting to know the user. This includes understanding how the user is likely to use the solution and requires the designer to think of creative and innovative uses for the solution.

Consider the pencil. Where the pen's purpose is duration, the pencil's purpose is practice. Putting this in the framework of design thinking, empathizing with the user means acknowledging the frustration that comes with the inability to redact mistakes when using a pen. The user's need, in this case, is a writing utensil with the capacity to be erased from a slate or paper, thus cutting down on paper waste. Challenging the assumption that the pen is the only or perhaps best writing utensil and proposing the pencil as an innovative solution is the third step in design thinking. Prototypes began with lead sticks and progressed towards the modern graphite pencil, even advancing towards a mechanical pencil. Iterating through prototypes and testing those prototypes has led to the current pencil.

Accessibility and the User Experience

"Accessible" means that individuals with disabilities are able to independently acquire the same information, engage in the same interactions, and enjoy the same services within the same timeframe as individuals without disabilities, with substantially equivalent ease of use ("ICT Policy Procedures", 2019). Disability, according to the Americans with Disabilities Act As Amended (ADA), means a physical or mental impairment that substantially limits one or more major life activities ("The Americans with Disabilities Act ", 2017). Design thinking, therefore, needs to take into account the needs of users with disabilities; without this aspect, analysis of user experiences will be missing an entire user demographic.

Consider the pencil again. How will a user without use of their arms or legs be able to write using a standard pencil? The answer is quite simple: they cannot. Design thinking needs to revise the first step of empathizing with the user. Suddenly the user profile is much broader than the average user who has full ability in his or her limbs. This means that the user needs have dramatically changed from needing a writing utensil that can be erased to a writing utensil that can be used without the use of limbs. Challenging the assumption that the pencil is one of the only writing utensils available forces developers and designers to think outside the box and begin to consider how a person who does not have the use of their limbs may write independently. Innovative solutions include a word processor and speech-to-text technology, where users can speak, and the word processor transcribes their words. Another innovative solution is a touch screen device, where a user can use a mouth stick or other input device to type out words. Prototypes include basic text editors and keyboard-based software and iterating through these has led to the development of complex word processors, assistive technology that recognizes speech and converts it to text, and touch screen devices.

UX, Personas, and Users with Disabilities

One way that accessibility can be shifted into the development and design phase is through personas. Personas are fictional representations of actual users, and they are applied in the early stages of product development (Segue Creative Team, 2016). They are used to bring common user needs to the forefront and are highly detailed, providing as much information as possible about the fictional user and how he or she may experience the solution. They can be based on actual user interviews or they can be created based on the solution's goals or vision (Segue Creative Team, 2016). Often, they are used as meaningful archetypes, answering the question of how a given user would respond to a given scenario (Dam & Siang, 2019). Personas are a foundational point of design thinking because they provide concrete points with which the developer can empathize, and express users' needs based on reactions to scenarios. Personas allow developers to challenge assumptions about how the solution will be used and assumptions about the users themselves, which then leads to innovations in the product's design.

Accessibility can be integrated into the design process by developing personas who are users with disabilities. There are many different types of disabilities, with invisible disabilities (e.g. learning disabilities, ADHD, and mental health disorders) being the highest population. Thus, the development of personas with disabilities necessitates basic knowledge of disabilities and understanding of how a disability can impact a user. It also requires some knowledge of assistive technologies and how they can support users with disabilities. For example, a blind screen reader user would need all images to have alternative text, while a keyboard-only user would need all elements to receive focus in the tab order.

It is common to overlook disability entirely when evaluating user demographics, because disability does not come to the forefront when developing many products. Furthermore, if developers and designers are aware of the extent of users with disabilities, often there is lack of understanding or misunderstanding of how disabilities can impact the user. Without understanding how different disabilities can impact users, it is impossible to develop an accurate sense of user needs and, from there, to develop realistic personas. Keep in mind, though, that every individual with a disability experiences it differently, and personas can only approximate a

user's experience. Also, consider that disability is only one aspect of a user; therefore, it should be allotted the same weight as the persona's other characteristics. It is a challenge to keep disability in perspective, because it so often bars users from being able to access even the beginning stages of solutions. The inclination can be to weight disability needs more heavily than other user needs, as their impacts feel far-reaching. However, neglecting other user needs in order to provide access to users with disabilities undermines the solution's efficacy. This makes for a complex balance, providing for most user needs while also addressing the needs of users with disabilities, which may be only a small slice of users. It is easy to skew the balance towards either set of needs but neglecting either side weakens the solution's potential. Though this balance may seem difficult, it prompts innovation and forces developers and designers to consider alternative ways to achieve the product's goals and vision.

The Accessibility Process

There are a few steps that developers and designers can take to incorporate accessibility from the beginning of a solution's design. One way of doing this is to create additional personas that are users with disabilities. This may mean expanding the typical number of personas, as the standard four to seven personas will be unable to capture the varied impacts of disabilities on users. It is also advisable to make the primary persona a user with a disability (Dam & Siang, 2019). Having a primary persona with a disability means that disability will never be far from developers' and designers' minds, raising accessibility components' priority in the overall design process.

Another way to keep accessibility present is to include it in scenario design. Through scenarios, developers and designers can anticipate how a user with a disability might respond to the solution under given circumstances. This might be a different reaction than a persona without a disability, because the user with a disability may not even be able to initially access a solution, if the solution is inaccessible. This forces developers and designers to consider different types of input, including keyboard-only access and assistive technologies like alternative mice and screen readers. Developers and designers may want to include scenarios where the persona with a disability needs to use assistive technology, which can help discern design components' efficacy. For example, a screen reader user's persona can help developers and designers check fieldsets and IDs in a form.

There are multiple benefits to including characteristics of users with different disabilities and their needs in persona and scenario development. First, it saves time and money that would otherwise be spent remediating accessibility issues. It makes the solution available to all demographics, not inadvertently losing the users with disabilities market, which pervades all ages, sexes, etc. It also aligns with organizational values of diversity and inclusion. Finally, it minimizes legal risk.

The Benefits of Designing with Accessibility in Mind

Minimizing legal risk is often the primary reason that accessibility is included in early development and design stages. It is insufficient for only quality assurance teams and company lawyers to know the laws surrounding accessibility; developers and designers also need to have a basic understanding of the laws in order to incorporate accessibility into early designs and prototypes. Understanding the laws leads to understanding obligations regarding accessibility and minimizing liability.

The key laws that govern accessibility federally in the United States are the Americans with Disabilities Act As Amended (ADA) and the Rehabilitation Act of 1973. While the ADA does not provide specific guidance on digital accessibility, it does prohibit discrimination and ensures equal opportunities for persons with disabilities in employment, state and local government services, public accommodations, commercial facilities, and transportation ("Policies", 2016). Websites and much of the current digital landscape are considered places of public accommodation, which means they are required by law to be accessible. The ADA does not offer guidelines for creating accessible content.

The Rehabilitation Act of 1973 has two sections, Section 504 and Section 508, that both relate to accessibility. Section 504 is a federal law designed to protect the rights of individuals with disabilities in programs and activities that receive Federal financial assistance from the U.S. Department of Education. Section 504 provides: "No otherwise qualified individual with a disability in the United States . . . shall, solely by reason of her or his disability, be excluded from the participation in, be denied the benefits of, or be subjected to discrimination under any program or activity receiving Federal financial assistance" The Office of Civil Rights enforces Section 504 in programs and activities that receive Federal financial assistance from the Department of Education. Recipients of this Federal financial assistance include public school districts, institutions of higher education, and other state and local education agencies ("Policies", 2016). Section 504 is similar to the ADA in that it prohibits discrimination and casts a wide net of responsible parties.

Section 508, refreshed in 2018, is a set of standards that requires Federal agencies to make information and communication technology (ICT) accessible to employees and members of the public who have disabilities in a comparable manner to the access experienced by employees and members of the public without disabilities. The revised Section 508 Standards apply to ICT that is "procured, developed, maintained, or used" by agencies of the Federal government. Section 508 was enacted to eliminate barriers to ICT, make opportunities available for persons with disabilities, and encourage development of technologies that will help achieve these goals ("Section 508", 2019). One of the benefits about Section 508 is that it specifies the Web Content Accessibility Guidelines (WCAG) version 2.0, level AAA, as the standard to which Federal agencies are held. Note that this section is intended for Federal offices only.

Case precedent, however, shows that Federal agencies are not the only offices subject to WCAG 2.0's rigor. Cases such as UC Berkeley (2016), Harvard and MIT (2016), and the Miami University of Ohio (2016) show in their resolution agreements requirements that the universities uphold WCAG 2.0, level AA. WCAG 2.0 has three levels, A, AA, and AAA. Federal offices are held to level AAA, while the majority of resolution agreements only specify that institutions

meet level AA. It is also clear from recent cases, such as *Gil v. Winn-Dixie* and *Robles v. Domino's Pizza*, that organizations beyond higher education are being held to similar accessibility standards. Therefore, it is in the best interest of organizations to proactively create solutions that comply with WCAG 2.0, level AA at the least. Accessibility needs to be treated as an inherent component of a solution, similar to how a bug would be treated, as opposed to being treated as an optional feature (Urban, 2018).

An additional benefit to designing with accessibility in mind is that it saves both time and money. Bolting on accessibility features after the design phase typically takes either considerable time or great expense. Developers and designers will need to spend additional hours remediating the new solution's design in order for accessibility components to work smoothly with the solution's other components. Alternatively, external consultants can be hired to remediate the solution, which can make the design costs skyrocket. Including accessibility from the starting point may appear to cost time and money at the outset, but it ends up saving both in the long run.

Not only does initially including accessibility save money, but it can also make money. By creating solutions that are inherently accessible, companies open up their solutions to the users with disabilities market. As previously mentioned, this market is considerable, spanning all demographics. According to the Center for Disease Control, as many as 26% of Americans self-identify as having a disability, which makes users with disabilities the largest minority group in the United States (CDC, 2019). Because of this vast market, it is in a company's best financial interests to design products that are accessible out of the box.

Another benefit to including accessibility is less tangible but equally as important: aligning with organizational values of diversity and inclusion. Because there are so many individuals who identify as having a disability, it is highly likely that users with disabilities can be found within the company producing a solution. Building accessibility into a product publicly demonstrates a commitment to diversity of ability, which is encouraging to employees with disabilities. Also, companies that produce inaccessible solutions may inadvertently be limiting their employee candidate pool: it is difficult for a user with a disability to work on a solution that is inaccessible to them.

Conclusion

Overall, designing with accessibility in mind saves time and money and broadens a solution's user base. Not only is accessibility legally required, it is also financially sound and accords with diversity and inclusion values. In conclusion, it is this author's recommendation to shift accessibility as close to the beginning of the design process as possible. This can be done using personas and scenarios in which users may access a product through alternative means. Using personas and scenarios of users with disabilities leads to stronger product development, because it forces developers and designers to innovate accessibility solutions, thus creating a more robust product with a broader vision. Accessibility involves everyone engaged in a solution's design; it cannot be left up to quality assurance to suggest bolting on accessibility features. Involving the entire solution team in the accessibility process creates solutions that are elegant, effective, and available to all users.

References

Dam, R., & Siang, T. (n.d.). Personas – A Simple Introduction. Retrieved November 7, 2019, from <https://www.interaction-design.org/literature/article/personas-why-and-how-you-should-use-them>.

Disability Impacts All of Us Infographic. (2019, September 9). Retrieved November 7, 2019, from <https://www.cdc.gov/ncbddd/disabilityandhealth/infographic-disability-impacts-all.html>.

ICT Policy Procedures. (2019, August 20). Retrieved November 7, 2019, from <https://accessibility.its.uconn.edu/ict-policy-procedures/>.

Interaction Design Foundation. (n.d.). The Basics of User Experience Design. doi: <https://www.interaction-design.org/ebook>

Policies. (2016, November 29). Retrieved November 7, 2019, from <https://accessibility.uconn.edu/policies/>.

Section 508. (2019, January 1). Retrieved November 7, 2019, from <https://section508.gov/>.

The Americans with Disabilities Act, The Americans with Disabilities Act (2017). Retrieved from https://www.ada.gov/regs2010/titleIII_2010/titleIII_2010_regulations.htm

Urban, M. (2018, September). Incorporation of Accessibility into the It Lifecycle: Identifying the Needs Users with Disabilities, Best Practices, and Agile Development. Incorporation of Accessibility into the IT Lifecycle: Identifying the Needs Users with Disabilities, Best Practices, and Agile Development.

What is Persona Development and Why is it Important? (2019, October 6). Retrieved November 7, 2019, from <https://www.seguetech.com/persona-development-important/>.

Copyright notice



The 2020 ICT Accessibility Testing Symposium proceedings are distributed under the Creative Commons license: Attribution-NonCommercial-NoDerivatives 4.0 International (CC BY-NC-ND 4.0, <https://creativecommons.org/licenses/by-nc-nd/4.0/>).

You are free to: Share — copy and redistribute the material in any medium or format.

Under the following terms: Attribution — You must give appropriate credit, provide a link to the license, and indicate if changes were made. You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use. NonCommercial — You may not use the material for commercial purposes. NoDerivatives — If you remix, transform, or build upon the material, you may not distribute the modified material. No additional restrictions — You may not apply legal terms or technological measures that legally restrict others from doing anything the license permits.

Social Media Accessibility Testing

Gian Wild

AccessibilityOz
Melbourne, Victoria, Australia
gian@accessibilityoz.com

Keywords: Social; Social Media; Testing Guidelines.

Track/Theme: Other: Social Media.

Abstract

Social media is an incredibly important tool in modern society. Social media testing should be part of any WCAG2 testing. Gian Wild covers why social media needs to be tested and what should be tested when evaluating the compliance of a site. Requirements include using social media accessibility features, such as alt attributes, captions etc., as well as providing alternatives for users who find social media impossible to use.

Why does social media need testing?

Social media is everywhere. According to Kietzmann, Hermkens, McCarthy and Silvestre (J.H. Kietzmann, May - June 2011) there are five main reasons for social media usage:

- Personal – examples include creating online commentary of personal thoughts, sharing photos, discussing social events
- Work – examples include creating a resume, making professional connections, discussing job opportunities
- Entertainment – examples include sharing videos, following discussion from celebrities, discussing strategies in online games
- Provision of goods and services – examples include sharing information about a product, responding to user feedback about a product, accessing government services
- Education – examples include participation in online classes, sharing learning resources

It is not just young people who use social media. Ninety percent of people aged 18 to 29 years use social media; more than 80% of people aged 30 to 49 years; almost 70% of people aged 50 to 64 years and 40% of people aged 60 years and older (Pew Research Center, 2019). There are more than 3.81 billion active social media users (Clement, 2020). On average, people have 8.3 social media accounts (Cooper, 2020). The average time a user spends on social media each day is more than three hours (Gilsenan, 2019).

What needs to be tested?

Over several years of testing the accessibility of social media, Gian Wild has devised a checklist of items that should be tested to ensure social media is as accessible as possible to people with disabilities.

Organization contact information is available on the social media Account or About page

It is important that organization contact information is available on the social media Account or About page. Each social media network offers different levels of detail. For example, in Twitter, as shown in Figure 1, a short description is provided with a location and URL.



Figure 1 - Stanford University's Twitter page

In Facebook, detailed information can be provided, such as an address, map, type of organization, opening hours, busy times, phone number and other locations. See Figure 2 for an example of the Facebook About page for the World Health Organization.

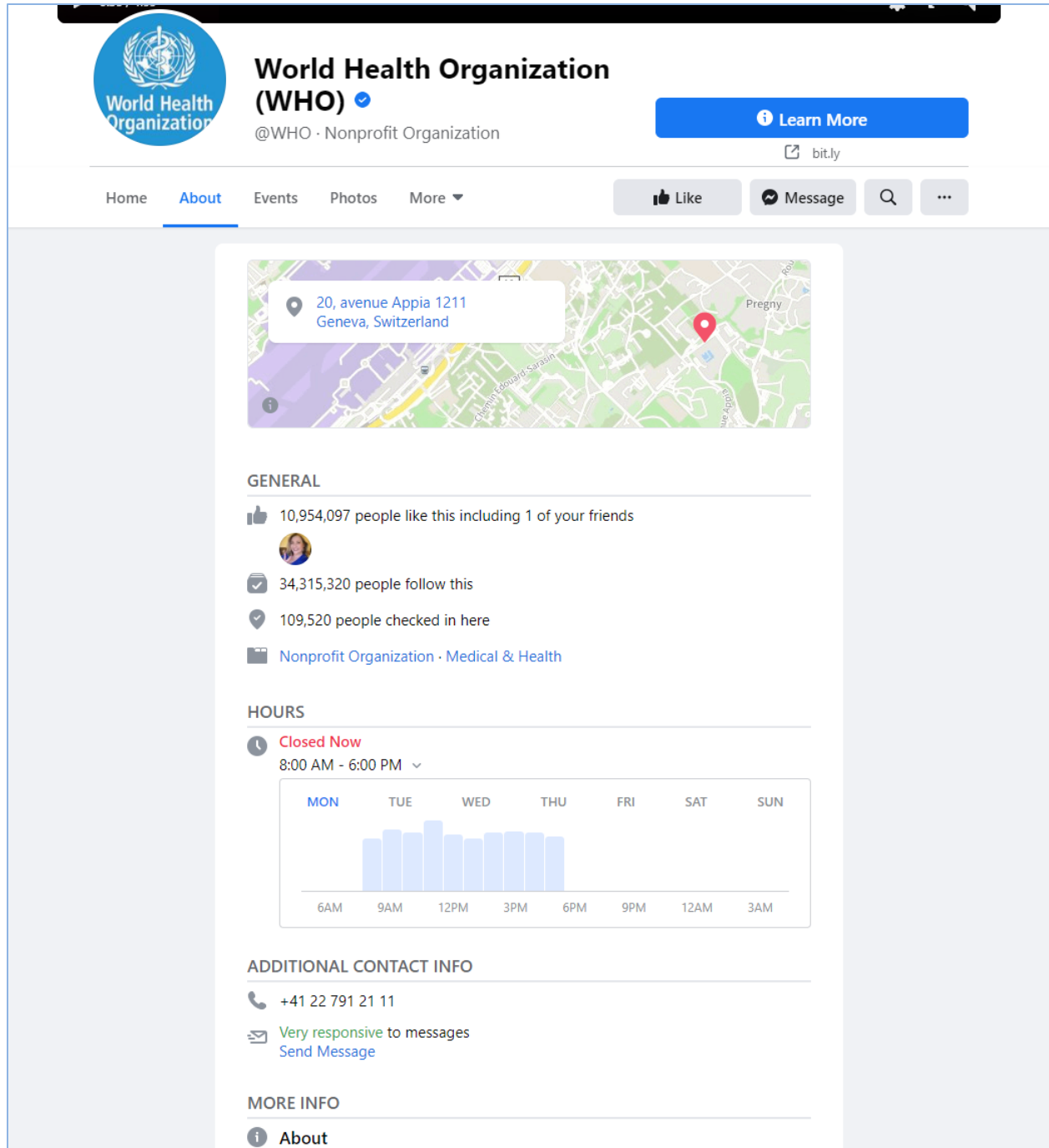


Figure 2 - Facebook About page for the World Health Organization

Content on social media is posted to multiple social media outlets

Not all social media networks are accessible to all people. Some groups will prefer one social media network over another; and some users will not be able to use a social media network at all. Therefore, it is important that the same content is posted to multiple social media networks to ensure all users can access the content.

Alternative apps are provided for social media networks

In some cases, the social media network is completely inaccessible to users. In many cases, an alternative interface or app has been built that provides additional accessibility features. However, it cannot be assumed that users know about these alternatives; therefore, providing links to these alternative apps or interfaces is recommended. In Figure 3, links have been provided to alternative apps to Twitter, such as Plume for Android and Easy Chirp.

Other Tips

For some, using the mobile version of Twitter is a more accessible alternative. It is more basic in function and layout and makes navigation easier.

There are also many apps for Twitter that can yield a more accessible experience. [Plume for Android](#) devices and [Easy Chirp: an accessible alternative to Twitter](#). Easy Chirp operates in a similar way to the main Twitter website but provides a more consistent layout, good keyboard navigation, better support for assistive technologies such as screen readers. It also allows users to add alternate text and long descriptions to images.

Figure 3 - Links to alternative apps for Twitter

Alternative apps for Facebook

- [The mobile view of Facebook](https://m.facebook.com/) (<https://m.facebook.com/>)
- [Metal for Android](#)
(<https://play.google.com/store/apps/details?id=com.nam.fbwrapper&hl=en>)

Alternative apps for LinkedIn

- [LinkedIn Lite for Android](#)
(<https://play.google.com/store/apps/details?id=com.linkedin.android.lite&hl=en>)

Alternative apps for Twitter

- [EasyChirp](http://www.easychirp.com/) (<http://www.easychirp.com/>)
- [Twitterrific on Mac](https://twitterrific.com/mac) (<https://twitterrific.com/mac>)
- [Twitterrific on iOS](https://twitterrific.com/ios) (<https://twitterrific.com/ios>)
- [Metal for Android](#)
(<https://play.google.com/store/apps/details?id=com.nam.fbwrapper&hl=en>)
- [Plume for Android](#)
(https://play.google.com/store/apps/details?id=com.levelup.touiteur&feature=nav_result)
- [TwInbox \(add-in for Microsoft Outlook\)](#)
(https://www.techhit.com/TwInbox/twitter_plugin_outlook.html)

Alternative apps for YouTube

- [Accessibility for YouTube Android mobile app](#)
(<https://support.google.com/youtube/answer/6087602?hl=en&co=GENIE.Platform=Android>)
- [YouDescribe for desktop or iOS](https://youdescribe.org/) (<https://youdescribe.org/>) - access to audio described YouTube videos

Links to social media accessibility tips are provided on social media networks

Just as users may not be aware of the alternatives to social media networks, users often do not know the accessibility tips about a particular social media network. Sometimes these tips are buried in blogs and not easily found. In Figure 4, there is a link provided to the page [Using YouTube with a screen reader](https://support.google.com/youtube/answer/189278?hl=en) (<https://support.google.com/youtube/answer/189278?hl=en>).

Tips to improve YouTube Accessibility

When using a Screen Reader

YouTube offers several versions of its embeddable player, including an HTML 5 version and a Flash version. The HTML 5 player is fully accessible but the Flash version is not so screen readers may have difficulty using the player interface due to lack of keyboard shortcuts. A way around this to use an alternate viewing portal that has accessible player controls.

YouTube also provides instructions on [Using YouTube with a screen reader](https://support.google.com/youtube/answer/189278?hl=en).

Figure 4 - YouTube accessibility when using a screen reader

Accessibility tips for common social media networks

- [Facebook Accessibility Page](http://www.facebook.com/accessibility) (<http://www.facebook.com/accessibility>)
- [Facebook Accessibility Tips](http://www.facebook.com/help/141636465971794) (<http://www.facebook.com/help/141636465971794>)
- [Facebook keyboard shortcuts](http://www.facebook.com/help/156151771119453/) (<http://www.facebook.com/help/156151771119453/>)
- [LinkedIn Accessibility](https://www.linkedin.com/accessibility) (<https://www.linkedin.com/accessibility>)
- [Accessibility Tips on Twitter](https://twitter.com/alltips?lang=en) (<https://twitter.com/alltips?lang=en>)
- [Which screen reader should I use for Facebook?](https://www.facebook.com/help/485531161504971) (<https://www.facebook.com/help/485531161504971>)
- [How do I use a screen reader for Instagram?](https://help.instagram.com/1178723545597542) (<https://help.instagram.com/1178723545597542>)
- [Use YouTube with a screen reader](https://support.google.com/youtube/answer/189278?hl=en) (<https://support.google.com/youtube/answer/189278?hl=en>)

Accessible terminology and features have been used on social media networks

It is important—and a tenet of the W3C Web Content Accessibility Guidelines—that any accessibility features inherent in a system must be utilized. Different social media networks have different accessibility features. Most of them enable the user to turn off auto-playing of videos, and some allow the addition of alternative text for images or captions for videos.

For detailed instructions on how to turn off auto-play on each social media network, see the AccessibilityOz article [Social Media Accessibility: Turning Off Autoplay](https://www.accessibilityoz.com/2019/11/social-media-accessibility-turning-off-autoplay/) (<https://www.accessibilityoz.com/2019/11/social-media-accessibility-turning-off-autoplay/>).

As of September 2020, the following accessibility features are available. For detailed instructions on how to use these accessibility features, see the AccessibilityOz article [Social Media Accessibility Features](https://www.accessibilityoz.com/2019/11/social-media-accessibility-features/) (<https://www.accessibilityoz.com/2019/11/social-media-accessibility-features/>).

Facebook accessibility features

- Alternative text for images
- Captions for videos
- It is recommended that you load videos to YouTube and add captions there, and then add to Facebook

Instagram accessibility features

- Alternative text for images
- It is recommended that you burn captions into any videos

LinkedIn accessibility features

- Alternative text for images (desktop only)

Pinterest accessibility features

- Alternative text for images
- It is recommended that you burn captions into any videos

Twitter accessibility features

- Alternative text for images
- Use camel case (e.g. #AccessibilityTestingForum)

YouTube accessibility features

- Captions for videos
- It is recommended that you host the video on YouTube and play in an accessible player on your web site such as OzPlayer (by AccessibilityOz) or AblePlayer
- It is recommended that you link to a transcript

Social media feed is replicated on the site and/or provided through daily, weekly or monthly email digests

In some cases, users will not be able to access any social media networks; in this case, it is important to replicate the feed on the site or provide a newsletter-style option. In Figure 5, the AccessibilityOz tweets are replicated on the site.



The screenshot shows a webpage layout with two main columns. The left column is titled 'News' and contains two articles. The first article is titled 'Keeping busy: webinars and conferences galore!' and discusses CEO Gian Wild's presentation at UX Australia and a podcast interview. The second article is titled 'Accessibility webinar series from AccessibilityOz & CommonLook!' and mentions a collaboration with CommonLook. The right column is titled 'Tweets @accessibilityoz' and displays three tweets. The first two are retweets of a tweet from @id24conf about an Inclusive Design 24 talk by @gian. The third tweet is a promotional message about recorded webinars and upcoming content.

News

Keeping busy: webinars and conferences galore!

What have we been up to? Our CEO Gian Wild presented on Mobile Accessibility Testing at UX Australia earlier this month. She was also interviewed about the subject on the UXA Podcast (recording available via SoundCloud). Did you miss the accessibility webinar series we ran with CommonLook? The recordings are now available on demand: Using

Accessibility webinar series from AccessibilityOz & CommonLook!

AccessibilityOz is teaming up with CommonLook to provide a series of free

Tweets @accessibilityoz

RT @id24conf: #id24 On 17 September @ 5:00 UTC, our sixth Inclusive Design 24 talk is by @gian: <https://t.co/TeA8Tfyqbe> <https://t.co/ZcC28C...> - 2 hours ago

RT @id24conf: #id24 On 17 September @ 2:00 UTC, our third Inclusive Design 24 talk is by @llegal, @accessaces and @ohmydeity: <https://t.co...> - 8 hours ago

What have we been up to, and what are we doing next? We've got recorded webinars for you to check out, and upcoming... <https://t.co/8hc9KeD8ut> - 2 weeks ago

Figure 5 - The @accessibilityoz Twitter feed replicated on the AccessibilityOz homepage

Social media icons are consistent on the site

Consistency is an integral part of the W3C Web Content Accessibility Guidelines. Therefore, it is important that social media icons are consistent throughout a web site. If not, the user may infer that the different icons link to different accounts.

Conclusion

In conclusion, it is recommended that the following items are tested, in addition to testing a web site for accessibility compliance:

- Organization contact information is available on the social media Account or About page
- Content on social media is posted to multiple social media outlets
- Alternative apps are provided for social media networks
- Links to social media accessibility tips are provided on social media networks
- Accessible terminology and features have been used on social media networks
- Social media feed is replicated on the site and/or provided through daily, weekly or monthly email digests
- Social media icons are consistent on the site

References

- Clement, J. (2020, May 18). Social media - Statistics & Facts. Retrieved from Statista: <https://www.statista.com/topics/1164/social-networks/>
- Cooper, P. (2020, February 20). 140+ Social Media Statistics that Matter to Marketers in 2020. Retrieved from Hootuite: <https://blog.hootsuite.com/social-media-statistics-for-social-media-managers/>
- Gilsenan, K. (2019, September 27). 2019 in Review: Social Media is Changing, and It's Not a Bad Thing. Retrieved from Global Web Index: <https://blog.globalwebindex.com/trends/2019-in-review-social-media/>
- J.H. Kietzmann, K. H. (May - June 2011). Social media? Get serious! Understanding the functional building blocks of social media. *Business Horizons*, Pages 241-251.
- Pew Research Center. (2019, June 12). Social Media Fact Sheet. Retrieved from Pew Research Center: Internet and Technology: <https://www.pewresearch.org/internet/fact-sheet/social-media/>

Copyright notice



The 2020 ICT Accessibility Testing Symposium proceedings are distributed under the Creative Commons license: Attribution-NonCommercial-NoDerivatives 4.0 International (CC BY-NC-ND 4.0, <https://creativecommons.org/licenses/by-nc-nd/4.0/>).

You are free to: Share — copy and redistribute the material in any medium or format.

Under the following terms: Attribution — You must give appropriate credit, provide a link to the license, and indicate if changes were made. You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use. NonCommercial — You may not use the material for commercial purposes. NoDerivatives — If you remix, transform, or build upon the material, you may not distribute the modified material. No additional restrictions — You may not apply legal terms or technological measures that legally restrict others from doing anything the license permits.

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

Stacy Ford (Marketing & Outreach Chair)

Accessible Technology Coordinator, Montgomery College

Erin Kirchner-Lucas (Sponsorship Chair)

Senior Director, Digital Accessibility, RedShelf

Andrew Nielson (Awards & Presentations Chair)

ICT Accessibility Subject Matter Expert, General Services Administration

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

Trudy Knudsen Breen

Senior Sales Executive, AbleDocs

Jennifer Chadwick

Lead Accessibility Strategist (North America), Siteimprove

Ann 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.

Mark Lapole

Accessibility UX Lead (G Suite), Google

Peter McNally

Senior Consultant, User Experience Center, Bentley University

Eduardo Meza-Etienne

Director of Compliance, eSENTIAL Accessibility

Ryan Pugh

Director of Accessibility, Onix Networking

John Rempel

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

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

Dominique Wheeler

Senior Accessibility Technician, New Editions Consulting, Inc.

Gian Wild

CEO, Founder and President, AccessibilityOz

Author Index

Bostic, Trevor, 81
Brunelle, Justin F., 81
Chudnov, Daniel, 81
Davis, Ann Marie, 65
Diallo, Halima, 65
Dobre, Jolie, 101
Eng, Katherine, 55
Feldman, Matt, 1, 33
Groves, Karl, 25
Gunderson, Jon, 19
Häkkinen, Markku, 33
Higgins, John, 81
Hoffman, Al, 25
Horton, Michael, 55
Kedarshetty, Aruna, 101
King, Alan, 65
Law, Chris M., 1, 25, 43
Lewis, Anil, 25, 43
Mayo, Lisa M., 101
Miller, Laura, 29
Miller, Mark, 29
Nielson, Andrew, 55
O'Connor, Corbb, 33
O'Neill, David, 25
Paciello, Mike, 3, 33
Paiva Sr., Marcelo, 115
Paiva, Marcello, 115
Prentice, Vince, 65
Rodriquez, Sue Ann, 91
Rothberg, Madeleine, 33
Rowland, Cyndi, 25
Ryan, Kelli, 91
Sloan, David, 29
Smith-O'Connor, Kristen, 65
Stanley, Jeff, 81
Tracy, Brittany, 81
Vera, Claudio Luis, 37, 115
Wahlbin, Kathy, 29
Waites, Todd, 29
Wandke, Daman, 123
Weber-Hottleman, Kathryn, 129
Wheeler, Dominique, 65
Whiting, Jonathan, 75
Wild, Gian, 5, 137