Proceedings of the 2017 ICT Accessibility Testing Symposium: Automated & Manual Testing, WCAG 2.1, and Beyond

Washington, DC, USA

October 26 & 27, 2017

Partnership for Public Service Conference Center, Washington DC

Organized by Accessibility Track Consulting, LLC

www.ictaccessibilitytesting.org

Contents

Introduction from the Chair

Dr. Chris M. Law

Chair, The 2017 ICT Accessibility Testing Symposium

In this second year, the ICT Accessibility Testing Symposium has grown in every way since the inaugural event held at the National Federation of the Blind in November of 2016. We have more workshops, more panels, more papers and more participants attending this year at the Partnership for Public Service venue in Washington DC. The growth in our event reflects the continued growth of the accessibility testing field. More and more providers are offering testing tools and services, and consulting/management support services continue to increase in scope and capabilities accordingly.

The symposium was started with the intention of providing a forum for scientific and practice-based presentations and discussions among peers. In response to requests from last year's participants, we've added break-out discussion and meeting facilities to complement the Q&A sections of the presentations. Last year's popular 'Speed Business Card Exchange' networking opportunity is repeated in this year's program. New for this year is a closing panel featuring experts from the field discussing the present and the future of four important areas of research, development and practice in accessibility testing: (1) Integration across the development life-cycle; (2) Higher Education; (3) Standardization; and (4) Integration with the mainstream testing field. We hope that this type of panel will become an annual feature of the symposium.

On behalf of the Committee, our sincere thanks to the presenters and panelists sharing the benefits of their experience in this forum. I would also like to add my personal thanks to the members of the Committee, without whom this event could not take place. I encourage all attendees to find committee members throughout the event (they have identifiers on their badges), not just to make new connections, but to provide feedback and guidance on how we can build on our momentum for future events.

Finally, thanks to you for attending. Collectively, let's continue to grow our field, and keep making progress in the months and years to come, benefitting people with disabilities.

—Dr. Chris M. Law

Keynote.
Trust but Verify: The Value of Accessibility Policies and Testing to Support Commitments

Paul Schroeder

Director of Public Policy, at Aira

For over three decades, Paul Schroeder has worked to improve public policies and programs to expand opportunities for people with disabilities. Paul is the Director of Public Policy at Aira, and was formerly the Vice President of Programs and Policy for the American Foundation for the Blind.

Paul discusses efforts to create policies such as the Communications and Video Accessibility Act and Section 508, and describes how testing and verification is critical to ensure that principles are put into practice to ensure accountability for results.

Workshop. Introduction to Accessibility Testing (‘Testing 201’)

Gian Wild

AccessibilityOz
Melbourne, Victoria, Australia
gian@accessibilityoz.com

Abstract

The Accessibility Testing workshop will cover a whole range of testing requirements: starting with an overview of testing – who, when, what and how. This will be followed by an in-depth discussion, with exercises, of how to develop a scope analysis for a site (based on the W3C Evaluation Methodology). The afternoon will be spent looking at testing tools – with a demonstration of the main automated accessibility testing tools such as Deque’s WorldSpace, Level Access’ AMP, SiteImprove and OzART – followed by a demonstration and discussion of one-page testing tools such as WebAIM’s WAVE, the Paciello Group’s Colour Contrast Analyser, mobile testing tools and readability tools. The workshop will be run by Gian Wild, CEO of AccessibilityOz. Gian started in the accessibility industry in 1998 and built an automated accessibility testing tool in 2000 after being disappointed with the results from Bobby – the world’s first automated accessibility testing tool. Gian spent six years with the W3C contributing to WCAG2 and has been testing web sites for almost twenty years.

Overview of presentation

In this one day workshop, the following will be discussed:

Choosing who will do the testing

Will the testing be done internally or externally? If internally, do staff have the correct skills? Are their other tasks going to be completed by someone else or do they need to complete the testing in conjunction with their daily job? If externally, will it be a consultancy or a contractor? How will the decision be made? What is the budget? Who will make the final decision? Who will manage the external people? Will assistive technology testing be done? If so, it should be done by people with disabilities who are very familiar with their assistive technology. Can existing staff do this assistive technology testing?

Choosing when to test

While building a web site accessibility should be considered when writing requirements; at wireframe and design; and template. Training should be conducted for staff and the final site should be tested.

If an organization has many web sites then a plan needs to be developed to ensure that all sites are tested in an efficient manner.

Choosing what to test

What should be tested? Automated testing means that the entire site can be tested, but there is still a lot of manual testing required. Which pages should be manually tested? When choosing pages consider templates, processes, popular pages, pages required by law, pages aimed at people with disabilities, pages with unusual technology or functionality, third-party widgets and standard pages like the home-page, contact us page and search feature. Utilizing the W3C Evaluation Methodology will assist in identifying pages to review.

Choosing how to test

There are a number of testing methods and all may be applicable. Sites should be tested with an automated testing tool as well as manually. However, there are other options as well – should the site be tested by users of assistive technologies, on multiple operating systems and browsers and/or on different mobile and tablet devices?

Testing tools

There are many tools available including WAVE, Web Developer Toolbar, Deque Worldspace, Level Access AMP, AccessibilityOz OzART, Paciello Group's Color Contrast Analyzer.

Choosing how to present findings

There are numerous ways to present results. The presentation of findings will be dependent on whether the site is an existing site, if it is being rebuilt or if it is being retired. In most cases a Word document is provided with examples of results. Automated testing tools allow for more detailed results that can be handed directly to the developers to fix. A walkthrough of the results for the project manager and developers is always helpful in addressing thorny accessibility issues.

Copyright notice

Workshop. Open Source Tools for Evaluating and Inspecting Web Accessibility of Organizations to Individual Pages

Dr. Jon Gunderson

University of Illinois at Urbana/Champaign
jongund@illinois.edu

Mark McCarthy

University of Illinois at Urbana/Champaign
mcmccar2@illinois.edu

Abstract

The workshop will provide a hands-on experience on how to use open source tools to evaluate and understand the accessibility of entire organizations to individual pages based on the W3C Web Content Accessibility Guidelines (WCAG) 2.0 Level A and AA requirements. Participants are encouraged to bring and use their own devices during the workshop to analyze websites of their choice. The workshop will use their results as a discussion point of how the tools provide information on accessibility and discuss the strengths and limitations of automated testing. The workshop will discuss how the OpenAjax evaluation library is based on the W3C HTML5 specification and ARIA Authoring Practices to meet WCAG 2.0 Level A and AA requirements. The workshop provides a listening point for the developers for workshop participants to provide feedback to improve the design and utility of the tools and to guide development of new features.

Learning objectives:

1. Using site-wide summary reports for planning, managing and reporting web accessibility.
2. Use of HTML5 and ARIA design techniques for meeting WCAG 2.0 requirements.
3. Inspecting the accessibility of an individual web pages

AInspector Sidebar

URL: http://ainspector.github.io

AInspector Sidebar for the Firefox browser supports the evaluation of web pages for the accessibility features required by the W3C Web Content Accessibility Guidelines 2.0 Level A and AA Success Criteria. The tool provides both summary information and detailed information on the accessibility requirements that apply to a web page (e.g. WCAG 2.0 requirement filtering). Users can highlight and inspect element information on the page associated for each requirement. Each result has information on the requirement, techniques that can be used to implement the requirement and links to more information. AInspector Sidebar uses the open source OpenAjax Evaluation Library, Rules and Rulesets.

Functional Accessibility Evaluator

URL: http://fae.disability.illinois.edu/

The Functional Accessibility Evaluator (FAE) 2.0 provides accessibility information on an entire website for WCAG 2.0 Level A and AA Success Criteria. FAE 2.0 provides both website summary and page level detail on the accessibility requirements that applied to the pages with in a website. FAE 2.0 uses the same open source OpenAjax Evaluation Library, Rules and Rulesets as AInspector Sidebar. FAE 2.0 and AInspector Sidebar are designed to complement themselves. FAE 2.0 provides an overview of accessibility of a website that can be used in design, planning and quality assurance for project management and AInspector Sidebar provides detailed information and inspection capability of a particular page.

Bookmarklets for Accessibility Visualization

URL: https://accessibility-bookmarklets.org/

Bookmarklets can be added to any browser to provide visualizations of specific accessibility information that can helps people understand the functional accessibility of a web page by making hidden accessibility information visible to sighted web developers. Current Bookmarklets include ARIA Landmarks, Headings (H1-H6), Lists, Images and Forms.

ARIA Authoring Practices

URL: https://www.w3.org/TR/wai-aria-practices-1.1/

The ARIA Authoring Practices provide design information and working examples of ARIA enable widgets that meet Section 508 and W3C WCAG 2.0 accessibility requirements. It is important when reporting accessibility issue to developers and designers that you can refer them to technical resources that can help them understand the problem and show them effective solutions.

Copyright notice

Workshop. 508 and Mobile Accessibility: How, When, Where and Why

Alyson M. Muff, Ph.D.

ICF International
alyson.muff@icf.com

Joseph De Rosa

ICF International
joseph.derosa@icf.com

Abstract

This workshop aims to contribute to the understanding of the impact of contextual components of mobile device application and the level of accessibility involved and the ability to evaluate and test these levels. We will include specific examples and engage the audience in specific tasks. Our overall goal is to illustrate the need for a greater focus on the interrelationship between such aspects as the person, their capabilities (or lack of) the task, physical environment, social environment, temporal elements, and the technical components of touch screen and voice mobile devices.

Mobile Accessibility

“Mobile accessibility” refers to making websites and applications more accessible to people with disabilities when they are using mobile phones and other devices. Mobile technology has redefined modern society and re-parametrized’ the way people live, work, think, and learn. In the short time frame of 20 years, the mobile device has overtaken, overrun and rendered some technologies – obsolete.

The rapid advancements in the design and development of mobile devices have a significant impact on the nature of user interaction, as they can be used at any time in any location and by any one. As such there is an ever-increasing focus on the user interfaces of touch-screen devices, voice and the implications this has on the context of use and context ‘of person’. The idea of accessibility has extended into many communities as people differ widely in their physical state, mental state, style of use, and social-technological context. Contextual factors include: the physical state and capabilities of the person, the physical device, the task at hand, environment; social and technological components. All of these factors have significant impact on user performance, behavior, and outcome. These issues need to be taken into consideration during the design stage of the mobile application development process and as importantly – during the accessibility testing process.

This workshop will provide a profile of the ways in which mobile device usability evaluation methods are being adapted to reflect technological advances with a serious focus on accessibility testing and the context-of-use parameters during the accessibility testing and evaluation process. Ignoring the context of use for any and all users would be foolhardy. We will discuss the need for a shift in evaluation method paradigms with a particular focus on methods involving the user in need of assistive technologies as part of the overall development process. Having a better understanding of the context-of-use may contribute to improvements in adaptable mobile device usability and the implementation of assistive technologies – voice and touch in particular.

Mobile Devices

Mobile devices can typically be categorized into four types: unintelligent gadgets, cellular phones (mobile phones), smart phones, and devices with operating systems. Users may access distributed services anywhere, anytime using mobile devices that adapt to the users' situations. Our focus is smart phones with a side note on tablets as appropriate.

Given that the focus of this workshop is on assistive technology and the usability of touch screen mobile devices through an enhanced understanding of context of use and person – we will focus on testing and evaluating the touch and voice device characteristics as they apply to the user in specific contexts given a hearing, vision, physical/motor, and cognitive impairment.

Accessibility, Tools and Evaluation Processes

Looking at the current literature regarding accessibility and ‘real world’ evaluation few studies have been conducted using the current tools readily available on mobile devices with context of person and place in mind.

During the workshop we will introduce the audience to the accessibility features that are available with most mobile devices. We will review, demonstrate, and evaluate the accessibility options using the Apple iPhone as our primary device. We will evaluate and summarize the accessibility options available for Android devices, as well. We will also review tablets highlighting similarities and differences to the mobile phone. Accessibility accessories will also be a topic for discussion. These accessibility accessories are necessary for devices where users have limited or no movement.

The structure of the workshop is to give an overview of the mobile device accessibility option – followed by an example where we invite audience participation. We will conclude each section of the presentation with a summary of findings, an evaluation of the outcome, and audience feedback and input. We plan to focus on the usability issues inherent to the use of the device when accessing content to demonstrate the critical need for an intersection between accessibility and usability in the mobile world.

Mobile Accessibility Aids

During the workshop we will focus on accessibility aids related to hearing, vision, physical/motor and cognitive limitations.

Copyright notice

Workshop. Integrating Accessibility Across the Development Lifecycle

Kathleen Wahlbin

Interactive Accessibility
363 Willis Road, Suite 121
Sudbury, MA 01776
www.interactiveaccessibility.com
kathyw@ia11y.com

Mark Miller

Interactive Accessibility
markm@ia11y.com

Jeremy Curry

Interactive Accessibility
jeremyc@ia11y.com

Abstract

Many organizations today are struggling with the best way to support accessibility and maintain accessibility compliance over time. Conforming to the WCAG 2.0 and the upcoming WCAG 2.1 is a corporate-wide effort that goes well beyond technical accessibility. Real success comes from ensuring that accessibility is considered and implemented throughout the development lifecycle. Accessibility must encompass the entire organization. This includes support from upper management, policies, roles and responsibilities, procurement, training and more. This session will demonstrate these concepts using the capability maturity model (CMM). During this half-day workshop, participants will perform an accessibility development lifecycle maturity assessment and create an action plan that they can use to take back to their organization to improve their accessibility program. Come and learn how to build accessibility knowledge throughout your organization and drive complete organizational change for the better by integrating accessibility into all aspects of business.

Integrating Accessibility Across the Development Lifecycle

Introduction

To produce a usable and accessible product, accessibility should be integrated into every stage of development. Before the project starts, it should be defined what accessibility means for the project. Product and project managers need to make accessibility a priority and ensure that accessibility is built into the overall process, track accessibility results and document decisions made throughout product development lifecycle. The business requirements should include the user needs for people with disabilities. If third-party applications or development frameworks are needed for the project, they should be evaluated for accessibility before they are purchased. During interaction design, wireframes, and the product design, accessibility requirements must be met and more importantly, the accessibility decisions made during design must be communicated to the developers. Development must integrate accessibility and conduct unit testing to ensure that the code meets the accessibility requirements. Quality assurance needs to ensure that the product meets the business requirements, product accessibility requirements and works with assistive technology and browsers.

Organizational Accessibility Integration

In addition to key upper management support, accessibility should be integrated into corporate policies that directly impact the development of a product including design guidelines such as brand usage, web style guides, and design pattern libraries, technology guidelines such as technology strategy, browser and assistive technology support, and code libraries, procurement policies, product launch checklists, exception policies and procedures, support center call center scripts, FAQ documents, and social media guidelines.

Validation and measurement are key to success. Establishing metrics and tracking progress is important and should be integrated into business controls. The end goal of an accessibility program is to produce products that are accessible and usable for people with disabilities but making the statement that the products will be 100% compliance in a certain timeframe is unrealistic. Instead, it is important to set both short-term and long-term goals to drive accessibility transformation. Accessibility metrics will drive accessibility improvements and will uncover issues such as potential issues with overall processes, base technologies, issues with project and personnel management, training gaps, issues with developer and testing tools, potential issues with prioritization of priorities, and more.

Capability Maturity Model

This session will provide an in-depth detail how to integrate accessibility into their organization's workflow to ensure that products are accessible and usable by all and how to use those results to assess where they are as far as organizational accessibility maturity using the capability maturity model (CMM).

The session will cover topics such as:

- What is an accessibility program?
- How to integrate accessibility and assess organization accessibility maturity
- Accessibility policies and how to integrate accessibility into existing policies
- Accessibility roles and responsibilities in the development lifecycle
- Procurement processes

- Testing strategies – automated, manual, assistive technology
- Importance of tracking project decisions and measuring results
- Exception policies and procedures
- Training

During this half-day workshop participants will perform an accessibility development lifecycle maturity assessment. Round table discussions will focus on the following one of these five key topics and then will share their results with the group:

- Strategies for integrating accessibility and defining project roles and responsibilities
- Challenges with integrating accessibility in the development lifecycle and how to drive organizational change
- What and how to ensure accessibility throughout the development lifecycle including testing strategies
- Techniques for validation and measurement to ensure that efforts are producing results
- How to build accessibility knowledge throughout the organization

Participants will learn from other participants on what has worked or not worked for other organizations. Together participants will create an action plan that they can use to take back to their organization to improve their accessibility program.

Copyright notice

Panel. Accessibility Conformance Testing (ACT) Panel Discussion: Harmonize Accessibility Testing

Wilco Fiers (Chair)

Deque Research
Rachmaninoffplantsoen 56, Utrecht, Netherlands
wilco.fiers@deque.com

Shadi Abou-Zarah

W3C Web Accessibility Initiative (WAI)
shadi@w3.org

Alistair Garrison

Level Access
alistair.garrison@levelaccess.com

Katie Haritos-Shea

Deque Systems, Inc.
katie.haritos-shea@deque.com

Kathleen Wahlbin

Interactive Accessibility
kathyw@ia11y.com

Abstract

The W3C Web Content Accessibility Guidelines (WCAG) continues to gain widespread recognition around the world. While WCAG 2.0 has been designed and written to be far more testable than its prior version, there are still differences in interpretation and testing of accessibility. Last year, Accessibility Guidelines Working Group (AG WG) launched the “Accessibility Conformance Testing (ACT) Task Force”, to help address this issue. ACT TF is scoped to develop a specification for automated, semi-automated, and manual web accessibility testing procedures. This envisioned W3C standard should complement WCAG, and allow the development and sharing of testing procedures with a more consistent interpretation of WCAG.

Background

Evaluating the conformance of web content to the WAI Web Content Accessibility Guidelines (WCAG) [https://www.w3.org/TR/WCAG20/] is often a non-trivial task. Varying interpretations are manifested in evaluation tools and testing methodologies, with often conflicting results. That is, the same web content might be deemed to have ‘passed’ accessibility requirements by one method, yet ‘failed’ by another. This contributes to confusion for those who people responsible for ensuring the content they produce meets the required standard.

Several initiatives and research projects have been undertaken in Europe, the United States, and elsewhere to address interpretation issues. These resulted in a number of different testing approaches, each with their own collection of advantages and disadvantages. However, because these approaches were attempted in relative isolation of each other, it did little to actually harmonize WCAG interpretations worldwide. This is the problem that the ACT Task Force [https://www.w3.org/WAI/GL/task-forces/conformance-testing/] is looking to address.

Panel Discussion Topics

This panel will give an update on the work of the Accessibility Conformance Testing (ACT) Task Force and invite feedback from the audience, to help shape the effort. A number of topics will be approached by this panel.

ACT Rules Format 1.0

The second public working draft of the ACT Rules Format 1.0 was published on September 12, 2017. The panel will outline the goals of this draft, how it came about, and what some of the key features of the ACT Rules Format are. Following this we will look at some of the feedback and discussions that will be going into future drafts, to address topics like how to manage accessibility support in a consistent manner during accessibility testing.

Developing and publishing ACT Rules

In order to reduce the interpretations differences that people get from different accessibility testers (tools and human auditors), the ACT Task Force is looking to put together a list or rules. These will teach accessibility testers and developers of accessibility tools about established ways of getting accessibility data. During this part of the panel discussion we will talk about what it takes to get this done.

Building a repository of accessibility test cases

As one of the first steps to harmonizing accessibility testing, the ACT Task Force is looking to build up a repository of accessibility test cases. Many organizations have examples of good and bad practices in accessibility as test cases or training material. We’ll discuss where we are in bringing those together, and how others can contribute to and use these resources.

The benefits of standardizing accessibility testing

The ACT Taskforce is contributed to by a variety of accessibility tool vendors and accessibility service providers. The slow shift from treating accessibility test methods as a business commodity, to something that should be shared and harmonized, is the main reason for most of these organizations to contribute. We will explore how these organizations think they, and their customers benefit from harmonization in the accessibility test space.

The future of accessibility testing

In the last part of the panel discussion, the group will talk about what they expect will come out of the work of the ACT Task Force. What will the practical benefits be to accessibility testers and tool developers, how will rules be maintained and how will new technologies drive this. There may also be implications for future versions of WCAG, with test procedures describes separate from the accessibility requirements.

Copyright notice

Panel. EPUB Accessibility Conformance, Testing and Certification: A Comprehensive Implementation

George Kerscher, Ph.D. (Chair)

DAISY Consortium
Missoula, Montana, USA
kerscher@montana.com

Avneesh Singh

DAISY Consortium
avneesh.sg@gmail.com

Marisa DeMeglio

DAISY Consortium
marisa.demeglio@gmail.com

Rachel Comerford

Macmillan Publishing
rachel.comerford@macmillan.com

Charles LaPierre

Benetech
charlesl@benetech.org

Amy Mason

National Federation of the Blind
amason@nfb.org

Abstract

Digital publishing offers a path forward to full access to publications for persons with disabilities. Despite all the advantages it offers, however, publishers are still creating content that fails many basic Web Content Accessibility Guidelines (WCAG) success criteria. More problematic is that publishers are often not aware that WCAG applies to digital formats, like EPUB, or how to translate the page-centric nature of WCAG to a packaged multi-document format. Many aren’t even sure what they need to do to pass, or how to begin checking

conformance. The DAISY Consortium has taken the lead role in the development of an accessibility specification for EPUB and is undertaking work on a tool for conformance testing and reporting. Together, these will aid in understanding and applying conformance tests against EPUB publications throughout the lifecycle from production to distribution to consumption.

Background

Publishing Standards

EPUB is a distribution and interchange format for digital publications and documents based on the Open Web Platform (HTML, SVG, CSS, JavaScript, etc.). It defines a means of representing, packaging and encoding structured and semantically enhanced Web content for distribution in a single-file format.

On January 5, 2017, the International Digital Publishing Forum (IDPF) approved the specification "EPUB Accessibility 1.0: Conformance and Discovery Requirements for EPUB Publications" (https://www.w3.org/Submission/epub-a11y/). This specification builds on WCAG 2.0, adding publishing specific requirements and success criteria. It also requires schema.org metadata that declares the accessibility of the publication.

More importantly, this specification brings clarity to the question of how to make EPUB publications accessible, setting the stage for further improvements in the production of accessible content.

Accessibility Checker for EPUB (Ace)

The DAISY Consortium is working on a project to develop an automated testing tool that will help publishers verify that their publications meet accessibility standards. Named the Accessibility Checker for EPUB (Ace), this open source project builds on the aXe accessibility engine, enabling validation of both packaged (zipped) and unpackaged EPUB publications.

The code base is designed to apply automated verification techniques, as well as provide assistance for the manual checking of publications. The expectation is that Ace will be used by publishers in their production processes, by distributors in their ingestion processes, and by organizations providing consulting and certification services.

To support the manual inspection of items not automatable, the report generated by Ace can be used in conjunction with a web site designed to assist in manual checking. The generated check list will enable a customized list based on content present in the publication.

Data visualizations of the overall document structure, as well as visualizations of specific items, such as images with their surrounding text can be used to provide people involved with the manual checking an efficient workflow to assure that the publication is accessible. This will be paired with a knowledge base that provides guidance on techniques that may be used for accessibility.

Global Certified Accessible (GCA) Program

On June 22, 2017, Benetech launched the official start of their Global Certified Accessible program. This pilot incorporates consulting and certification services and has been running since the first quarter of 2017.

This program enables publishers to have their EPUB publications certified as conforming to the EPUB Accessibility 1.0 specification, and marks a first where a trusted third party with a focus on accessibility and publishing is certifying content as accessible to the new specification.

Publisher Participation

The Publishing@W3C (https://www.w3.org/publishing/) initiative has many participating publishers. This participation, coupled with the early adoption of the GCA program, is encouraging. We expect to see certified EPUB 3 content that meets the EPUB Accessibility specification at WCAG 2.0 AA levels by the end of 2017.

Macmillan Publishing, for example, is looking to replace their home-grown software with the Ace tool. We also expect to have the early adopters in publishing to provide feedback and recommendations on Ace.

Delivery Platforms

Now that discovery metadata is a required part of meeting accessibility requirements for EPUB, the vendors of distribution platforms have opportunities to promote accessible content in their catalogues and order fulfillment processes.

VitalSource has been a champion in this regard. VitalSource plans to use the Ace tool as a routine component in their ingestion process. The automated components and the ability to extract accessibility metadata will enable informed purchasing decisions. We want to promote this model and have all commercial distribution channels, and all library catalogues include this information.

Policy and Legal Implications

Over the past five years, the National Federation of the Blind have hosted inclusive publishing conferences at their Baltimore headquarters. With the standards-based approach to certification, we can expect to see policy and best practices focus on born-accessible certified EPUB materials integrated in to the fabric of publishing.

The Panel

The goal of the panel is to familiarize the audience with the developments and issues around accessibility conformance and testing of EPUB content through the full lifecycle of a publication. Topics to be covered include:

- the development and requirements of the EPUB Accessibility specification;
- the development of the ACE testing tool and the challenges of applying and automating WCAG with packaged web content;

- integration of automated and manual accessibility testing in publishing workflows and vendor ingestion process;
- the development of certified testing authorities;
- legal issues around digital publishing and accessibility conformance.

Copyright notice

Panel. Accessibility Testing: The Present, The Future

Anil Lewis (Chair)

National Federation of the Blind
alewis@nfb.org

Matt Feldman

The Paciello Group
mfeldman@paciellogroup.com

Cyndi Rowland

WebAIM, National Center on Disability and Access to Education
Utah State University
cyndi.rowland@usu.edu

Shadi Abou-Zarah

W3C Web Accessibility Initiative (WAI)
shadi@w3.org

Karl Groves

Tenon.io
karl@tenon.io

Abstract

This panel will be the closing session of the symposium. Panelists will discuss the current state of the field as well as short-term and long-term trends. The panelists will also be relating their views with reference to the papers, panels and workshops that were presented at this year's symposium.

Topics

- **Accessibility testing and integration across the development life-cycle**
—Matt Feldman, The Paciello Group
- **Testing in Higher Education, and fostering the involvement of students in the testing field**

—Cyndi Rowland, WebAIM, National Center on Disability and Access to Education, Utah State University

- **Standardization in testing and reporting methods**
  —Shadi Abou-Zahra, W3C Web Accessibility Initiative (WAI)
- **Accessibility testing and integration with the mainstream IT testing field**
  —Karl Groves, Tenon.io

Copyright notice

Federal Website WCAG 2.0 Compliance and Accessibility Trends

Dr. Jon Gunderson

Coordinator of Accessible IT Group
Disability Resources and Education Services
University of Illinois at Urbana/Champaign
jongund@illinois.edu

Dr. Jonathan Lazar

Professor, Dept. of Computer and Information Sciences
Director, Undergraduate Program in Information Systems
Towson University
jlazar@towson.edu

Zipporah Goldenfeld

Department of Statistics
College of Liberal Arts and Sciences
University of Illinois
goldenf2@illinois.edu

Abstract

This paper analyzes a subset of WCAG 2.0 requirements that can be tested automatically, for over 600 federal websites. Using a similar methodology as in 2015, the federal web sites were evaluated in February 2017. The results show that overall federal websites have made slight improvements in accessibility since 2015. Some websites have made considerable improvements and other websites have developed more accessibility problems.

Introduction

U.S. Federal government web sites are required to be fully accessible for people with disabilities. Section 508 of the Rehabilitation Act has required web accessibility since 2001, a total of 16 years. Given the legal requirements, it is surprising that many Federal web sites have been out of compliance. There have been many possible explanations for the inaccessibility of Federal web sites in the 16 years period. While the U.S. Access Board has the responsibility to update the regulations, and the U.S. Department of Justice (DOJ) is required to perform data collection related to Section 508 compliance every two years, no government agencies have the statutory power to perform enforcement of Section 508 (Lazar et. al 2017). Over the 16 year period, only three data collection reports were published by the DOJ, in 2001, 2003, and 2012, respectively.

The most recent DOJ report, in 2012, only collected data on plans and perceptions about Section 508 compliance, not actual data on compliance. Because of the lack of data collection by the DOJ and other government agencies, private researchers have been filling the gap, providing information about compliance. Two earlier studies performed by two different research groups, published in 2006 (Jaeger) and 2011 (Olalere and Lazar), documented low levels of web accessibility compliance on federal websites.

Since 2006, the Section 508 regulations have been undergoing a "refresh," or an update. The original Section 508 regulations utilized a modified version of Web Content Accessibility Guidelines (WCAG) version 1.0, which quickly went out of date. The WCAG, coming out of the Web Accessibility Initiative of the World Wide Web Consortium, are considered to be the most well-documented, well-accepted accessibility guidelines in the world. Most governments that have ICT accessibility requirements, utilize the WCAG. WCAG version 2.0 was approved by W3C in 2008. Three draft rules of the new Section 508 were issued (in 2010, 2011, and 2015, respectively), and some included modified versions of WCAG 2.0, but the final rule issued in January 2017 included WCAG 2.0 by incorporation. Therefore, the current version of Section 508 includes WCAG 2.0 as the appropriate technical standard for web sites.

One of the challenges in assessing and evaluating Section 508 compliance, is that the Federal government publishes very little data on compliance, and not even information on the methods utilized to evaluate for compliance. From 2014-2016, it is known that all federal agencies were required to send data twice a year about Section 508 compliance, to the Office of Management and Budget (Lazar et. al. 2017). It is unknown, however, whether that data is accurate, or how it is utilized, or even what the data is, since it has never been publicly released. Therefore, it is up to private researchers to collect data on Section 508 compliance, and assess trends over time. Furthermore, it is known (Lazar et. al. 2017) that different agencies utilize different approaches, metrics, and evaluation methods for improving their Section 508 compliance.

The three common approaches for evaluating a web page for accessibility are usability testing (involving people with disabilities attempting representative tasks), expert reviews (also known as manual reviews or inspections), and automated testing (Lazar, Goldstein, and Taylor, 2015). Of those three methods, the only one that scales well to doing evaluations of tens of thousands of web pages (and U.S. Federal government has many more!) is automated testing. However, previous research has documented that automated tools can often give misleading or confusing results, and one challenge to using automated tools is using them in a manner where the results are interpreted appropriately (Lazar et. al. 2017). The method described in later sections of this paper specifically focuses only on using the "presence" of an accessibility feature rather than determining the effectiveness or correct implementation of a feature, because the presence of a feature required by WCAG 2.0 can be accurately tested using an automated tool, without requiring any manual check or human interpretation.

Two of the co-authors of this paper did a data collection effort in September 2015, evaluating 629 Federal web sites, with a total of 28,429 webpages to determine the presence of many accessibility features, which can be accurately determined utilizing automated tools (see Lazar et. al. 2017 or later research methodology sections for more information). February 2017 seemed to be a perfect time to replicate that data collection effort from September 2015, for two reasons: 1) a new administration was inaugurated in January 2017, and 2) the final rule for the new

Section 508 regulations was issued in January 2017. This is a pivotal time in the history of Section 508. Therefore, it is important to document current Section 508 levels at that time, and use that data as a milestone for future data collection efforts as a comparison. In addition, since the same methodology and the same 629 sites were evaluated, it is possible to examine whether site accessibility improved, remained the same, or declined, over the approximately 18-month period between September 2015 and February 2017.

Research Methods

A list of Federal websites was previously identified from an index of websites on USA.gov the U.S. government's official web portal, and utilized in the earlier study (Lazar et. al 2017). A potential of 1,094 websites were identified, and further analysis found broken and redundant links, reducing the total usable URLs to 629. The 629 URLs were analyzed using the OpenAjax Accessibility (OAA) Evaluation Library using the Functional Accessibility Evaluator (FAE) application to sample the first and second level pages of each website in September of 2015 and February of 2017. In September of 2015 a total of 28,429 pages were analyzed and in February of 2017 a total of 23,713 pages were analyzed. The evaluations used HTML5 and ARIA Techniques ruleset of the OAA Evaluation Library Version 1.0.

It is important to note that our study is not aiming to determine compliance with Section 508. Given that automated tools cannot fully determine if a feature has been implemented properly, doing only automated testing cannot determine compliance with guidelines or assess usability. However, automated tools are very good at determining the presence of a feature, and are also the only evaluation approach that scales well to evaluating large numbers of web pages. Many previously published studies utilize automated tools and a combination of manual checks, or assume that the presence of a feature equals the successful implementation of a feature. We do not use either approach. Instead, we only evaluate the presence of an accessibility feature. Evaluating the presence of accessibility features over thousands of federal web pages, can serve to evaluate the effectiveness of accessibility policies and procedures within U.S. Federal agencies.

Website Implementation Score

For each website analyzed, an implementation score was computed for the website based on the rule results for the website. An implementation score ranges from 0-100 to indicate the extent a rule requirement has been implemented for a website, with 0 meaning no implementation and 100 meaning fully implemented. An implementation score is only computed for rules which include pass/fail results and is calculated by dividing the number of passed items by the sum of the number of passed and failed items. For example, if a website contains 200 images and 150 of the images have an ALT attribute the implementation score is 75 (e.g. 150/200). An implementation score for a website is an average of the implementation scores of the individual rule results. There are 132 rules in the OpenAjax Evaluation Library HTML5 and ARIA Techniques ruleset and typically about 40-50 rules result in pass/fail results for a website.

Website Results

Website results were analyzed two ways. The first was to look at the overall implementation scores of the websites for 2015 and 2017 and the second way was to look at the distribution of implementation score differences for each website. The overall implementation score provides an indication of the level of implementation for each year and to see if there were any general trends in improvements or decline in accessibility.

Overall Implementation Levels

The data is also described in Figure 1. The implementation score ranges from 0-100 and are a measure of overall implementation of the rules with pass/fail results. A score of 100 means the rules are fully implemented and include rules like IMG elements have ALT attribute form controls have labels and page contains main landmarks. The rules included in the implementation score include all the rules in the OpenAjax Evaluation Library for the HTML5 and ARIA Techniques ruleset (132 rules). Not all the rules result in pass/fail results and only results with pass/fail results are included in the computation of implementation score. The Y-axis of the figure is the overall implementation score for a particular website and the X-Axis is the list of websites ordered from lowest to highest implementation score. The Figure shows two sets of data points, one set for 2015 and the other for 2017. Between 2015 and 2017, the figure shows a slight improvement of in overall implementation scores of about 2.2%. The figure has two strong features. The first 100 websites generally follow a linear progression from an implementation score of 20 to 60. The next 500 websites show an approximately linear progression of implementation score from 60 to 90. There are a few implementations scores near 100 (which would be considered an ideal score).

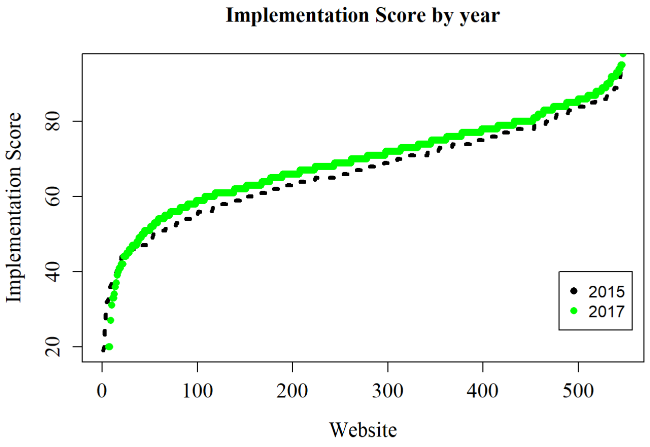


Figure 1 - Overall implementation scores for websites in 2015 and 2017

Changes in Website Implementation Level

A difference score was computed for each website by subtracting the 2015 implementation score for a website from the 2017 implementation score for the same website. This measure provides a distribution on the number of websites that are getting more accessible, less accessible or staying about the same. In this figure, a score of zero means the implementation score for a particular website did not change from 2015 to 2017. A positive score means there was an improvement in implementation and a negative score means there were more problems. The data in the figure ranges from about -20 to 20 for changes in implementation scores. About 60% of websites had minimal change implementation scores, about 30% of websites improving and 16% of websites getting worse. About 54% of website had no significant changes in implementation score. Figure 2 plots the difference data from the smallest (negative values) to the largest values and has three prominent features. The first 20 websites show a steep line of difference scores from -80 to -20 and then over the next 80 websites there is a curve from -20 to 0. The next 300 websites show an implementation score of about 0 or slightly above 0. The last 180 websites show an inverse curve of difference scores from about 2 to 58.

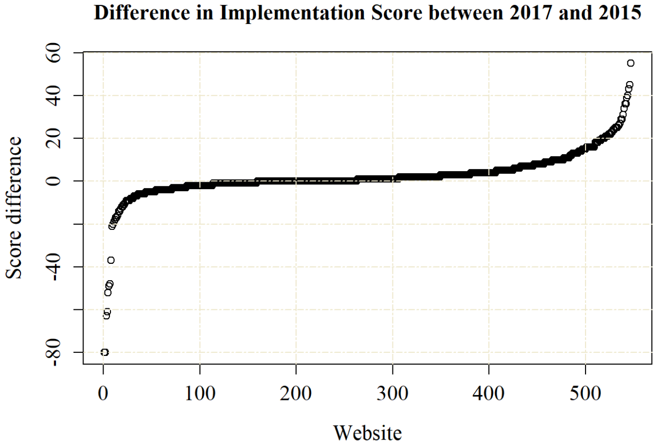


Figure 2 - Implementation score differences for each website

Individual Rule Results

This section provides details about changes in the presence of specific requirements from WCAG 2.0. The score for specific rules is based on the number of pages that completely pass a rule. For example, if a page has 20 images and 19 of the images have an ALT attribute, and one is missing an ALT attribute the page “fails” the rule. Only if all 20 images have an ALT attribute does the page “pass”. There are also some pages where a rule does not apply. So, this is not a highly granular measure and future data collection needs to address this limitation.

Alt text for Images

Figure 3 shows the results for the presence of an alt attribute for images which is a requirement of WCAG Success Criteria 1.1.1 “*Non-text Content: All non-text content that is presented to the user has a text alternative that serves the equivalent purpose, except for the situations listed below*”, and is a level A requirement. Overall, alt text is often present, with 2015 showing 71.8%and 2017 showing 70.6% of pages passing. There was a slight reduction in the number of pages passing, but also a slight reduction in the percentage of pages failing, 21.0% to 20.7%.

This is due to more pages not having images at all (e.g. labeled as n/a bars in the Figure 3). A high level of implementation is expected on such a well-known web accessibility requirement. The percentage of pages with no images (i.e. not applicable) was 6.2% in 2015 and 8.7% in 2017.

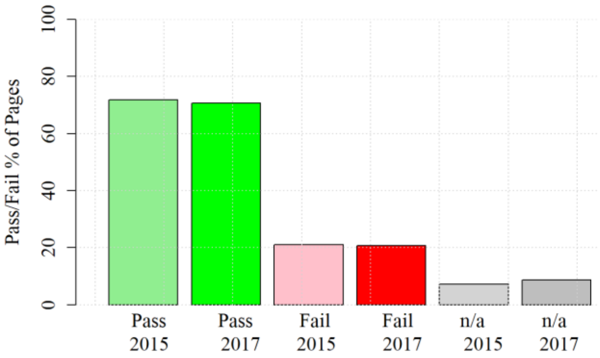


Figure 3 - Percent of pages that fully implement ALT attribute for images

Form Controls

Labels for form controls is a requirement of WCAG 2.0 Success Criteria 3.3.2 “*Labels or Instructions: Labels or instructions are provided when content requires user input*” and is a level A requirement. Labels orient users of screen readers to the purpose of a form control input. Unlike ALT text for images, for which an empty ALT text could satisfy the rule (for decorative graphics or spacers), a form control label must have text content to meet Section 508 and WCAG 2.0 accessibility requirements. The data, also shown in Figure 4, shows that in 2015 40.0% of pages had labels for all of the form controls on the page and 39.2% in 2017. The percentage of pages with at least one form control missing a label is 36.4% in 2015 and 35.8% in 2017. The percentage of pages with no form controls (i.e. labeled as n/a on Figure 4) was 23.6% in 2015 and 25.0% in 2017.

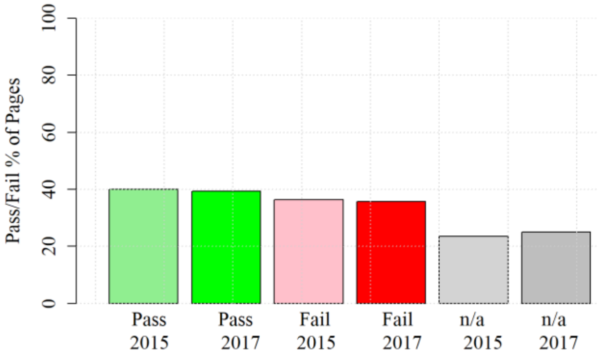


Figure 4 - Percent of pages that fully implement labels for form controls

Main Landmark

Landmark technology is a part of the W3C Accessible Rich Internet Application (Craig and Cooper, 2014) specification and a main landmark is designed to help screen reader users find and navigate to the main content of the page, providing an alternative to the skip to main link commonly used. Pages should have only one main landmark that contains the primary content on the page. Other landmarks (e.g. banner, contentinfo, navigation, complementary) are used to identify other types of content on the page to screen reader users. The main landmark is the most important and authors must intentionally define a main landmark using the main element or role=main attribute on an element that contains the main landmark. The main landmark is related to Success Criteria 2.4.1 "*Bypass Blocks: A mechanism is available to bypass blocks of content that are repeated on multiple Web pages"*, which is a level A requirement. Screen readers have commands to allow direct navigation to the main landmark on a page.

The data is shown in Figure 5 and it shows that in 2015 14.6% of pages had a main landmark and 22.9% in 2017. The percentage of pages without a main landmark is 85.4% in 2015 and 77.1% in 2017. Since a main landmark should be on every page, there are no pages where this requirement is not applicable (i.e. labeled as n/a in figure 5). The results show approximately a 50% increase in the percentage of pages defining a main landmark.

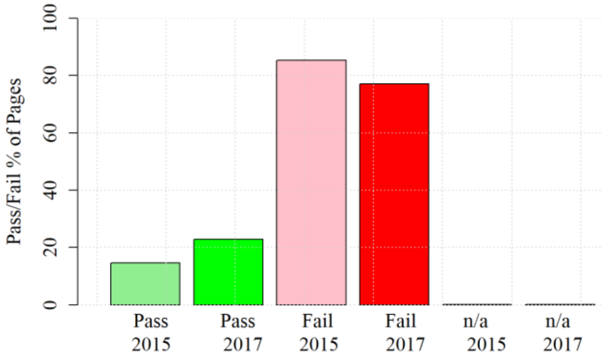


Figure 5 – Percent of pages that implement a main landmark

Discussion

Overall the data shows the accessibility of federal websites incrementally improved by a little over 2% in the 16 months between audits. The data shows that overall only about 70% of the possible accessibility features that were automatically tested in this study are being implemented. While there are still some websites with very low implementation scores, the vast majority range from 60% to 80% implementation. The two long standing rules related to the ALT attribute for images and labeling of form controls show two very different stories. ALT attribute for images is implemented at a high level and at twice the rate as labels for form controls, even though these requirements were part of the original Section 508 requirements from 2001. The data cannot tell us why there is such a difference in these basic requirements, but one factor maybe that having a text description for an image is easier for authors to understand from an accessibility perspective. It is easier for content authors to understand that if a person can't see an image and they need to have a text description of the image. Content authors may find the concept of a form control label more challenging to understand.

Data from the main landmark rule is an indication of Federal websites adopting the latest technologies to improve accessibility. ARIA and HTML5 are relatively new technologies and only recently have WCAG 2.0 techniques been updated to incorporate some of the new capabilities of these technologies to implement accessibility requirements. While the use of the main landmark is still relatively low, it is increasing at much higher rate (e.g. 50% increase) than the general increase in accessibility (e.g. 2.2%) for all rules.

The results invite additional investigation on why some sites are getting better and why some sites are getting worse. Why are some accessibility requirements (e.g. rules) highly implemented and other requirements less frequently implemented? We hope to examine this question in future research.

References

Craig, J. and Cooper, M. (eds.) (2014) W3C Accessible Rich Internet Application, https://www.w3.org/TR/wai-aria/.

Jaeger, P. (2006). Assessing Section 508 compliance on federal e-government Web sites: A multi-method, user-centered evaluation of accessibility for persons with disabilities. Government Information Quarterly, 23 (2), 169-190.

Lazar, J., Williams, V., Gunderson, J., & Foltz, T. (2017, January). Investigating the Potential of a Dashboard for Monitoring US Federal Website Accessibility. In Proceedings of the 50th Hawaii International Conference on System Sciences.

Lazar, J., Goldstein, D. F., & Taylor, A. (2015). Ensuring digital accessibility through process and policy. Waltham, MA: Morgan Kaufmann.

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

Copyright notice

Agile Accessibility: Ensuring accessibility throughout the Agile development process

Andrew Nielson, CSM, PMP

New Editions Consulting, Inc.
anielson@neweditions.net

Tyler Matney

New Editions Consulting, Inc.
tmatney@neweditions.net

Cindy Ryan

New Editions Consulting, Inc.
cryan@neweditions.net

Abstract

U.S. federal government agencies are increasingly migrating toward Agile methodologies for Information Communication Technology (ICT) development, creating a need for government IT professionals to revise their system development lifecycle approaches and related governance policies and practices. Like other forms of IT governance, Section 508 programs often struggle to force conformance validation designed for a traditional waterfall methodology into an Agile methodology. Seasoned Section 508 Program Managers accustomed to coordinating Section 508-related activities in the context of waterfall must still educate stakeholders about accessibility requirements while reconciling Section 508 conformance with sometimes unfamiliar Agile terms and concepts. After providing a brief overview of Agile concepts, this paper presents considerations and implications for incorporating accessibility throughout an Agile development process, including conformance governance and documentation.

Accessibility in Agile Methodologies

What is Agile?

Agile refers to a broad set of concepts that share the same four common values (from http://agilemanifesto.org/):

- Individuals and interactions over processes and tools
- Working software over comprehensive documentation
- Customer collaboration over contract negotiation
- Responding to change over following a plan

The twelve principles of Agile (http://agilemanifesto.org/principles.html) have led to various interpretations with application in a number of models, including those focused more on practices (e.g., Extreme Programming), those focused more on work flow (e.g., Scrum and Kanban), as well as a number of hybrids and combinations. The approach described below relies mostly on the Scrum approach (one of the most common models), to illustrate how accessibility practices should fit within Agile. Many of the same principles can be applied to other models of Agile.

The Agile Approach to Accessibility

Accessibility as an integral part of Agile development

As depicted in Figure 1, a product is built in constrained increments of time known as sprints in Agile. A sprint is typically two to four weeks. Prior to initiating a product build, all requirements are added to a Product Backlog. When planning for a sprint, a subset of these requirements is moved to what is known as a Sprint Backlog. These requirements are then used to build a product increment within the sprint. These products may be "done" (ready to provide to an end-user) – or not (requiring additional work in another sprint). Daily scrum meetings are conducted to assist the development team in collectively building the product increment within the sprint.

An effective Agile approach to developing accessible software and web content starts with the fundamental understanding that an application or website must be functional and usable for all users, regardless of ability. Accessibility is a habitual expectation that all development team members accept as part of the normal course of work. Naturally, adequate training is necessary for developers to know how to write code that is accessible and conforms to accessibility standards. Likewise, testers must be able to follow a standard test process to reliably validate conformance and overall accessibility.

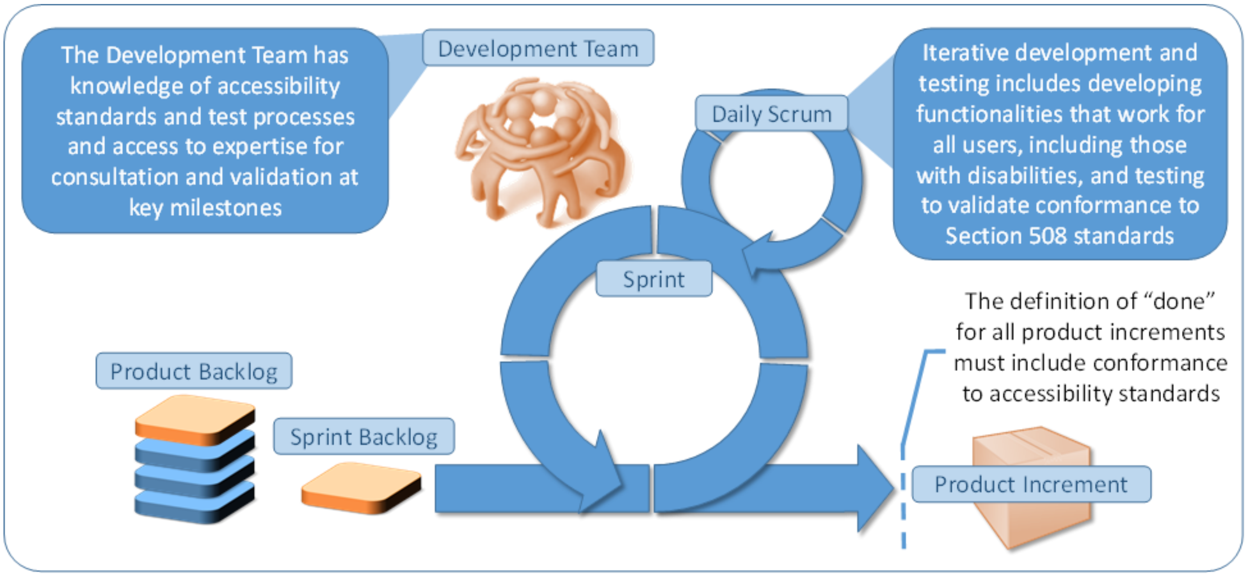

The Development Team has knowledge of accessibility standards and test processes and access to expertise for consultation and validation at key milestones
Development Team
Daily Scrum
Iterative development and testing includes developing functionalities that work for all users, including those with disabilities, and testing to validate conformance to Section 508 standards
Sprint
The definition of "done" for all product increments must include conformance to accessibility standards
Product Backlog
Sprint Backlog
Product Increment

Figure 1 - Accessibility should run throughout the Agile Scrum process.

Key accessibility integration points within the Scrum process include:

1. The development team has knowledge of accessibility standards and test processes and access to accessibility expertise for consultation and validation at key milestones.
2. Iterative development and testing includes developing functionalities that work for all users, including those with disabilities, and testing to validate conformance to accessibility standards.
3. The "definition of done" for all product increments must include conformance to accessibility standards.

The sections below further describe other key elements of accessibility in Agile development:

Include conformance to accessibility standards in the "definition of done"

Agile development teams must form a shared understanding of what it means for work to be complete. The "definition of done" can vary significantly from one development team to another, depending on client or product owner requirements, development team preferences, and the environment. All product increments must conform to Section 508 standards in order to be considered "done."

Incorporate accessibility standards in key artifacts

Regardless of an organization's specific software development methodology, ICT accessibility must be included, at a minimum, in all key artifacts.

- **Requirements:** Whether described as elements of a Product Backlog, System-Wide Specifications, or other designation of requirements, the development team must establish accessibility standards as technical requirements for its ICT development.
  - Each Agile team determines the best method to document these requirements for the team. Some teams include accessibility requirements in acceptance criteria, as separate user stories, or in other ways.
  - Whatever the documentation method, the team must understand the necessity for incorporating accessibility in requirements.
  - As with other system requirements, teams should identify any deficiencies in the team's knowledge, skills and abilities to implement accessibility requirements early in the process in order to better identify solutions to meet the accessibility requirements (e.g., training, hiring or subcontracting to an IT accessibility resource, working with an agency's Section 508 office, etc.)
- **Design and Architecture:** Conformance to accessibility standards influence design and architecture decisions, whether it relates to choices about specific types of technologies or user interface (UI) design considerations. The development team must include accessibility standards conformance in the earliest stages of design and architecture to promote the ability to conform to the standards in the subsequent development of the application.
  - User experience (UX) designers must address all requirements (including those for accessibility) when developing wireframes and visuals.

  - Strong UX designers typically have experience or education in designing to include elements that are universally usable – meaning that they are designed for all user types, including those with disabilities.

- **Test Plans:** A standardized accessibility test process for ICT content informs test planning, serving as elements of test scripts and describing the specific processes necessary to validate that "working software" really will work for all users.

Minimize subjectivity by following a standard accessibility test process

A development team should adopt and implement a standard test process for each application and for all features or functions of ICT to avoid confusion and minimize subjectivity in validating conformance to accessibility standards. Documenting the standard test process and following the applicable test procedures throughout iterative development establishes a common understanding of accessibility requirements and helps improve confidence that the ICT conforms to the standards before it moves to production.

Using Test-Driven Development and automated testing in Agile accessibility

The concept of Test-Driven Development (TDD) is one of the common hallmarks of Agile methodologies. In a nutshell, TDD requires writing tests to validate that functional and technical requirements are met before actually writing any code.

Writing the tests before writing code encourages developers to focus more acutely on requirements before shifting focus to develop features. The process also tends to produce simpler designs as it discourages developers from developing code outside of the scope of requirements (only write tests that validate requirements, and only write code sufficient to pass the tests). Assuming that accessibility is incorporated within requirements, TDD likewise requires understanding the accessibility requirements in order to write tests to validate that specific interface elements and content conforms to the requirements as they are developed.

Certain forms of TDD often make use of automated testing tools to facilitate code validation and increase the volume of tests that can be performed. A number of automated testing tools exist with rulesets devised to evaluate conformance with many of the WCAG 2.0 Success Criteria. Such tools can drastically decrease the time necessary to write and perform tests while improving the ability to identify flaws. However, in many cases, whether related to accessibility requirements or other functional requirements, some evaluations still require human cognition to determine whether certain features meet the requirements.

Given the automated testing tools currently available and the nature of some of the WCAG 2.0 Success Criteria, no amount of automated testing can completely eliminate the requirement for manual inspection of certain ICT content. Therefore, even high-functioning teams effectively using automated testing will require strong familiarity with accessibility in order to incorporate manual tests to evaluate conformance with all of the accessibility requirements.

Provide Section 508 subject matter expertise

Developers should be able to build accessible ICT and testers should be experienced in validating that it conforms to accessibility standards. Nevertheless, it may still be necessary to consult subject matter experts to confirm conformance or nonconformance to standards, help troubleshoot and remediate issues and evaluate the risks associated with non-conformant work products.

Experienced members of the development team may serve as accessibility subject matter experts, or the development team may need to rely on subject matter experts external to the team. Each development team should identify, based on the overall skills and experience of the team, how to distribute the time and attention of the accessibility subject matter experts throughout the development process to assist when necessary and promote full compliance with accessibility standards.

Major Differences Between Agile and Waterfall Development

This paper does not seek to promote Agile over traditional waterfall development. Like waterfall, Agile also has distinct benefits and drawbacks. IT accessibility practitioners, Section 508 Coordinators, and Program Managers can take better advantage of benefits and mitigate drawbacks to promote accessibility by understanding them.

Benefits of traditional waterfall development

Benefits of Waterfall	Accessibility Implications
Well-defined milestones and deadlines	• Makes testing timelines more predictable • Helps Section 508 Program Managers coordinate subject matter experts and testing resources across a portfolio of projects and activities • Lends to the ability to coordinate IT accessibility resources under a more centralized management structure • Helps facilitate governance and review of conformance at specific development milestones
Standardized documentation	• Facilitates location of accessibility-related information in standard process and system documentation (e.g., requirements, test plans, milestone reviews) • Helps facilitate governance and review of conformance at milestones via documentation of conformance assessments

Drawbacks of traditional waterfall development

Drawbacks of Waterfall	Accessibility Implications
Difficult, costly change	• Because accessibility testing often occurs late in the project lifecycle and only at specified milestones, accessibility-related defects often go unnoticed until after completion of accessibility testing • The later in the development lifecycle any defect is identified, the costlier it is to remediate; accessibility defects, in particular, can require significantly more time and effort to remediate than for those issues identified earlier in development; accessibility defects that make their way to production become exponentially costlier to remediate
Slow delivery and subsequent release schedules	• Because waterfall development projects tend to deliver entire collections of identified features and functionalities in a single production release, accessibility issues identified for remediation are typically grouped with similarly large collections of enhancements or modifications targeted for subsequent releases • Accessibility defects tend to have longer lifespans while awaiting the next release to include fixes to address the accessibility issues

Benefits of Agile methods

Benefits of Agile	Accessibility Implications
Adaptability	• Shorter development cycles with smaller, incremental iterations of functionality provide the ability to reprioritize requirements to respond to evolving project objectives, including the ability to prioritize accessibility issues when identified
Immediate user feedback	• Because Agile projects release functionality in small increments, users, including those with disabilities, begin to use the functions and features earlier in development and can provide feedback to developers • Agile teams can identify and remediate accessibility issues in early development rather than perpetuating the defects throughout later stages of development
Quicker delivery and shorter release timelines	• Accessibility-related defects, once identified, don't have to wait for a large collection of enhancements and modifications as part of a large subsequent release • Shorter, more flexible release timelines can more easily accommodate updates to specifically address accessibility issues

Drawbacks of Agile methods

Drawbacks of Agile	Accessibility Implications
Ambiguous timelines	• While one of the commonly held principles of Agile development instructs teams that each product increment should result in a "shippable" product, actual release schedules often fall to the whim of the development team, which can make it difficult to predict precisely when accessibility enhancements will move to production • Unclear release schedules can make it difficult to coordinate IT accessibility resources for those organizations that manage accessibility subject matter expertise under a more centralized structure • Organizations or project teams that attempt to apply a waterfall-oriented accessibility test process within Agile can find it difficult to conduct accessibility testing as a separate activity (i.e., attempting to test an application from top to bottom through a comprehensive accessibility test at each iteration); identifying Success Criteria applicable to only those elements developed and delivered within a particular product increment will enable the team to test only what is necessary to test as part of an integrated development and test process
Dependence on team's skills	• Because agile teams tend to be smaller (4-9 developers/testers), some team members may need to perform more than one role, and not every team member may have specific knowledge of accessibility best practices • Accessibility subject matter expertise can become spread thin, especially across Agile teams; naturally, training for all team members to familiarize them with accessibility requirements can mitigate the need for consultation with accessibility subject matter experts
Neglect of documentation	• While organizations may still enforce documentation requirements, and Agile development does not preclude robust documentation, the ideology nevertheless focuses on functioning work products over documentation; inadequate documentation of accessibility requirements and test procedures can lead to inadequate incorporation and validation of accessibility

Keys to Success in Implementing Agile Accessibility

To summarize, incorporating accessibility in Agile development methods will succeed when development teams view accessibility as an integral part of the process throughout the entire development lifecycle.

Key points of integration to successfully promote accessibility in Agile development:

- Include accessibility conformance in the "definition of done"
- Incorporate accessibility standards in key artifacts
- Minimize subjectivity by following a standard accessibility test process
- Follow TDD practices
- Provide Section 508 subject matter expertise to development teams, and ensure that all team members have adequate knowledge of accessibility requirements

Copyright notice

Integrating Accessibility and Usability Testing for Educational Content

Kris Anne Kinney

Accessibility Specialist
Educational Testing Service
660 Rosedale Road, Princeton NJ 08620
kakinney@ets.org

Jennifer Grant

Sr. Research Systems Specialist
Educational Testing Service
jgrant@ets.org

Jessica Silletti

Project Manager
Educational Testing Service
jsilletti@ets.org

Taiysha Hobbs

Accessibility Consultant/Tester
Accenture
800 N Glebe Rd #300, Arlington, VA 22203
taiysha.hobbs@accenturefederal.com

Abstract

Conforming to WCAG (Web Content Accessibility Guidelines) is but one part of achieving usable accessibility for educational content, specifically interactive tasks common in educational assessments. Conformance testing, through a combination of automated and manual testing, can only determine if technical accessibility requirements have been met. Usability for assistive technology users can only be reached through an understanding of the nature of the task and the characteristics of the user, expert review, and evaluation by target users of the content. Of particular concern in complex tasks, is the impact of the accessible implementation on cognitive load. For complex interactions, the user of assistive technology is in effect dealing with two user interfaces — that of the task itself, as well as that of the assistive technology. In education and other domains, it is vital to go beyond the baseline of technical WCAG compliance and evaluate interfaces for their usability by assistive technology users.

A Comprehensive Approach to Testing

We describe a comprehensive approach which uses a mix of automated, manual, and usability testing when evaluating accessibility. Automated accessibility testing can provide a the first pass of testing, but must be followed by manual review, . During manual testing, content is carefully examined using various assistive technologies and test methods to ensure compliance with all relevant WCAG recommendations. Assistive technology testing includes screen reader review with various user agents, as well as reviewing content for accuracy on refreshable braille displays. Other assistive technologies are used, for example, screen magnification and read aloud tools as appropriate for content. Other testing methods include ensuring interactive content is accessible and operable via the keyboard, thorough code inspection to verify that the correct markup has been used, examination of color contrast, and testing of specific interface elements, such as modal dialog boxes.

The extra step of performing usability testing helps to ensure that the content presented to a user of assistive technologies is, in fact, accessible and usable. Content that complies with the letter of WCAG but still presents a significant cognitive load (White & Hakkinen, 2017) or usability obstacle in its presentation and interaction may be deemed effectively inaccessible.

In the following sections, we describe the overall approach to accessibility testing and then focus on the role of prototyping and usability testing.

Identification and Prototyping

The first step in accessibility prototyping process is identifying, at an early stage, content that may pose accessibility and usability challenges. Working closely with content authors and alternate format specialists, the accessibility team examines which content or tasks present the greatest challenges from an accessibility standpoint, taking into account whether there are aspects which are novel or lacking established accessible design patterns. Content identified in this process become candidates for prototyping and usability testing. For a given content type or task, one or more accessible, web-based prototypes are developed using HTML, CSS, and JavaScript. To ensure the most consistently accessible experience with all user agents and assistive technologies, proper HTML semantic markup is used whenever possible, and WAI-ARIA attributes are added only when needed.

The accessible prototypes serve a dual purpose. Besides their use by participants in usability studies, prototypes that are validated in those studies serve can as an in-house reference for application developers.

Usability Studies

Once we have identified and prototyped an interaction model that is complex or that introduces unique features that are not likely to have been previously encountered by an assistive technology user, we conduct usability studies to evaluate whether the approach taken is appropriate.

We recruit participants for these studies based on the target user population, the nature of the accessibility features being evaluated, and the specific assistive technologies being used. When overall accessibility is being considered, we may recruit participants representing a broad range of disabilities. For other studies, we may recruit only participants who are blind or have low vision. Participants are asked to interact with the prototypes and rate their experience on a usability rating scale. During task interaction, think aloud protocols are used to gain a better understanding of the participants' interaction with the prototype. Critical incidents or difficulties during the interaction are logged by an observer, and are explored with the user during a post session de-brief.

If the feedback received is positive, the developers will use the accepted prototype as a model when implementing changes. Negative feedback may result in a minor change or occasionally a complete redesign of the accessible prototype, followed by another review by the relevant parties to ensure integrity of the task and technical feasibility. An additional study may be conducted to verify the issues have been addressed.

Automated Accessibility Testing

Automated accessibility tools, such as WAVE or aXe, are used to quickly identify obvious WCAG violations or warnings and provide an explanation of each issue, as well as information on how to fix each problem. Those issues that cannot be immediately or easily remedied will receive closer attention during subsequent manual testing and code reviews.

Evaluating Keyboard Accessibility

To ensure that content is accessible not only to screen reader users, but also users with mobility restrictions, we test using a keyboard-only approach. We ensure that interactive elements can be reached via the Tab and Shift+Tab keys, logical navigation order is maintained, visual focus indicators are present and discernable, and interactive elements can be activated via the appropriate key (the spacebar or the Enter key). In addition, an interaction that relies on an excessive amount of tabbing will be flagged as problematic.

Evaluating with Screen Readers and Refreshable Braille

A key step in evaluating accessibility is to utilize various screen readers, such as JAWS, NVDA, and VoiceOver, We also test it using either a refreshable braille device and/or software such as the JAWS Braille Viewer. We have found instances where content announced correctly using the screen reader text to speech synthesis but did not render correctly on the braille display. Depending on the target delivery environment, we may limit assistive technology testing to a predefined, recommended screen reader/browser combination. While using JAWS, we test content in both browse and forms mode. Such testing is done to ensure the following:

1. The user can properly navigate to all content
2. Logical navigation order is maintained
3. Form controls, such as buttons and input boxes, are properly labeled

4. The user can successfully activate or interactive with form controls
5. Form controls and other elements are behaving in predictable ways
6. A change to a form control’s state is properly announced and presented
7. Notifications are appropriate and are properly announced and presented
8. Links that open a new browser tab are clearly indicated as such
9. Dialog boxes and other popups are behaving correctly
10. Regions are properly announced and presented and required regions are present
11. The interaction does not present a significant cognitive load and is not unnecessarily verbose
12. The user can successfully review any responses that have been entered.

The testing results are shared with content and development teams, and issues are filed in the appropriate databases for tracking and resolution.

HTML Code Reviews

Once an accessible prototype has been successfully vetted by a usability study, it is then implemented by the application development team. Once implementation is underway, and the application is available for testing, a complete code inspection is done to ensure that the proper elements and attributes are being used, the attribute values are being set properly, the keyboard handlers are working properly, etc. In short, it is reviewed to ensure that the markup matches the accepted accessible prototype.

Typically, focused code reviews can be used for the purpose of troubleshooting issues. Unless the solution is obvious – a missing label on an input box or missing alternative text on an image, for instance – a thorough inspection of the code surrounding the area may be required to pinpoint the cause of the issue. A key process in the code inspection is to determine whether an issue is caused by either improper coding, a browser issue in which the code is correct, but the browser is not correctly providing accessibility specific information via the accessibility API to the assistive technology, or the assistive technology itself has an issue and is not correctly rendering the content. Depending on where the issue lies, problem reports may be filed with either the browser or assistive technology vendor. In the case that it is a coding issue, recommendations will be made to the developer and when appropriate code samples will be provided.

In addition to testing a specific accessible implementation, code reviews are useful to identify other issues, specifically with respect to WCAG 2.0 4.1. Such a review may uncover things such as deprecated elements and attributes, invalid use of WAI-ARIA attributes, improperly nested elements, and general HTML validation errors. While such coding issues may not result in perceivable issues to end users with or without assistive technology, they may result in unpredictable behavior in current or future versions of browsers and assistive technology.

Challenges

Applying the WCAG standards in educational content, with an ever-expanding array of innovative tasks and simulations, can pose significant challenges, and it is important to understand how to apply WCAG success criteria appropriately and to know when to pursue a different course of action. Simulations, for example, may contain dynamic, visual information which require multiple approaches for achieving accessibility, from textual descriptions, to sonification, to the application of emerging tactile or haptic displays.

With the upcoming WCAG 2.1, it is important to understand the potential impact of new success criteria on assessment content so that we can inform designers, developers, and accessibility testers as to how to design, apply and evaluate them. Where interactions break new ground in terms of accessibility, those building those applications should share with the web accessibility standards groups, browser and AT vendors where gaps exist in the technical standards, testable success criteria, and implementation techniques.

References

Chandrashekar, S., Stockman, T., Fels, D., & Benedyk, R. (2006, October). Using think aloud protocol with blind users:: a case for inclusive usability evaluation methods. In *Proceedings of the 8th international ACM SIGACCESS conference on Computers and accessibility* (pp. 251-252). ACM.

Krahmer, E., & Ummelen, N. (2004). Thinking about thinking aloud: A comparison of two verbal protocols for usability testing. *IEEE Transactions on Professional Communication*, *47*(2), 105-117.

White, J., & Hakkinen, M. (2017, March). *Perceptual and Cognitive Considerations in Accessible User Interface Design*. Presentation at the 32nd CSUN Assistive Technology Conference, San Diego, CA.

Copyright notice

Using JAWS as a Manual Web Page Testing Tool

Ryan Jones

Program Manager
Freedom Scientific (a VFO company)
11800 31st Court North
St. Petersburg, FL 33716
rjones@vfogroup.com

Rob Steinberger

Director of Software Engineering
Freedom Scientific (a VFO company)
rsteinberger@vfogroup.com

Abstract

JAWS for Windows (JAWS) is screen reading software developed to allow someone with little or no sight the ability to use a computer. Though it is specifically designed for the end-user, many people use it as a tool for testing web page and document accessibility. Join us as we learn about how with knowledge of a few keyboard commands and settings, JAWS can become a powerful tool for testing web content for accessibility. In addition, we will learn why manual testing with a screen reader is critical to achieving both technical and functional accessibility. Learn how with knowledge of just a few keyboard commands, you can simulate the experience of screen reader users navigating your content. We will also demonstrate upcoming enhancements that will dramatically increase the efficiency and productivity of people using JAWS as a testing tool.

Using JAWS as a Manual Web Page Testing Tool

Introduction

Accessibility is more than a set of guidelines and technical requirements. It is the ability for someone to access content and functionality in an efficient and meaningful way. Though fantastic automated testing tools exist, there is still no replacement for manual testing with assistive technology tools to insure that web content is functionally accessible and usable. In particular, testing with screen reading software is key to insuring content is usable by someone with a visual impairment.

JAWS for Windows (JAWS) is a screen reading tool that provides visually impaired users access to computer technology and information. It has a long history of providing a powerful screen reading experience for users and is also being used for manual web page testing by developers, testers, and designers. JAWS provides a real-world means of insuring that content can be used and understood by screen reader users.

JAWS is an extremely powerful screen reading tool with a large feature set and a significant amount of user-configurable settings. Using JAWS as a web page testing tool however only requires the use of certain features and settings related to how JAWS interacts with HTML based content. This allows someone who does not use JAWS on a regular basis the ability to learn certain functionality of JAWS without having to be as experienced as most end-users of the product. Developers for example can quickly turn JAWS on to test the screen reader functionality of a widget or control they are working on. QA testers can quickly test functionality of a data table or grid to insure that proper header information is being spoken. With proper training and an understanding of how JAWS works, anyone can use it for web page testing. There is no need for only a few specially trained individuals to perform accessibility related testing.

A note about browsers

At the time of this writing in September of 2017, JAWS currently works best in either Internet Explorer 11, Firefox (latest version), or Google Chrome (latest version). It is important to keep both JAWS and the web browser up to date to insure the best results. The current release version of JAWS is typically updated every four to six weeks and often the update includes enhancements to web browser support. Firefox and Chrome are also updated on a regular basis with accessibility related enhancements. At this time, it is not recommended to use Microsoft Edge for manual web page testing with JAWS. While the current version of JAWS has initial support for Edge, users are not recommended to use it as their primary browser at this time.

Configuring JAWS for testing

Many of the settings in JAWS exist to enhance the experience of the end-user. However, several settings are important for testers to understand in order to have an accurate and positive test experience. The following settings can be found and changed from within the QuickSettings Dialog box activated by pressing INSERT+V while focus is in a web browser.

- Document automatically reads (unchecked) – When checked, this setting causes JAWS to automatically begin reading information on the page when the page loads. It is recommended this setting be turned off to prevent focus from beginning to move through the page when it first loads.
- Filter consecutive duplicate links (Unchecked) - This option controls whether consecutive links pointing to the same location, one graphical and one text, are filtered. It is recommended this setting be turned off so that all text and image links will be spoken by JAWS.

Additional settings such as voice rate, typing echo, and keyboard layout can be found by activating the Startup Wizard from within the JAWS Help menu.

Note that if using a computer that does not have a numbers pad it is important to change the JAWS keyboard layout to laptop mode. By default, JAWS is configured for desktop mode which heavily relies on the numbers pad. For example, many commands use the INSERT key as a modifier. If the computer does not have a numbers pad or specific INSERT key, those commands would not be available. Setting the keyboard layout to laptop mode gives the option to use the CAPSLOCK key as the modifier rather than INSERT. The keyboard layout setting is easily found in the Startup Wizard dialog box activated from the Help menu.

Navigating a web page

Learning how screen reader users navigate web pages may help in determining the best way to conduct testing. There are two fundamental ways in which users often navigate web page content.

1. Reading the page line by line – In this technique, the user moves the JAWS virtual cursor to the top of the page and uses the DOWN ARROW key to move line by line. JAWS reads each line and identifies the type of element (i.e. link, button, edit box, etc.). The order in which the cursor moves through the page is determined by the document object model. This is a slow and tedious process (especially on a large page) but is the only way to insure JAWS navigates to all elements on the page.
2. Navigating to different elements – This technique involves using various JAWS shortcut keys that jump directly to elements such as links, images, tables, etc. For example, to quickly determine what heading elements are present, press the letter H which will move JAWS to the next heading. Other examples include T for table, G for graphic, E for edit box and L for list. Each letter of the alphabet is tied to a specific element or shortcut within JAWS. Navigating directly to elements is how the majority of JAWS users navigate web content. For example, JAWS users will often move to the top of the page and begin pressing H to navigate to the headings on the page.

In addition to using single letter commands to navigate, JAWS can generate lists of elements. Each list is displayed in a dialog box and also serves as a way to visually see the various elements on the page. Press the JAWS command INSERT+F3 to display a dialog box containing a list of all the various types of elements that can be displayed in a list form. Press ENTER on an element and a list of those elements on the current page will be displayed.

As part of the testing process it is important to understand what JAWS is saying and why it is saying it. For example, as JAWS reads a form field label it is important to know where the spoken information came from. Is it from the label element, title attribute, ARIA label, etc.? Knowing this information may help insure that the form field is communicating the appropriate information to JAWS and if not, why. This information is often useful when determining the priority of element attributes and how they are processed by screen readers. JAWS has a variety of settings that control which attributes of an element are spoken and learning to use these settings can be a valuable tool. Press INSERT+V when focus is in a web browser to activate the QuickSettings dialog box containing these settings.

Tables and forms

Navigating tables requires the use of specific JAWS commands designed to navigate and read information in tables. When focus is in a table, press INSERT+SPACE BAR followed by the letter T to activate the JAWS table layer mode. Once table layer mode is active, pressing the directional arrow keys will navigate through the cells in the table. JAWS will speak the information in the table along with any defined row and column headers. As focus moves horizontally through the table, JAWS speaks the column header along with the cell data. As focus moves vertically, JAWS speaks the row header along with the cell data. If no row or column headers exist, JAWS will simply speak the information in the current cell. Press ESCAPE when finished to exit the table layer mode and continue navigating through the page. As an alternative to activating the table layer mode, press and hold the CONTROL and ALT keys while pressing the directional arrow keys. JAWS will move through the table and speak defined row and column header info similar to the table layer mode behavior.

When using JAWS to work with form controls it is first important to understand a concept called Forms Mode. When Forms Mode is on, pressing alphanumeric keys on the keyboard will no longer activate shortcuts to various page elements such as headings or tables. The alphanumeric keys are passed directly to the browser as if JAWS were not running. This allows the user to enter characters in an edit box or text area without activating traditional navigation shortcuts. Forms Mode is required when focus is on a control requiring text entry or when focus is on a combo box or list box.

By default, JAWS turns Forms Mode on and off automatically when focus lands on an edit box or when tabbing through form controls that require Forms Mode to be active. JAWS indicates that Forms Mode is active with a high pitch pop sound. A slightly lower pitch pop is used to indicate that Forms Mode turned off. If focus is on a control that requires Forms Mode and it is not turned on, press ENTER to activate it. Use the QuickSettings dialog box to change settings related to whether or not forms mode activates automatically.

When testing form controls there are often three main things to check.

- Does each control have a meaningful label
- Is the tab order of the controls logical
- Can each control be manipulated with the keyboard

One of the most effective ways to quickly check the labeling and tab order of form controls is by activating the JAWS list of form controls. Press INSERT+F5 to display a list of all controls on the page. They are listed in tab order and the label displayed in the list will help indicate whether or not the control has an appropriate label.

Pressing TAB to move through form controls is also an effective way of checking for accessibility. When focus lands on a control, JAWS should announce the type and name of the control along with the current state or value. Keyboard commands such as SPACEBAR to check or uncheck a radio button or check box can then be used to test for keyboard operability. Press INSERT+TAB when focus is on a control to hear JAWS repeat the information about the control.

Getting JAWS help

A variety of help on using JAWS with a web browser is available. The following commands may be useful for quickly accessing various help topics.

- Hotkey help – Press INSERT+H from within the web browser and JAWS will display a list of specific JAWS commands for navigating web page content.
- Context specific help – Press CTRL+HOME to move focus to the top of the page and press INSERT+F1 to display information about the current page. Pressing this command while focus is on a web page element such as a link or button will provide specific information about that element.
- Browser specific help – Press and hold the INSERT key and press F1 twice quickly to load the JAWS help system and open the topic specific to the web browser being used.
- Display element information – Press INSERT+SHIFT+F1 to display technical information about the currently focused web page element. The information is derived from the document object model and also contains information about parent elements. This command is often very useful in determining why JAWS treats an element a certain way.

Speech History

The speech history feature displays a visual transcript of the last 50 things that JAWS spoke. This is often very useful as the information can be copied and pasted into other documents such as a test report or defect tracking tool. In addition, it allows users the ability to visually see what JAWS has been speaking which is often helpful for testers who don’t frequently use screen reading software. Think of speech history as a visual transcript for what JAWS is speaking.

Press INSERT+SPACEBAR followed by H to activate the speech history window. The information is displayed in the order in which JAWS spoke it. The most recent information is at the bottom. Pressing INSERT+SPACEBAR followed by SHIFT+H will clear the history buffer and start it over. Clearing the buffer may be useful before testing a portion of the page as it will eliminate clutter from other text that was spoken.

Conclusion

Using JAWS as a tool to manually test web pages is one of the most effective ways to insure that content is usable for someone with a visual impairment. Both visually impaired and non-visually impaired testers can learn to interpret what JAWS is speaking about web page content and apply it to the overall accessibility of a page. In addition, using JAWS features such as the speech history can make it easier to document and report accessibility failures and demonstrate to others how these failures impact usability. When a truly blended approach of both automated and manual testing is used, the end user will be given a technically accessible and functionally accessible experience.

Copyright notice

Continuous Accessibility Inspection & Testing

Alistair Garrison

Level Access
1600 Spring Hill Road, Suite 400
Vienna, VA 22182, United States
alistair.garrison@levelaccess.com

Modern software development practices

Knowing that things change (and oh how fast things change in the software industry especially!) the software industry has left behind Waterfall, with its out-dated QA gates, demands and doctrines, and embraced Agile development practices. Agile development practices, when properly followed, allow teams to absorb changes in requirements – quickly and easily…

Looking at Scrum, for example, as a very well-known Agile project management framework for software development projects we see that it promotes the use of small teams, in small sprint cycles with the intention of producing regular amounts of working code – which users can review and provide their feedback on.

Changes to the infrastructure to support the software development practices

DevOps (from 'development" and "operations") is really the collective noun for those changes to an organisation's infrastructure that enhance its Agile abilities in order to deliver software, of the highest possible quality, in the shortest possible times.

In a DevOps organisation the traditional concept of a "software release" cycle, changes over time, into an almost continuous cycle of service improvement. DevOps focuses on the heaviest possible leverage of automation and monitoring in the delivery "conveyor belt" processes:

- Requirement specification;
- Development;
- Software integration (Continuous Integration);
- Deployment;
- Testing (Continuous Testing in all the above);

Continuous Integration (CI)

CI is the practice of merging all working copies of code held in central repositories several times a day. The automatic execution of automatic tests is possible during different phases within the CI process, as just one of the automatable processes available through modern CI Servers. The typical CI process is:

- Developer develops
- Checks-in changes
- CI fetches changes
- CI builds
- CI tests
- CI notifies – Build Success / Failure

Continuous Testing (CT)

CT is the process of executing automated tests within areas of the software delivery "conveyor belt" in order to find and remove bugs at the earliest possible point. Continuous Testing can be done at lots of testing points, e.g.:

- Wireframe creation
- Ticket creation (Jira)
- Pre-commit – of code into a repository
- Commit – of code into a repository
- Build – Unit Tests / Acceptance Tests

And, as such Continuous Testing activities can be hooked into a broad range of tools, not only Continuous Integration.

Changes to testing activities to support Agile + DevOps

Testing typically breaks-down into two types:

1. Functional testing: checking software to ensure that it has all the required functionality specified in the functional requirements documentation;
2. Non-functional testing: checking the way a system operates, rather than its specific functionality.

With the emphasis on as much automation as possible within DevOps, many organisations are increasingly looking to break-down or change their current functional, and increasingly non-functional, testing procedures into as many fully automatable pieces as possible – so they can be added into their Continuous Testing schemes.

The main types of automatable tests generally produced in support of this are:

- unit tests: a unit test exercises a coded function to determine if it behaves as expected;
- end-to-end acceptance test: an end-to-end acceptance test exercises a feature, or more typically a part for a feature, through simulated user interaction with a user interface, to determine if software behaves in an acceptable manner.

Changes to the role of the QA Tester to support Agile + DevOps

There is a growing indication that the role of the QA Tester has changed within organisations embracing Agile + DevOps. No longer the gatekeepers at specific quality gates, or just bug finders, the QA Tester's role now pro-actively includes:

- Responsibility for continuous improvement and quality tracking across the entire software delivery "conveyor belt". The focus being on the identification / resolution of product and process issues, or potential areas of issue.
- Responsibility for automating existing processes; or changing existing processes to make them automatable. With the focus being on removing as many of the manual processes as possible from the software delivery "conveyor belt" – to make delivery as fast as possible.
- Responsibility for influencing both development and operational processes. Advocating:
    - best practices which are known to decrease bugs (like Test Driven-Development);
    - the adoption of tooling that:
        - supports those best practices (Unit Test frameworks; BDD Test frameworks consuming requirements written in Gherkin, etc.);
        - finds very specific issues.

Changes to software development practices to support Continuous Testing

DevOps is clearly designed to alter environments within organisations to speed up delivery, putting significant emphasis on the ensuring that the delivery pipeline is as automatable, and able to be monitored, as it can be.

As such, you can already see organisations moving to:

- Write product requirements in Gherkin (see below);
- Adopt coding languages that have good static inspection tools (linters, etc.);

- Adopt standardised and "most able to be tested in an automatable manner":
  - UI Component style guidelines;
  - UI Web Components (Polymer / Web Components);
  - UI Frameworks (e.g. Angular2, React.js,);
- Adopt test libraries / frameworks which "play nicely" with CI Servers.

Gherkin

It is a **Human Readable + Machine Readable**, Domain Specific Language that lets you describe software's behaviour without detailing how that behaviour is implemented. Requirements in Gherkin can be run through a linter (see below), and grammar checked automatically.

The magic with regard to capturing your requirements in Gherkin directly is that you can then use those exact same requirements as automated acceptance tests in Behaviour-Driven Development (BDD) frameworks (e.g. Cucumber).

Linter

Any tool that flags suspicious code / formatting in static source code.

Web Components

W3C's Web Components are fully-encapsulated easily re-usable and incorporable UI components.

It is important to emphasis the envisaged changes to an organisation's processes that Web Components will certainly bring. In the past, there was a tendency for organisations to standardise on the look and feel of components, but leave the coding to developers. This lead, in many cases, to the real-world problem that an organisation might have several websites which looked and felt the same, but which were built completely differently. From a DevOps point of view, this is inefficient, as you could have multiple sets of similar tests to maintain. Which all take precious time to execute.

Web Components now enable a company to have UI components that are "write once, run anywhere" ready.

Incorporating new testing types into Continuous Testing

Moving forward, we discover new areas within our IT systems which become important to test:

- Performance;
- Security;
- Accessibility;
- Some future test type, etc.

A similar pattern can be followed each time when incorporating a new testing area into an organisation's existing testing scheme. Typically, an organisation with start by:

- breaking down what testing they'll need to do into functional / non-functional pieces;
- identifying what (in terms of artefacts) they'll be testing;
- identifying where that testing will take place;
- identifying if their current tool-chain, within their current processes, can support that; or:
    - if changes will be needed to those existing processes or tools;
    - if new tools or processes or tools will need to be created;
    - if changes to what is being tested will enable better automation.

And, when we think about overall process improvements, this same process can also be run against pre-existing areas of testing to discover if more automation can be achieved by using the new / changed tools or processes.

Accessibility testing – just another test type!

Functional / Non-functional pieces

Historically, the functional and non-functional aspects of Accessibility testing were amalgamated within end-user testing – which was probably fine in Waterfall. But, now with Agile + DevOps we need to see them properly pulled apart. There are clear functional and non-functional components in Accessibility Testing:

- **Functional Accessibility Testing**: checking that people with disabilities, with required as necessary support from one or more Assistive Technologies, can undertake each relevant function specified in the requirements.
- **Usability testing for people with disabilities (by preferably people with disabilities)**: checking the usability of each relevant function specified in the requirements from the point of view of people with disabilities (with required support from one or more Assistive Technologies).

Emphasis on automatic test coverage

With the ultra-high value a DevOps organisation places on automation, it is, however, more and more likely that we will see shifts occurring in software design / build in order to maximise the automatic test coverage of a software product.

From an accessibility perspective, for example, if you have two WCAG 2.0 Techniques that broadly achieve the same thing, and one is much more testable in an automated manner than the other, then the more testable technique is likely to win-out for implementation.

Non-blocking usability testing

That said, it is still very necessary for organisations to ensure that usability is also checked, but in a DevOps organisation this might well be in the form of a non-blocking "conveyor belt" style dip-test, rather than the historically blocking quality gate.

Take for example a QA Tester who was also a certified "Trusted Tester". They would now have responsibility for testing a "sample" of features within products; both for technical accessibility, and usability.

Such a non-blocking usability testing strategy could be used to inform / improve the overall software delivery conveyor belt – so it inherently starts to produce accessible UI interfaces which are highly usable; without becoming a delivery bottleneck.

What build artefacts might contain an accessibility issue?

It is important to try and identify as many build artefacts as possible that might contain accessibility related issues. With regard to accessibility testing, a good starting list would be:

- User stories relating to UI components / features;
- UI component style guide;
- Static source code:
    - relating to the creation of UI components;
    - relating to containers for UI components e.g. templates;
- Data that's bound with UI components – possibly from a Content Management System;
- Rendered DOM content from dynamic code interactions.

UI component style guide

The UI component style guide contains examples of each type of UI component included in a web-based product. If best practices are followed, each UI component is usually provided, typically in its various different envisaged styles, in its own vanilla web page.

Importantly, the content of the UI component style guide should be covered by unit tests and BDD tests – which in-turn allows coverage by accessibility tests (see below).

Where will this accessibility testing take place?

The QA Tester should identify the most convenient, and useful, point for testing. When thinking about running automatic tests, it is useful to look for event hooks within systems – which when triggered can run functionality of differing types. If we are thinking about the test artefacts we've already identified for accessibility testing the natural testing points would be as follows:

- When a ticket is raised in Jira, the text within that ticket could be assessed.
- When static code is being worked on, the static code could be automatically assessed whilst the developer is typing.
- When static code is committed, the pre-commit event in CI servers could trigger a number of smoke tests to run; rejecting code which was found to contain issues.
- When the code is built, the CI build process can test the rendered DOM:
    - Smoke Test (Happy Path)
    - Unit Tests
    - Acceptance Tests
- When working code with working features becomes available, a “conveyor belt” sample dip test can be run to assess the usability of the product.
- When the product is released, live monitoring tools can continue to watch it’s accessibility; and provide “exploratory style” real-world uses cases back into the testing system to enable increased coverage.

Tooling for accessibility testing

Static Artefact Inspection

An extension to a ticket management system (like Jira) that “on ticket creation” analyses the text of the ticket to determine if the ticket:

1. Is formatted correctly (Gherkin); and
2. Is for a UI Component / UI feature; and if it is:
    - Determines if accessibility is mentioned; and
    - Determines if behaviours relating to the operation of the component using only the keyboard have been provided.

Static Code Inspection

Intellisense is a feature built into many Integrated Development Environments (IDEs). The feature provides coding hints, or warnings, whilst a developer types in the code.

As mentioned, a linter is any tool that flags suspicious code / formatting in static source code. By hooking into the pre-commit event, in source control servers, a linter (chosen specifically for your coding standard) can be run. Several linters exist for popular UI Component Frameworks which allow the source code for UI components to be checked for accessibility issues.

Render code testing

By hooking into build events, in CI servers, different QA testing Unit testing frameworks / BDD testing frameworks can be run.

A number of standalone Accessibility Testing Engines are now available (some as Open Source projects) which are built in JavaScript, and designed to allow a full range of accessibility tests to be run on the rendered DOM.

The old “only 25% of accessibility issues are testable through automatic tests” is also quickly dying (thankfully, for clean uptake in DevOps organisations), as several areas of Machine Learning can be readily applied to the accessibility testing space – for example, classifiers, natural language processing, image recognition, etc… Integration packages (again, some available as Open Source projects) allow these Accessibility Testing Engines to be easily included in test files.

The full power of these Accessibility Testing Engines becomes available when they are run through QA testing Unit testing frameworks / BDD testing frameworks – in effect, you use the QA framework to move the UI component to each of its states; and you fire the tests when each of those states has been reached – providing complete test coverage of user tasks.

Sampling

Sampling is a very efficient way to monitor if a delivery “conveyor belt” is producing products which are of the right quality. In effect, this covers not only the idea that the building processes build the right quality products, but also that the Continuous Testing done within an organisation captures “all” of the tested for issues.

In recent years, accessibility testing has been moving beyond the use of spidering tools, as they are generally limited to only being able to test content in one DOM state – the page load DOM state. For full coverage of all states you really need an Accessibility Testing Engine, integrated with a QA testing framework (as mentioned above).

In recent years, accessibility testing has been moving beyond the use of spidering tools, as they are generally limited to only being able to test content in one DOM state – the page load DOM state. For full coverage of all states you really need an Accessibility Testing Engine, integrated with a QA testing framework (as mentioned above).

However, spidering tools work well as sampling tools, as you are simply looking to identify if the systems you have in place produce error free products. And, for this purpose it does not matter that you only test one DOM state.

A CI-server can be used to initiate a sampling tool – ideally, however, a sampling tool would be set to run in a scheduled job – weekly, monthly etc, rather than on-build. The reporting from such sampled monitoring could be made available through the spidering tools (as reports), or via the CI-server dashboard.

Live monitoring tools

The job of creating high-quality web applications, does not, however, stop when they are launched – even if your Continuous Accessibility Testing system already provides you with a good deal of confidence in the accessibility of your product.

The ability to receive test results from all DOM states reachable by users, whilst they are using your live products / services is now achievable through embedded analytics-style scripts. These scripts enable the QA team to detect all real-life use cases of the live product / service, in a similar, but more encompassing manner, than exploratory testing. The reporting from live monitoring is generally available via a dashboard.

Summary

In summary, organisations practicing Agile + DevOps now have well-defined patterns for Continuous Delivery, and as an integral part of Continuous Delivery - Continuous Testing.

The platforms supporting Continuous Delivery enable new test types to be more easily accommodated into testing schemes, with QA testers becoming the enablers of the required business change. The selection of style choices, and building frameworks, based on their testability naturally enables higher and higher test coverage by automatic tests – with accessibility testing being no exception.

And, with the creation of Accessibility Testing Engines (with every increasing fully automated test coverage), and their designed integration with QA Testing Frameworks, the possibility of Continuous Accessibility Inspection and Testing becomes a real, very achievable “low-bar” possibility for organisations.

Accessibility testing, is at the end of the day, just another test type to integrate into a broader Continuous Testing scheme – and realistically should be treated as such.

Copyright notice

From Desktop to Mobile Device: Who Is Left Behind? Moving Toward Global Accessibility Standards

Alyson M. Muff, Ph.D.

ICF International
alyson.muff@icf.com

Joseph De Rosa

ICF International
joseph.derosa@icf.com

Abstract

Mobile devices (mobile phones, smart phones, PDAs, etc.) have changed the world of information and are slowly transforming and rendering obsolete the desktop experience. Within the last decade there has been growth in making information available and accessible to people with disabilities and/or special needs. This growth is due to general advances in technology, advances in mobile technology, governmental support (Strategic Plan for Improving Management of Section 508, 2013), and the latest update to Section 508, known as "the refresh," that went into effect March 21, 2017. With the gains and growth in making information accessible there are still those who are left behind. This paper discusses the issues still to be addressed in order to create viable standards that result in global accessibility. The harmonious integration of usability and accessibility standards and best practices is the key to creating global mobile standards that ensure no one is left behind.

Overview

Mobile devices (mobile phones, smart phones, PDAs, etc.) have changed the world of information and they are slowly transforming and rendering obsolete the desktop experience. Within the last decade there has been exponential growth in making information available and accessible to people with disabilities and/or special needs. This growth is due to general advances in technology, advances in mobile technology, governmental support (Strategic Plan for Improving Management of Section 508, 2013), and the latest update to Section 508, known as "the refresh," that went into effect March 21, 2017. With the gains and growth in making information accessible there are still those who are left behind. This paper discusses the issues still to be addressed in order to create viable standards that result in global accessibility. We believe that usability and accessibility in the mobile environment must work harmoniously in order to achieve success in creating global standards that leave no one behind.

Background

The research of the past 20 years pertaining to accessibility and usability evaluation methods is based on the evaluation of websites using desktop and laptop computers (Interaction Design Foundation 2007, Rocha, 2017). Much has changed over that time period – including a dramatic and global shift to a mobile reality. The majority of desktop computers are housed in a workplace environment and laptop computers although ‘movable’ are not mobile. The mobile reality is dynamic, migratory, and personal. This shift continues to have significant impact on users and the evolving usability and accessibility standards. With the rapidity of technological change, the need for global standards becomes more important. Usability and accessibility standards continue to coexist and as such, both need to be addressed in defining a mobile experience. As such, a brief review of the similarities and differences between the standards of the two domains is in order.

In the literature, different ways can be found of defining accessibility and its relation with usability: ISO/IEC Guide 71 (2001) defines Accessible Design as “design focused on principles of extending standard design to people with some type of performance limitation to maximize the number of potential customers who can readily use a product, building or service”. On the other end, ISO 9241- 171 (2006) and 9241-20 (2006) define accessibility in a very different way as “usability of a product, service, environment or facility by people with the widest range of capabilities”, introducing a tight connection with usability.

The Web Accessibility Initiative (WAI), founded by the World Wide Web Consortium (W3C) Web, gives a widely accepted general definition of Web accessibility: “accessibility means that people with disabilities can perceive, understand, navigate, and interact with the Web, and that they can contribute to the Web” (2008).

Accessibility is influenced by knowledge and familiarity. For example, an individual’s familiarity with a system may make it accessible while the person with no familiarity may well be lost. Accessibility is a pervasive quality in terms of systems and devices. There must be a guaranteed delivery of all parts of a system so that a sufficient level of accessibility will be maintained to avoid compromising the achievement of a result that is accessible overall (Pernici, 2006). This leads to the understanding that accessibility and usability while implicitly related are rarely explicitly analyzed together, either in the context of the Web or other computer-based systems and certainly not in the mobile context.

There are research claims that usability problems affect all users in the same way whereas accessibility problems hinder access for people with performance limitations (Thatcher, 2003; Petrie, 2009). Much of this research also maintains that there are problems typically considered as accessibility problems that affect non-disabled users. Accessibility and usability share responsibility for a successful and productive user experience. The significance of this relationship deepens in the mobile environment.

Accessibility and Universal Usability

Shneiderman proposes “universal usability” (2003) as a term to encompass both accessibility and usability, but notes that “access is not sufficient to ensure successful usage”. He defines a different ranking of accessibility in comparison with usability: accessibility is a first but not sufficient requirement to achieve **universal usability.**

Accessibility cannot be considered as the sole requirement in order to allow people with performance limitations to use interactive systems and usability cannot be considered as encompassing all the possible problems encountered by every user. This means that something that is accessible may not be usable, and vice versa. More specifically, as Petrie (2007) states, three kinds of problems affect the satisfactory use of applications, namely “pure accessibility” problems (affecting people with performance limitations), “pure usability” problems (affecting persons without limitations) and “universal usability” problems (affecting every user).

Major differences between mobile and desktop/laptop environments

What does this all mean for Mobile devices? How do the constructs of accessibility and usability mesh in a way that allows for inclusion? As a first step – let’s take a look at the major differences between mobile and desktop/laptop environments focusing on context.

Mobile is mobile. Users are on the move – transient and constantly on the go. This means that the context of use constantly changes. This is not true of the desktop/laptop context where the person is ‘in place’ and tasks are largely static or at the very least – sequential. Mobile users can swap between apps, messaging, video and the task at hand is driven largely by personal desire and/or by real-time events. Moreover, connectivity may drop in and out and even the way users hold the device may change and with that, the perspective of the screen display will change.

Simplicity: Given the changing context of mobile usage it stands to reason that users’ tasks are foreshortened and simplified given the display and implicit task space and expectations. Most mobile users are not writing their first novel on their mobile device. As a result, long, complex interactions are not natural to the device and context of use. This is not the case with the desktop/laptop where the reverse is often true and complex tasks are often the norm.

Data Generation: Mobile has led to a wealth of new personal data generation. GPS positions, pictures of the user, their friends and family, communication data between colleagues, friends, etc.

Privacy Concerns: Mobile devices run a higher risk of being lost and/or stolen as opposed to the desktop/laptop configuration. The perception, use, and ‘story space’ of a mobile device is oriented toward personal use and as such, there is increased concern related to privacy and security. Look around at people on the street and note the ‘tethering’ of person to device. That is not apparent with a laptop much less a desktop.

No Single Experience: Mobile is mobile but vastly different for every user – even the same user with several devices. The mobile experience will change based on the device and the capabilities of the device. Differences in browsers, operating systems, events, etc. underscore the need to rely on accessibility and W3C standards to afford users of a certain degree of consistency and reliability. Mobile also encompasses tablets. That's a different experience too – tablets aren't desktops but they offer more "desktop like" functionality than a smartphone does. Desktops and laptops are not standardized but there are limitations to the operating system(s), software, etc. The maturity of the static, in-place computer fluctuates less than the mobile device as does the software. These hardware and software limitations are imposed on users who may vary but have a proscribed and more limited set of options as opposed to the mobile environment.

Input Options: Mobile offers different forms of input such as touchscreens, voice inputs, user movements, etc. These are the mobile experiences that are not universally available on desktop/laptop computers. The input range illustrates that mobile users have needs significantly different than those of the desktop/laptop user.

Mobile Devices and the Challenges

Mobile devices highlight the importance of contextual factors of use including: the physical state and capabilities of the person, the physical device, the task at hand, environment; social and technological components. All of these factors have significant impact on user performance, behavior, and outcome. Moreover, all of these factors have significant impact on the accessibility and usability of the mobile device and associated applications.

As noted in our discussion thus far, the rapid advancements in the design and development of mobile devices have a significant impact on the nature of user interaction, as they can be used at any time in any location and by any one. As such there is an ever-increasing focus on the user interfaces of touch-screen devices, voice and the implications this has on the context of use and context 'of person'. The idea of accessibility has extended into many communities as people differ widely in their physical state, mental state, style of use, and social-technological context.

The next section concentrates specifically on the accessibility combined with the usability challenges related to the mobile context. These challenges must be resolved in order to obtain a product that can be adequately accessible and usable, and thus considered to be universally accessible and inclusive.

Assistive Technology on Mobile Devices

Display: Most of the built-in assistive tools on mobile devices are similar to those used on computers, however their capabilities are often lacking or hampered by the nature of mobile devices. Text-to-speech, or screen readers, are available on some devices and can be obtained via third-party software (Nokia, 2011). The downside to mobile screen readers is that they do not always work with all applications or functions of mobile devices and therefore still allow barriers to access to exist (American Foundation for the Blind, 2011). Moreover, many of these tools are contextually unaligned. For example, a screen magnifier used on a smartphone will take up the entire screen as opposed to a large desktop screen where only a portion of the screen is visible. The mobile display area is also likely to distort the space which impacts contrast and color. The quality of the mobile device screen also has impact as some devices do not offer sufficient adjustability for brightness and contrast.

Audio: Many mobile phones come equipped with voice control capabilities which can prove beneficial to those with vision or motor difficulties. According to the American Foundation for the Blind (2011) these capabilities work for a limited selection of tasks (e.g. placing a call) and but do not offer full access to the device. What that means is determining information such as battery level, incoming calls, missed calls, and voice alerts is a challenge for the visually impaired. Voice echo (where the device reads the name of the key when pressed) is not a universal mobile device feature – another limiting factor to information access. Privacy is another issue associated with audio and 'audible readouts' on mobile devices further indicating a need for expanded control of device functionality by the user.

Time Out: Mobile devices often cancel operations if input is not received within a designated interval. Users with cognitive disabilities or physical challenges would benefit from an option allowing control over the time allowance for response. Websites and applications that place time restrictions on input should consider the increased difficulty of typing on a mobile device and allow a means for extending the allotted time frame.

Navigation

Mobile devices can increase independence and a sense of empowerment (Kane, 2009). Anthony (2013) notes that users with physical disabilities are able to operate mainstream mobile devices (tablets, smartphones) despite the challenges. Less strength is required to use a touchscreen as opposed to physical buttons (2010). However, the accuracy of using directional gestures is often error prone. Multi-touch gestures are difficult for those with motor impairments and those individuals with visual impairments (low vision) have difficulty perceiving the boundaries between application icons. Blind and visually impaired users and users with decreased motor functionality avoid the use of a mouse on a desktop/laptop. The preference is to use a keyboard while listening to a screen reader when navigating through content (Estes, 2016). It is important to keep navigational techniques in mind when developing content to be viewed on mobile devices. Tagging content, sequencing headers and main sections are considered best practice regardless of the type of user and associated capabilities.

Situational Factors

There is a multitude of factors associated with mobile device use that intrude upon the experience of all users - particularly users who are physically and/or cognitively challenged. Situational impairments brought on by contextual factors can affect how users interact with a device, such as lighting, glare, noise, rain, weather, crowded spaces (Sears, 2003). When the user is in motion all of these factors serve to reduce input the speed of interaction and increase error (Yesilada, 2010). The user's ability to read information on a mobile device is impacted by motion affecting text legibility and reading comprehension based on screen size, lighting, color, and contrast.

Both usability and accessibility - while seemingly independent - are inextricably linked in the everyday use of mobile devices. Upon close inspection, the WCAG 1.0 and 2.0 standards address both usability and accessibility. These standards serve as a sound foundation for an evolving set of universal standards that would serve to evaluate mobile computing in terms of both accessibility/usability global standards that would leave no one behind.

Mobile Usability heuristics and Accessibility Evaluation

No one left behind means just that. Anyone seeking to access information and experience should have the ability to do so. Global accessibility standards should not be viewed as an end-game or an unyielding set of rules not to be broken. Rather, global accessibility standards should be viewed as an ever expanding understanding that pays attention to user and context. Looking for a perfect solution or standards set is a fool’s game. The winner’s game would be to examine what is available – put that to use and continue to evolve the resulting guidelines over time and the changing mobile technology.

What happens next?

There is a pattern in the literature that highlights the partnership between the category of mobile application, the WCAG 2.0 guidelines, usability heuristics, and the ensuing and persistent errors that impact accessibility. These patterns can be associated with WCAG 2.0 mobile guidelines and usability heuristics (Bertini, 2009) in terms of the impact on accessibility. Of note is that there is clear guidance on conformance that if adhered to, will result in conformance to usability and accessibility. This conformance is the path to global accessibility and usability standards that seek to leave no one behind. The table below illustrates the concurrence of the WCAG 2.0 Mobile Accessibility Principles and universal usability heuristics. The development of new mobile hardware and applications would do well to incorporate both the principles and heuristics to ensure a globally accessible and positive experience for all users.

Table 1: Concurrence of Accessibility Mobile Standards and Usability Heuristics

WCAG 2.0 Mobile Accessibility	Usability Heuristics
Mobile accessibility considerations primarily related to Principle 1: Perceivable - Small Screen Size - Zoom/Magnification - Contrast	**Flexibility, efficiency of use, and personalization.** Mobile users should be able to tailor and personalize frequent actions, as well as to configure the system dynamically in accordance with contextual needs. Ability to set type font
Mobile accessibility considerations primarily related to Principle 2: Operable Keyboard Control for Touchscreen Devices - Touch Target Size and Spacing - Touchscreen Gestures - Device Manipulation Gestures - Placing buttons where they are easy to access	**Flexibility, efficiency of use, and personalization** **Good ergonomics and minimalist design**; audio options, keyboard simulations; thoughtful use of screen real estate, boundary mapping. Match the system and real world. Enable the user to interpret the information provided by presenting information in a logical, consistent and understandable order and format. Display graphical information in a way that is interpretable and accessible to the user.
Mobile accessibility considerations related primarily to Principle 3: Understandable - Changing Screen Orientation (Portrait/Landscape) - Consistent Layout - Positioning important page elements before the page scroll - Grouping operable elements that perform the same action - Provide clear indication that elements are actionable - Provide instructions for custom touchscreen and device manipulation gestures	**Consistency and mapping:** The user's conceptual model of the possible function/interaction with the mobile device or system should be consistent with the context. It is especially crucial that there is a consistent mapping between user actions interactions (on the device buttons and controls) and the corresponding real tasks (e.g., navigation in the real world). **Ease of input, screen readability and glanceability.** Mobile systems should provide easy ways to input data, possibly reducing or avoiding the need for the user to use both hands. Screen content should be easy to read and navigate through, notwithstanding different lighting conditions

WCAG 2.0 Mobile Accessibility	Usability Heuristics
Mobile accessibility considerations related primarily to Principle 4: Robust • Set the virtual keyboard to the type of data entry required • Provide easy methods for data entry • Support the characteristic properties of the platform	**Flexibility, efficiency of use, and personalization** **Consistency and mapping** **Ease of input, screen readability and glanceability.** Mobile systems should provide easy ways to input data, possibly reducing or avoiding the need for the user to use both hands. Screen content should be easy to read and navigate through, notwithstanding different lighting conditions **Realistic error management**. These shield mobile users from errors: when an error occurs, the system helps users to recognize, diagnose, and, if possible, recover from the error. Mobile computing error messages should be plain
	Esthetic, Privacy, and Social Conventions. Take esthetic and emotional/personal aspects of the mobile device into account – making sure the user's data kept safe and private. Mobile interaction should adhere to social convention.

Opportunity and Progress

One of the main challenges faced by current Information and Communication Technology (ICT) is making all kinds of information and services accessible and usable by all possible categories of users through mobile devices (mobile phones, smart phones, PDA, etc.). This challenge is not limited to the big companies that produce mobile devices, but engages all the stakeholders involved in the chain, from the content producer to the final user, such as network providers, service deliverers, software developers, and even search-engine giants.

To meet these challenges stakeholders and developers need more information about users and context of use such as: a) where location matters, and b) where location does not matter. The first category includes location-aware applications such as maps, navigational devices, walking directions. Here the device should 'sense' the environment and adapt the presentation of information accordingly. The second category is the opposite of the first and means that the location is ubiquitous – as in reading email, the news, sending text, and accessing the internet. (Users still prefer a desktop/laptop configuration for large document creation (Alshehri, 2010).) Here the idea would be to unshackle the user from time and place this implies an inherent dependence on data and network availability.

Understanding this simple categorization of use and device should simplify the manner in which usability and accessibility are joined to create viable and far-reaching solutions that are cost-effective for companies and a positive outcome for all users. The significance of this relationship and the associated benefits of attending to both – simultaneously should and will lead to positive user experience and steady progress towards a global set of inclusive, user-aware standards.

Conclusions and Moving Forward

This paper looked at the evaluation challenges related to the mobile context. Such challenges are essential to be dealt with in order to obtain a product that can be adequately accessible and usable, and thus considered to be universally accessible which aids and abets the development of global accessibility standards. Accessibility and usability are not the same and an application may be usable but not accessible and the reverse. Success in the mobile environment depends on the harmonious collusion between usability and accessibility. Finding the intersection of shared goals would further the development of applications that keep both perspectives in mind. In conclusion – the idea is to formulate standards that coalesce into a global framework of accessibility created in a way that attends to information that transforms gracefully to a mobile environment keeping in mind the user and context of use.

References

American Foundation for the Blind. Web (2011, June 20). Retrieved from “Cell Phone Accessibility Overview.” http://www.afb.org/Section.asp?SectionID=4.

Anthony, L., Kim, Y., and Findlater, L. (2013). Analyzing user-generated YouTube videos to understand touchscreen use by people with motor impairments. Proc. CHI ’13, ACM Press, 1223–1232.

Bertini, E., Catarci, T., Dix, A., Gabrielli, S., Kimani, S., Santucci, G. (2009). Appropriating heuristic evaluation for mobile computing. Int. J. Mobile Hum. Comput. Interact. 1(1), 20–41.

Estes, M. (2016, May). Time for a Response: Responsive Design, Accessibility, and University Websites. Retrieved from http://etd.auburn.edu/handle/10415/5214

Interactive Design Foundation. (2017, July 31). Mobile Usability Research – The Important Differences from the Desktop. Interaction Design Foundation. Retrieved from https://www.interaction-design.org/literature/article/mobile-usability-research-the-important-differences-from-the-desktop.

ISO 9241-20 (2006). Ergonomics of human-system interaction—Part 20: Accessibility guidelines for information/communication technology (ICT) equipment and services.

Kane, S.K., Jayant, C., Wobbrock, J.O., and Ladner, R.E. Freedom to Roam (2009). A Study of Mobile Device Adoption and Accessibility for People with Visual and Motor Disabilities. Proc. ASSETS’09, 115–122, ACM Press.

Naftali, Maia, & Findlater, Leah. (2014). Accessibility in Context: Understanding the Truly MobileExperience of Smartphone Users with Motor Impairments, Inclusive Design Lab HCIL College of Information Studies, University of Maryland, College Park, MD.

Nokia Accessibility. Nokia. Web (2011, June). “Making Mobile Devices Accessible For All.” Retrieved from http://www.nokiaaccessibility.com.

Pernici, B. (ed.) (2006). Mobile Information Systems: Infrastructure and Design for Adaptivity and Flexibility, p. 237. Springer, Berlin.

Petrie, H. and Bevan, N. (2009). The evaluation of accessibility, usability and user experience: In C. Stephanidis (Ed.) The Universal Access Handbook. London: Taylor & Francis.

Quiones, D., & Rusu, C. 2017, August). How to develop usability heuristics. Computer Standards & Interfaces archive Volume 53 (C), 89-122. Retrieved from http://dl.acm.org/citation.cfm?id=3085763

Lee, W. H., Lee, H.K, & Geraghty, L. (2017, Sept. 17). The usability attributes and evaluation measurements of mobile media AR (augmented reality). Retrieved from http://www.tandfonline.com/doi/full/10.1080/23311983.2016.1241171

Rocha, T., Carvalho, D., Bessa, M. et al. (2017). Univ Access Inf Soc 16: 581. Retrieved from https://doi.org/10.1007/s10209-016-0489-5

Shneiderman, B.: Promoting universal usability with multi-layer interface design.
In: Proceedings of the 2003 Conference on Universal Usability (CUU 2003).

Strategic Plan for Improving Management of Section 508 of the Rehabilitation Act (January 24, 2013) Retrieved from https://obamawhitehouse.archives.gov/sites/default/files/omb/procurement/memo/strategic-plan-508-compliance.pdf

Thatcher, J., Waddell, C.D., Henry, S.L., Swierenga, S., Urban, M.D., Burks, M., Regan, B., Bohman, P. (2003). Constructing Accessible Web Sites. Glasshaus, San Francisco.

Wegge, K.P., Zimmermann, D. (2007). Accessibility, usability, safety, ergonomics: concepts, models, and differences. Universal Access in HCI, Part I, HCII 2007, LNCS 4554, pp. 294–301, Springer- Verlag, Retrieved from https://link.springer.com/chapter/10.1007/978-3-540-73279-2_33?no-access=true

Web. 20 (2011, June). Web Content Accessibility and Mobile Web: Making a Web Site Accessible Both for People with Disabilities and for Mobile Devices. Retrieved from http://www.w3.org/WAI/mobile.

Yesilada, Y., Harper, S., Chen, T., and Trewin, S. (2010). Small-device users situationally impaired by input. Computers in Human Behavior 26, (3), 427–435.

Copyright notice

AED COP, How It Helps Government Agencies Consistently Test and Remediate Electronic Documents for Section 508 Conformance

Nicshan Floyd

Office of Accessible Systems and Technology
U.S Department of Homeland Security
301 7th street SW Washington DC USA
nicshan.floyd@hq.dhs.gov

Abstract

In October 2012, the federal Chief Information Officer’s Council (CIOC) Accessibility Community of Practice (ACOP) established the Accessible Electronic Document Community of Practice (AED COP) with the goal of improving accessible content, advancing the field of accessibility, and creating testing and authoring artifacts that are reusable across many agencies. The AED COP is comprised of subject matter experts from multiple federal agencies. This year the AED COP improved its partnership with industry leaders who develop software used to author electronic documents. By doing so, the groups are able to work together to make it more intuitive to create Section 508 conformant documents as well as enhance automated Accessibility Checkers that are based on the 508 refresh guidance. By improving authoring tools and by following a consistent streamlined testing process, it is the hope of the AED COP that every document author is able to quickly and easily create accessible content.

The Purpose of the AED COP

- Increase awareness of the importance of access to accessible electronic documents across the federal community.
- Promote successful strategies which increase the ability of federal employees to create accessible electronic documents.
- Advance the field of accessibility for all participating agencies by creating a repository of accessibility artifacts.
- Identify and improve the alignment for the definition of requirements for accessible electronic documents across federal government for all participating agencies.
- Promote successful strategies which create the highest level of accessibility for documents at the lowest cost.
- Identify and supply best practices to the CIO Council Accessibility Committee Best Practices Subcommittee.

AED COP Committee Representatives

- Department of Defense (DOD)
- Department of Education (ED)
- Department of Health and Human Services (HHS)
- Department of Homeland Security (DHS)
- Department of Justice (DOJ)
- Department of Labor (DOL)
- Department of State
- Department of Transportation (DOT)
- Department of Veterans Affairs (VA)
- Federal Reserve Board (FRB)
- Internal Revenue Service (IRS)
- National Aeronautical Space Administration (NASA)
- National Archives and Records Administration (NARA)
- National Institute of Health (NIH)
- Social Security Administration (SSA)
- US Access Board

AED COP Artifact Types

The AED COP has created 20 artifacts and two online training videos to serve as guidance for authoring, testing and remediating electronic documents for accessibility. This guidance is in alignment with the Revised Section 508 standards. The electronic document formats covered by the AED COP at this time are: MS Word, MS PowerPoint, MS Excel, MS Outlook and PDF. To access the artifacts and video trainings go to www.Section508.gov\Refresh-Toolkit\Test.

The Artifacts created can be divided into four categories:

- **Baseline Artifacts:** These documents contain a set of 22 systematic repeatable test criteria that cover Revised Section 508 standards and align with applicable WCAG 2.0 Level AA success criteria. Individual tests contain sufficient information and instruction to make a consistent and unambiguous measurement independent of other tests. Each test case provides the following information:
    - **Numbered Requirement**: How the component(s) should function in order to meet the related standards.
    - **Rationale:** An explanation of the elements/components the requirement is addressing (technical aspect in layman’s terms), effects on accessibility, consequences of incorrect implementation on accessibility (AT functionality), and the benefits of correct implementation.

- **Related Standards:** Applicable Section 508 standards and alignment with relevant WCAG 2.0 success criteria. *Note:* A 508 standard or WCAG criteria may be addressed by multiple tests.

Example of a Baseline Artifact

Inline Elements

Requirement [All Documents]	1. Meaningful text and objects must be placed inline.
Rationale [All Documents]	
…technical aspects	Text and objects can be formatted in documents to be 'inline' or 'floating' / 'wrapping'. Inline text and objects can be accessed by moving the keyboard cursor from element to element. Floating objects can be placed in front or, behind, or wrapping around the inline objects but they cannot be reached via the keyboard cursor.
…effects on accessibility	AT relies on the keyboard cursor to move through text and objects. Therefore, AT users cannot access floating objects.
…consequences	Floating content such as images overlapping inline text or tables that are surrounded on all sides by continuous text are not accessible via the keyboard cursor and therefore not accessible to AT users.
…benefits	Placing meaningful text and objects inline means all document content can be read and accessed by those who rely on navigation via the keyboard cursor.
…rationale summary	Summary: Text and objects can be formatted as inline or floating/wrapping. Floating text and objects are not accessible via the keyboard cursor and therefore not accessible to AT users.
Related Standards [All Documents]	508 1194.21 SW (a): Keyboard Accessibility 508 1194.31 FPC (a): Use Without Vision 508 1194.31 FPC (b): Use With Low Vision 508 1194.31 FPC (f): Use With Physical Limitations WCAG2 1.3.1: Info and Relationships WCAG2 2.1.1: Keyboard
Tools Necessary [Word 2013]	Physical System Keyboard, Draft view, Accessibility Checker

Test Instruction 1a: Manual Find of Applicable Components [Word 2013]	a. Set the document view to 'Print Layout' (View Tab > Document Views > Print Layout). Examine the document for meaningful text and objects. Objects include: • Meaningful images/pictures (including images of text and images in tables) • Shapes (Call out boxes) • SmartArt • Chart (Diagrams) • Tables • Text boxes • Icons with hyperlink • Other objects Note: • In Word 2013, text is always placed inline.
Test Instruction 1b: Accessibility Checker Find of Applicable Components [Word 2013]	a. Run the Accessibility Checker and look for "Object Not Inline" errors.
Test Instruction 2a: Manual check for Inspecting/Using Components [Word 2013]	a. Set the document view to 'Draft' (View Tab > Document Views > Draft). Compare objects that show in Print Layout view to objects that do not show in Draft View. Objects that do not show in Draft view may not be inline. Note: • Images and text boxes that are inline may show a placeholder in Draft View, even though the actual content may not display. • Decorative objects do not need to be inline. • Running Header and Running Footer content (including page numbers) do not need to be inline - see test for Running Headers & Footers, #12.
Test Instruction 2b: Accessibility Checker for Inspecting/Using Components [Word 2013]	a. Errors are listed under "Object Not Inline". Note: • Decorative objects do not need to be inline. • Running Header and Running Footer content (including page numbers) do not need to be inline - see test for Running Headers & Footers, #12.
Test Instruction 3a: Section 508 Failure Conditions [Word 2013]	• Meaningful text or objects are not inline. ○ Fails 1194.21(a): Keyboard Accessibility. ○ Fails 1194.31(a): Use Without Vision ○ Fails 1194.31 (b): Use With Low Vision

Test Instruction 3b: WCAG2 Failure Conditions [Word 2013]	- Meaningful text and objects are not inline. ○ Fails 2.1.1: Keyboard
Test Instruction 3c: Baseline Requirement Test Results [Word 2013]	- Any failure in 3a. ○ Fails Baseline Requirement #1 - Meaningful text and objects are inline). ○ Passes Baseline Requirement #1
Advisory: Tips to enhance or streamline test processes [Word 2013]	- For those creating a streamlined process from this Baseline, review the guidance in “Developing a Streamlined” starting on page 11. - Grouping multiple associated objects is a best practice.

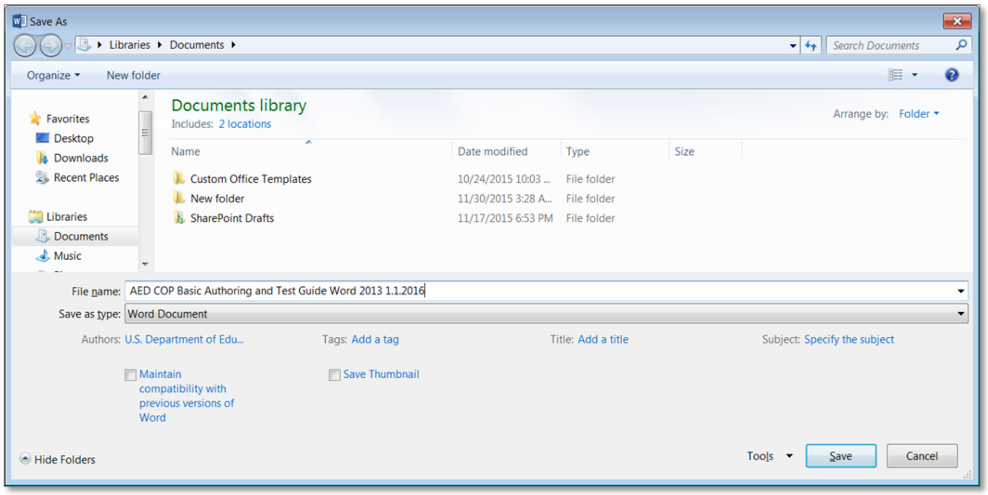
- **Authoring Artifacts:** Authoring artifacts provide guidance on how to use the native application’s built-in tools to author accessible content that is in accordance with the Revised Section 508 standards. These documents do not contain best practice guidance. It is recommended that Departments and Agencies add any additional guidance that meets their unique environment needs.

Example of Authoring Artifact

Save as a Word document (.DOCX) with a descriptive filename

A descriptive filename that identifies the document or its purpose helps everyone (including people with disabilities) locate, open, and switch between documents. In addition, the document must be in a “.docx” format; the authoring and testing instructions in each section are only for MS Word 2013 documents.

Author Accessibly
Go to File>Save As



File name: AED COP Basic Authoring and Test Guide Word 2013 1.1.2016

Save as type: Word Document

Authors: U.S. Department of Edu...

Tags: Add a tag

Title: Add a title

Subject: Specify the subject

□ Maintain compatibility with previous versions of Word

□ Save Thumbnail

1. Save as a Word Document (.docx).
2. Save your document with a descriptive filename.

Check Your Work

Look at the file in Windows Explorer **OR** the title bar in MS Word.

1. Check that the file type is a Word 2010 document (.docx).
2. Check that the filename is descriptive and identifies the document or its purpose.

For example, Document1.docx is not a descriptive name; however, OMBReport387_2102014_v2.0.docx is an example of a descriptive filename.

- **Detailed Checklist Artifacts:** Detailed checklists provides guidance on how to test and how to author documents for accessibility. Users can expand a checkpoint and choose to receive information on either how to test or how to author. All checkpoints contained in these artifacts correspond with the test cases found in the Baseline artifacts.

Example of Detailed Checklist

Is the file name descriptive, is the file in the .docx format, and is the file NOT protected?

How to test

Instruction 1: Look at the filename in Windows Explorer **OR** the title bar in MS Word. An example of a non-descriptive file name is “Document1” An example of a descriptive filename is “FY16-Report” The file must be in the “*.docx” format for accessibility testing to be possible.

NOTE: If the document extension is not displayed, open your documents folder in Windows Explorer, select “Tools>Folder Options>View>uncheck ‘Hide extensions for known file types’>OK.”

> **Test A:** Is the filename descriptive and does it identify the document or its purpose? If not, the document fails this test.
> **Test B:** Is the file in “Word Document (.docx)” format? If not, the document fails this test.

How to author for accessibility

A descriptive filename that identifies the document or its purpose helps everyone (including people with disabilities) locate, open, and switch between documents. In addition, the document must be in a “.docx” format because these authoring and testing instructions will only work if the file is in the “.docx” file format. Document restrictions limit or prevent users of assistive technology from reading or editing the document. If you must use document restrictions, turn them off during testing and then ensure assistive technology users have access to the password.

Select “File tab>Save As”

- Save as type: “Word Document (*.docx)”.
- Save the document with a descriptive filename.

- **Printable Checklist Artifacts:** These documents function as reporting templates to capture test results related to Baseline test cases. Each document element that must be tested is accompanied with a list of questions. A tester has the ability to respond Yes, NA or No to each question. Additionally, the tester has the ability to add comments at the end of the test report. Departments and Agencies are allowed to add any additional test criteria that meets their unique environment needs.

Example of Printable Checklist

The Word document (.docx) is saved with a descriptive filename.

The file type is Word document (.docx) format □ **Yes** □ **No**
The filename identifies the document or its purpose □ **Yes** □ **No**

How Industry Leaders are Supporting the AED COP

This year the AED COP has strengthened its partnerships with industry leaders such as Microsoft, Adobe and NetCentric with the goal of improving the intuitiveness of authoring, remediation and automating accessibility testing tools to make it easier to generate accessible content even if a person is not familiar with accessibility requirements. Some of the work that has already been completed includes:

- Microsoft has worked with the AED COP to generate accessibility authoring guides for Office 365 products which includes Office for Android and IOS. Microsoft has also enhanced it automated accessibility checker to better align with the Revised Section 508 standards. Additionally, Microsoft is using Artificial Intelligence (AI) to help individuals create accessible content and automatically add alternative text to images. Future work is being done to walk individuals through the process of generating accessible content.
- Adobe has added several features to Acrobat DC to make it easier to remediate PDFs. Some of the enhancements include the ability to automatically delete all empty tags, an endless undo feature added to the Tags pane and the ability to export PDF content to MS Word has been improved.
- Net Centric has added the WCAG 2.0 criteria to their accessibility checker.

Challenges We Face and How to Resolve Them

With the release of the Revised Section 508 standards and ever evolving technologies, individuals find it increasingly difficult to understand what is required to make electronic documents accessible and how to make them conformant with the Revised Section 508 standards. Facing these problems head-on, the AED COP will continue to work with industry leaders to improve their products so that individuals are guided through the process of authoring and testing electronic document content for compliance.

Additionally, the AED COP will continually update existing artifacts to align with improvements made to technologies as well as add new guidance to support other file formats such as LiveCycle and Publisher. By having authoring tools include accessible templates and more detailed instructions for making documents accessible, individuals will have a better understanding and will be more capable of creating accessibility content. For those that will need more assistance, individuals can take advantage of the AED COP's ever evolving library of free training videos, authoring and testing guides.

Copyright notice

Making an Airline Site Accessible

Gian Wild

AccessibilityOz
Melbourne, Victoria, Australia
gian@accessibilityoz.com

Abstract

In November 2013 the Department of Transport released a Rule regarding Nondiscrimination on the Basis of Disability in Air Travel: Accessibility of Web Sites and Automated Kiosks at US Airport. This requires that all: “U.S. and foreign air carriers that operate at least one aircraft having a seating capacity of more than 60 passengers, and own or control a primary Web site that markets air transportation to consumers in the United States … must make their web pages providing core travel information and services accessible to persons with disabilities.” This requirement comes into effect on December 12, 2015. The DOT Rule requires compliance with WCAG2, Level AA and: User testing be conducted with people with disabilities (vision, auditory, tactile and cognitive disabilities); and A specific form be included as part of the booking engine for users to require assistance at the airport.

Scope of DOT requirements

We found that the airlines who contacted us believed that they only needed to make booking a flight accessible, however the requirements are much broader than that:

(i) Web pages associated with obtaining the following core air travel services and information that are offered on your primary Web site are conformant by December 12, 2015:

a. Booking or changing a reservation, including all flight amenities;

b. Checking in for a flight;

c. Accessing a personal travel itinerary;

d. Accessing the status of a flight;

e. Accessing a personal frequent flyer account;

f. Accessing flight schedules; and

g. Accessing carrier contact information.

(ii) All remaining Web pages on your primary Web site are conformant by December 12, 2016.

Nondiscrimination on the Basis of Disability in Air Travel:
Accessibility of Web Sites and Automated Kiosks at US Airports
(http://www.regulations.gov/#!documentDetail;D=DOT-OST-2011-0177-0111)

We also found that the airlines we worked with had not realised that **all** their web sites needed to be made accessible by December 12, 2016. One of the airlines we worked with (and are still working with!) has over fifty web sites; so making them all accessible will take time!

In our work we found a number of accessibility issues that were unique to airlines and often overlooked.

Use of color in the booking engine

The inaccessible use of color can severely affect a user's ability to understand information, or to interact with features such as a seat selector or route map. A large percentage of the population is color-blind, so the accessible use of color should be considered a high priority issue.

A booking engine is the feature that customers use to book a flight online. It is a complex process and includes interactive features such as the ability to review and choose a desired flight or select a specific seat on the aircraft, perhaps with particular requirements such as additional legroom.

The booking process also requires that important information such as terms and conditions and baggage policy information is clearly available and accepted by the person making a booking prior to payment.

Use of color

WCAG2 Level A Success Criterion 1.4.1 (Use of Color) says:

> "The intent of this Success Criterion is to ensure that all users can access information that is conveyed by color differences, that is, by the use of color where each color has a meaning assigned to it. If the information is conveyed through color differences in an image (or other non-text format), the color may not be seen by users with color deficiencies. In this case, providing the information conveyed with color through another visual means ensures users who cannot see color can still perceive the information.
>
> Color is an important asset in design of Web content, enhancing its aesthetic appeal, its usability, and its accessibility. However, some users have difficulty perceiving color. People with partial sight often experience limited color vision, and many older users do not see color well. In addition, people using text-only, limited-color or monochrome displays and browsers will be unable to access information that is presented only in color."

Understanding Success Criterion 1.4.1 (http://www.w3.org/TR/UNDERSTANDING-WCAG20/visual-audio-contrast-without-color.html)

Complex interactive features often make use of color to highlight items such as flight paths on an airline route map, the availability of a certain flight or type of fare, or seat types in a seat selector. People who are color-blind may not be able to differentiate between items (e.g. an available or unavailable seat) where the only difference between them is the color.

On many systems we also found that links within the content are not underlined by default and differ from standard text by color alone. Being able to identify and follow links to important information such as terms and conditions, help information, baggage policy etc. is essential to any user (not just people with disabilities) prior to finalising a flight booking. It is vital that this information is clearly presented and can be accessed by all users.

The color issue is also compounded if items relying on color alone also do not meet color contrast requirements.

Color contrast

WCAG2 Level AA Success Criterion 1.4.3 (Contrast – Minimum) says:

> “The intent of this Success Criterion is to provide enough contrast between text and its background so that it can be read by people with moderately low vision (who do not use contrast-enhancing assistive technology). For people without color deficiencies, hue and saturation have minimal or no effect on legibility as assessed by reading performance (Knoblauch et al., 1991). Color deficiencies can affect luminance contrast somewhat. Therefore, in the recommendation, the contrast is calculated in such a way that color is not a key factor so that people who have a color vision deficit will also have adequate contrast between the text and the background.”
>
> Understanding Success Criterion 1.4.3 (http://www.w3.org/TR/UNDERSTANDING-WCAG20/visual-audio-contrast-contrast.html)

We encountered the use of low contrast shading (or greying out) in most airline systems reviewed. This was evident within the booking system when highlighting flight unavailability, within date pickers and when showing the navigation stages through the booking process.

All information presented as text in a data table, link, promotional images, and form validation error messages for example should comply with color contrast requirements to ensure visual readability by the greatest number of users.

User testing

We often found that these requirements were an afterthought for the airlines trying to meet DOT compliance. One airline argued that it would be sufficient to test the booking process with one staff member with a disability only.

Special Assistance form

Two of the airlines we worked with already allowed users to specify that they needed assistance due to vision or physical impairments, however the DOT Rule requires that airlines provide a way for users to request:

- Wheelchair assistance;
- Seating accommodations;
- Escort assistance for vision impaired customers; and
- Stowage of assistive devices, such as wheelchairs.

Although these examples were the only ones provided in the DOT rule, this form should also provide ways for users to request or specify:

- An accompanying service animal;
- Pre-boarding requirements;
- The presence of devices such as pins, pacemakers and other implants;
- Additional carry-on baggage limits for the transportation of medical equipment; and
- Reduced airfare costs for attendant carers.

In conclusion

The DOT rule is much more than just “WCAG2 compliance” and unfortunately this was often not clear to the airlines affected.

Copyright notice

The Unity Engine as An Extensible Platform: Findings from the System for Wearable Audio Navigation

Phillip Isaac Roberts

Georgia Institute of Technology / AMAC Accessibility
Atlanta, Georgia, USA
phil@proberts.us

Keenan May

Georgia Institute of Technology
Atlanta, Georgia, USA
kmay@gatech.edu

Bruce Walker

Georgia Institute of Technology
bruce.walker@psych.gatech.edu

Jeff Wilson

Georgia Institute of Technology
jeff@imtc.gatech.edu

Abstract

The research, prototyping, and development of new technologies and devices are faster and ever more sophisticated. However, the creation of accessibility tools for new formats lag, and what those needs are for emerging technologies must be researched. Researchers at the Sonificiation Lab at Georgia Tech have used a variety of tools in research and development for the System for Wearable Audio Navigation (SWAN) project. Concurrent development has been done on a wearable prototype and a virtual reality (VR) simulation of real-world scenarios using the Unity Engine and consumer VR equipment. Virtual Reality and especially Augment Reality(AR) tools are already being utilized in research and consumer settings, and it is unlikely that current forms will benefit the visually impaired community. There is a need to quickly and easily share extensible accessibility tools for research and development on emerging formats.

Background

The System for Wearable Audio Navigation, or SWAN, is an ongoing research project under Bruce Walker et al. in the Sonification Lab at Georgia Tech. The project has been a continuous effort to research systems and interfaces to assist a variety users who are visually impaired, either situationally or conditionally. Through iterations of the project, a wearable prototype and a virtual reality simulation have been concurrently developed. The VR simulation is used to rapidly design and administer tests and scenarios that may otherwise be time consuming, difficult, or potentially hazardous in a real-world scenario. The current phase of development, SWAN 2.0, leverages new tools and technologies for the wearable and VR prototypes.

SWAN has used a variety of participants, both with visual disabilities and without. During formative research, a lot of data can be collected on the initial design flaws of a prototype, and general research on audio cognition, which we can use our general research population. Our tools allow us to simulate various visual conditions, such as a firefighter scenario navigating through a smoky room. Another scenario we have tested is finding a pill bottle where we simulate low vision by blurring the participant’s vison beyond a short field of view, requiring participants to physically move their face close to correctly find objects. While using filters to approximate impairment cannot substitute for including participants with those impairments, what these filters do well is allow us the opportunity to quickly test ideas and concepts on our readily available general research pool (i.e., Georgia Tech students), before spending the time and effort of recruiting participants with specific impairments.

We have involved users with disabilities at various points in our research. This has been especially insightful for the design of the audio interface and other insights. The Sonification Lab conducted an extensive studies on bone-conducting headphones (May & Walker. 2017) because of the need to have unobstructed ears in a real-world environment.

There are a few reasons for involving participants with disabilities and without. If we consider bringing this technology to market, it we should consider many users and purchasers of the technology. Economies of scale are also a factor when bringing to market a product, and a product is used by more people will in general be easier to purchase for all. Audio displays are used in a variety of settings, and more advanced tools for spatial awareness and navigation could potentially find a variety of uses. To this end, we must understand both the common and unique needs for the visually impaired and the general population and how to design for different user groups.

The Unity Engine

The Unity Engine, which was created as videogame engine, has been adapted as a research tool in many of the labs at Georgia Tech and elsewhere. This is for a few reasons: Unity uses high-level programming languages which in general are faster for prototyping ideas, there is a wealth of tutorials and instructional materials, the interface is easily modified and extended, and the Unity Asset Store has several useful materials and extensions that are relatively inexpensive or even free. Related to Virtual Reality, Augmented, and other forms of “Mixed Reality”, Unity has provided several resources and tools to aid in the development of these technologies. Many new commercial Alternative Reality products support software created with the Unity Engine. Overall, these factors have created an open environment and community around the Unity Engine that allows members to easily share and modify resources. This can be especially empowering for groups and individuals without the resources to design or code these tools or materials themselves. For researchers, this is helpful as a lot of design or coding work can be adapted or modified from existing materials, and more time can be dedicated to actual research. That is not to say everything is provided, but there is a considerable reduced development burden.

Extensible Tools

What do we mean by an "extensible tool" for accessibility? This is an idea that has emerged from our work with SWAN and the Unity Engine. As mentioned, Unity has heavily integrated into their software engine a marketplace for user content, called the Unity Asset Store. This is a resource where a variety of code, tools, assets, and models can be obtained for reasonable prices or oftentimes free. This is similar to an API or library, however the integration is usually surface-level or somewhat patchwork. An extensible tool does not provide much on its own, but it can be programmed for additional functionality or combined with other tools to make a more functional prototype or resource.

The Unity Engine as an Extensible Tool for Accessibility

What is lacking on the Unity market and similar platforms are accessibility tools. Currently there are two on the Unity Asset Store that we could find, this is not an endorsement of any product, these are provided as examples: the UI Accessibility Plugin (UAP), and FBInput Handler. UAP is oriented to make mobile games screen reader accessible, and FBInput Handler allows input remapping, which is especially important for physical accessibility.

One example comes from Georgia Tech's Interactive Media Technology Center (IMTC). IMTC has created a suite of extensible accessibility tools for the Unity Engine. These modify the GUI to simulate conditions of color blindness and vision loss and are easily integrated into existing Unity projects. This has been used for several research initiatives, and is available for Georgia Tech personnel in the IMTC Github repository. Within the gaming industry, Electronic Arts has created a suite of accessibility toolkits for use across their franchises, which has been very effective in improving the accessibility of their games.

For the SWAN VR prototype, we used the Unity Engine, a variety of existing tools from the Unity Asset Store, and the Oculus Rift and Steam VR toolkits. The scenes for scenarios are assembled from free models and objects from the Unity Asset Store, the HMD unit uses the Steam toolkit, and Head-Related Transfer Functions (HRTFs) and audio spatialization are using the Oculus Audio SDK. Visual filters are modified from other graphics applications. Our research with the VR prototype has focused on creating an accurate audio interface that responds to the simulated environment. We have found that testing through different settings and scenarios reveals different weaknesses in an interface. Our scenario of moving through a small room showed the limits of perceiving the accuracy of the distance and location of nearby audio cues, and the scenario of a busy city block showed the need to be able to be correctly alerted to hazards and navigate safely. Each of these scenarios could be designed using a relatively small amount of researcher work hours, and minor details modified in mere minutes. Testing can be performed at any time in a controlled environment and free of environmental hazards and conditions. In the real world, an intersection, nearby construction area, or subway station can pose many potentially life-threatening situations. Failure in such a scenario is not an option, thus the need to extensively test in a safe environment before even approaching a full scenario.

Another example how VR is being applied in research comes from IMTC. In cooperation with the Center for Disease Control, they have prototyped a simulation for training proper protocols for the care of Ebola patients, egress, and maintaining sanitation. As alternate reality devices are more reality available, so too are their possible applications, and the need for accessibility considerations.

Bricolage and the Maker Community

Bricolage is a French loan-word that has its origins in the DYI (Do-It-Yourself) and maker communities. Many fields have adapted its meaning for their own use but for this paper we will be referring to it in more of its original sense: That is, bricolage is a process of using readily available materials and hacking and tinkering with those materials in some way to explore ideas and projects. In recent years, there have been a variety of DIY products released on the consumer market that allow more refined prototyping. Additionally, there is a wealth of instructional materials and example projects online and within local maker communities. The combination of these resources allows individuals and groups—who may be lacking the resources or expertise of larger organizations—to create fairly high-quality projects. While many members of the maker community are amateurs by definition, their work and participation can be meaningful and empowering or even a means of activism (Busch, 2015). Additionally there is a great educational value in involving and educating people in DIY projects (Kuznetsov, et al. 2012).

The accessibility community may benefit from the philosophy of bricolage and engaging maker communities. We may explore ways to create low-cost of entry accessibility tools by hacking and modifying existing technologies. In a more practical sense, there are many underserved individuals who could benefit from this work. Many in the accessibility community could be considered makers themselves. There are many communities dedicated to sharing tools, hacking, and tinkering with existing technologies. Not to mention trouble-shooting software after every update. For many, adapting tools and hacking technology is necessity instead of a hobby.

Perhaps we can learn and apply these concepts into more professional settings. Tools for Life at Georgia Tech uses many methods similar to bricolage in their work. They interview and research and individual’s accessibility needs, gather tools and resources, and modify and fit them as necessary to best serve the needs of an individual. Tools and technologies may be chosen not necessarily for a pre-determined purpose, but rather for their extensibility so that they may be customized for a purpose at hand. Anecdotally, the ubiquitous hackable microcomputers, sensors, and components have been a great asset and helps make the level of research and prototyping we do possible.

Coming Alternate Reality Accessibility Barriers

We will leave others to speculate the market success of various products, virtual and mixed realities, However, certain applications of augmented or mixed-reality (MR) are currently being research applied in research or professional settings, and several applications look promising. Various form of heads-mounted display (HMD) systems or visors have been used in research settings, especially medicine, manufacturing, product prototyping, inventory and shipping, and military applications. Mixed Reality applications pose serious questions for ICT professionals. There are many potential accessibility failure points for HMDs: depth of field and focal issues, color contrast and luminosity, mobility related to head tracking, text and image size, etc. This is especially important because many applications are so deeply embedded in an optimized process.

Imagine the example of a nurse using a HMD that navigates the nurse through a hospital, as they carry the correct materials and medication to specific patients. If the nurse suffers from a mobility impairment in the neck or spine, they may be unable to fixate on a target, or vital information may not be displayed. Any visual impairment may confound basic tasks, as they struggle to perceive the display. There may be additional affects job performance, which would be extremely detrimental for the nurse.

The focus on the Unity Engine, Virtual Reality, and other forms of Augmented Reality in this document are for several reasons: these technologies and platforms are relatively new for the general population, there will be a need to address the accessibility of these products, and tools are readily available to developers to share, improve, and work-on. We need to ask ourselves many questions, “How should an audio interface compensate for visual information?”, “How do we notify a user their head is fixated on a correct target?”, “What levels of contrast are necessary within the limitations of HMD displays?”, “If a user cannot position themselves, what do we display and how do we interface?”, “How does a HMD fit within the overall accessibility of a task or process?” These areas are new, and will require thought and effort. Companies which are developing these products are generally struggling enough to find viable applications that may profit from. With the focus on “How can this product be viable?” there is the danger that accessibility is not considered, or the resources for accessibility development simply are not available.

Extensible Accessibility Tools

Returning to our subject, what then is an extensible accessibility tool? We have examples elsewhere of more general purpose toolkits and APIs, maker tools and materials, software marketplaces, and the A11y initiatives for web accessibility. In general, extensible tools are light-weight, easily implemented and modified, and are shared in an open manner where they can be examined and improved by a community. They may or may not be free, but a large majority (or certainly popular) extensible tools are free to use or under a general use license. They may not have the features of a fully developed API, but that is not necessarily ideal. Because one may not understand the extent of a problem that is being solved, a tool that can be adapted to more situations can be more beneficial than a tool that serves a singular purpose well, but cannot be easily modified or integrated.

Extensible tools have communities. We should keep in mind the varies forums, repositories, and communities dedicated to sharing and improving extensible tools. These serve not only as the locations where ideas and tools are shared and improved, but also as an opportunity for education and advocacy.

What accessibility developers and advocates can do is continue and expand initiatives to encompass emerging technologies. We should seek opportunities to share, develop, and educate. We should continue to develop our own channels and resources, but also engage other communities, especially developers for these technologies. The Unity Asset Store is only one of many resources where we could provide tools and information. Developers big and small often welcome the opportunity to engage more potential customers, especially when the means to do so are well-designed and quick to implement. DIY communities too are particularly passionate and open to improving and developing practical solutions. Deviating to a perspective of social empowerment, the accessibility community has much to gain by continuing to engage and involve themselves in other communities—improving awareness, acceptance, and important allies in our efforts to improve accessibility for all.

References

Stacey Kuznetsov, Alex S. Taylor, Tim Regan, Nicolas Villar, Eric Paulos. (2012). At the Seams: DIYbio and Opportunities for HCI ACM conference on Designing Interactive Systems 2012 (DIS 2012), Newcastle, UK June 11-15, 2012.

Otto von Busch. (2014). Hacktivism As Design Research Method. The Routledge Companion to Design Research (2014).

May, K. R. & Walker, B. N. (2017). The effects of distractor sounds presented through bone conduction headphones on the localization of critical environmental sounds. *Applied Ergonomics* 61 (2017), pp. 144-158. http://dx.doi.org/10.1016/j.apergo.2017.01.009 Published online: Feb 2017.

Dingler, T., Lindsay, J., & Walker, B. N. (2008). Learnability of sound cues for environmental features: Auditory icons, earcons, spearcons, and speech. Proceedings of the International Conference on Auditory Display (ICAD 2008), Paris, France (24-27 June).

Wilson, J., Walker, B. N., Lindsay, J., Cambias, C., & Dellaert, F. (2007). SWAN: System for Wearable Audio Navigation. Proceedings of the 11th International Symposium on Wearable Computers (ISWC 2007), Boston, MA (11-13 October).

Copyright notice

Creating Professional Accessibility Reports

Tim Offenstein

Campus Accessibility Liaison
University of Illinois at Urbana-Champaign
1304 W. Springfield Ave.
Urbana, IL 61801
United States of America
timo@illinois.edu

Abstract

The concept of accessibility evaluation as a distinct profession is growing. With the increase in liability, litigation and a desire to be inclusive of all audiences, public and private institutions have a growing need for understanding how IT products meet accessibility requirements. This paper prescribes an approach for creating usable accessibility reports that engage the client and provide a teachable environment. The focus is upon building a relationship with the client through clear, concise communication, promoting the idea that the accessibility expert should be considered a part of the team, and working towards a common goal.

Introduction

Accessibility is often viewed as a complex, difficult, and negative topic, one best avoided until the end of the project life cycle. Unfortunately, that perspective only makes the situation worse and increases the challenge accessibility professionals face as, commonly viewed, "bearers of bad news." The typical setting is one where the product release is looming and the team is tired and ready to cap off the project and celebrate; hardly the kind of atmosphere conducive to introducing more work and fixes.

Given these matters are often beyond our control, accessibility professionals are placed in an awkward position even before we enter the room. Therefore, it’s important to utilize the tools we have to communicate a positive message, especially when the situation feels negative.

This paper will look at two factors for accomplishing a positive and clear approach. First, we will briefly discuss the *mindset* necessary, and then we’ll deal with the *report* aspect of this process.

Our Mindset

You’ve accepted the job of conducting an accessibility evaluation of a web application that has already been purchased and is in use at your school. The product was first used two semesters ago by enthusiastic faculty who loved the easy-to-use tool for their classroom. At the time, no one considered whether accessibility was a factor. Now, however, a faculty member has read an article about a fellow institution getting sued for using a similar type of application.

First, it irritates you that no one considered asking, prior to purchasing, if disabled audiences would enjoy an equivalent experience with this software. And secondly, it’s awkward evaluating a product that you feel they’re going to use regardless of what you find. So, now you’re set to rip into the evaluation and tear the tool apart. You feel a sense of justification with each issue you find, especially the showstoppers. Unintentionally, you’ve set yourself up to fail your role as an accessibility expert.

The first tool to use for a successful evaluation is foundational; it is your mindset. Remember that you’re entering a potentially difficult scenario, particularly if the product has a poor degree of accessibility. Take a deep breath, a step back and consider the following:

1. **Do not take it personally.** If you allow your emotions to get involved, as the above paragraphs suggest, then you’ve lost the objectivity necessary to successfully guide the client to an accessible product.
2. **Use solution oriented language.** Be conscious of how you’re framing the issue so the client understands there are best practices for solving their issue.
3. **No problem should be treated as insurmountable; almost every issue is solvable.** Your goal is to get them on board with solving the roadblock. That’s hard to do if you approach it as a doomsday scenario. So be positive, but not naively so.
4. **Present yourself as the expert on accessibility because that’s what you are.** You have been brought into the project because you’re the one who knows accessibility; therefore, speak with confidence.
5. **You are not the problem; you are a part of the solution.** Your role is to help the client identify any accessibility issues. After identifying an issue, you work to help them understand and address the issue as best you can. Identification is your primary role. And remember, you didn’t create the issue, but you can help solve it.

An important aspect of having a successful mindset is understanding that you are creating a *relationship* with the client and their developers. See yourself as joining their team as an important proponent of building, not only an accessible tool, but a comprehensive product that will benefit all audiences. Helping them see a broader perspective enhances your contribution as a part of the team. Accessibility is not something most people, without disabilities, think about unless prompted; however, with this approach, accessibility is no longer seen in a negative light but rather as an opportunity for innovation that can increase the marketability of their product.

Framing the right mindset will go a long way toward preparing a positive, solution-oriented report and proactively getting your client aligned with the same goals you have.

Our Report

The accessibility evaluation report is your tool for educating, enlightening and helping the client to see the value in producing an accessible product. As such, the clarity with which the report is composed becomes immensely important. It should *not* be viewed as a compliance checklist noting whether a regulation has been met. Rather, the accessibility report is an opportunity to do the following:

- Identify and communicate about the issues: The report gives you an opportunity to explain issues that the developer, most likely, has never considered. For some, this will be a new experience – make the most of that.
- Teach the client about the basics of accessibility: If the developer learns WHY alternative text is important, rather than just HOW to add `alt="…"` to the code, they're better equipped to build successful sites in the future.
- Educate the client in terms of "seeing" their product through the experience of an audience they may not have considered: You can expose them to how screen reading software announces their page or to using tab key navigation instead of a mouse.

By approaching your evaluation report as an opportunity to teach, the client will be better equipped to build accessible tools in the future. You are empowering the client so they'll be less likely to repeat the same mistakes.

As mentioned above, often times your report may be the first exposure to accessibility practices that some clients will have. You're probably dealing with stereotypical thinking such as the idea that accessible web pages are dull and ugly or accessibility inhibits creativity and stymies innovation. Granted, there may be real obstacles to overcome, like bells and whistles that the client feels they must have; but by clearly outlining the issues involved, you help the client make the right decisions. Additionally, by drafting a clear, easy-to-understand report, you avoid contributing to the stereotypical thinking that accessibility is dry, boring and overwhelming.

Structuring the Report

If your report has a clean, clear structure to it, it is *consumable*. It's easier to comprehend and understand. The specific layout will differ depending upon your personal tastes and context, but here is a basic outline of the necessary components for a consumable report:

1. **Header/banner Information** – Your company name, evaluator, report number, date, contact e-mail. You may decide to place this information somewhere other than the header, but it needs to be readily available.
2. **Title** – Include the client name so it's easy to identify what's being evaluated.
3. **Section 1** – Project name and the URL of the site being evaluated. Do not assume everyone knows the exact address. You don't want them thinking you're looking at site X, when you're evaluating site Y. The URL spells that out.
4. **Section 2 – Scope –** This important section is where you spell out caveats that often go unannounced. You're setting the stage in section 2 so that assumptions are clarified and expectations are understood.
    a. **Ranking system** – This should be upfront and clear. The issues must be ranked according to their severity so that the developer can set priorities for what needs to be fixed first. This triage is also a way of educating the client.

    b. **Testing environment** – List the platform, operating system, software/version numbers and the assistive technology being used to test their site. It is important that the client be able to replicate your findings.

c. **Disclaimer** - Make it abundantly clear that no evaluation is exhaustive; we’re not perfect and something may be missed. Further, explain that an accessibility evaluation is an iterative process that needs to be conducted more than once. Often when one issue is fixed, another pops up; so, you want to spell that out at the start.

5. **Section 3 – Summary –** Although writing a summary is the last step in the process, it needs to be positioned upfront for the busy executive who needs to be informed of the findings but who does not necessarily need the detailed results. The summary needs to be frank, succinct and compassionate. The product may be a mess, but you don’t want to alienate the client by being disparaging. Diplomacy is called for in writing the summary. Remember, your goal is to get them onboard with making the product a success.
6. **Section 4 – Code Review –** Web sites need to have clean, well-formed code that complies with the Document Type Declaration (DTD). Often this step is overlooked, but a poorly written web page can greatly impact how screen reading software interacts with it.
7. **Section 5 – Automated Tool Evaluation –** Be sure to reference the tool and how to locate it in case the client wants to use it. Most legal settlements call for an automated tool evaluation, so this is an important part of the report. However, note that automated testing typically covers only 40 to 60 percent of the issues and cannot evaluate some of the most impactful showstoppers, such as lack of keyboard support.
8. **Section 6 – Manual Check –** Use of a basic accessibility checklist will help you ensure there are no significant gaps in your testing (see referenced checklists at the end of this paper). Manual checks often uncover showstoppers that automated checks miss, such as keyboard accessibility or the efficacy of link text. This section is critical and quite often becomes the largest part of the evaluation report.
9. **Section 7 – Mobile and Tablet Evaluation –** This is an area that needs a separate section, apart from desktop evaluation. Be sure to verify with the client that they want their product tested in this regard, because not all web applications are meant for the mobile environment.
10. **Section 8 – PDF Review –** Adobe PDFs are everywhere, particularly on government sites. It is usually necessary that a brief review of them be included in a comprehensive report. Luckily, Adobe provides accessibility checking functions with recent versions of Acrobat, so it is easy to run their checker and then report on that.
11. **Section 9 – General Usability –** It’s common, when doing an accessibility evaluation, to come across areas of the site where the usability can be improved. It’s helpful to your client for you to mention these items even though they don’t tie directly to accessibility requirements. That being said, usability issues are 3 to 5 times harder for disabled audiences to solve than others. Tweaking the usability of a site from an accessibility perspective can benefit everyone.

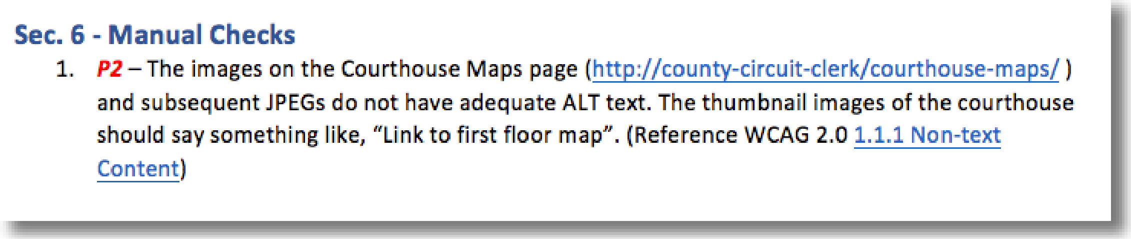
Notice that the above outline is structured with “**Numbered Sections**.” *Every* finding or item in a report *must* be numbered. This cannot be overstated because it means that each issue is categorized, findable and easily referenced. This is particularly important in follow-up reports and cross-referencing because it avoids confusion about which issue is being discussed.

Use Case Scenarios

The approach discussed so far is a straightforward, issue-by-issue technique based on automated tools and checklists. It should be noted that there are other methods too, most notably the Use Case Scenario. This evaluation technique involves setting one or more tasks to be completed by a hypothetical disabled persona, similar to computer-based usability testing. Start by writing a brief outline of the task to be accomplished and then report on the accessibility of each step. This technique is most useful for application evaluation, e.g., Microsoft Outlook. Several use cases should be tried in order to avoid taking too narrow a path and missing features and functions of the software. Use case scenarios are the best way to assess the *functional accessibility* of an application; i.e., how accessible the application actually is for accomplishing tasks, rather than how well it conforms with a given accessibility standard.

The Details

Yes, the “devil’s in the details” but that’s also where your attention provides the most beneficial report for your clientele. Figure 1 shows a typical finding that demonstrates how to incorporate useful information for a comprehensive report:



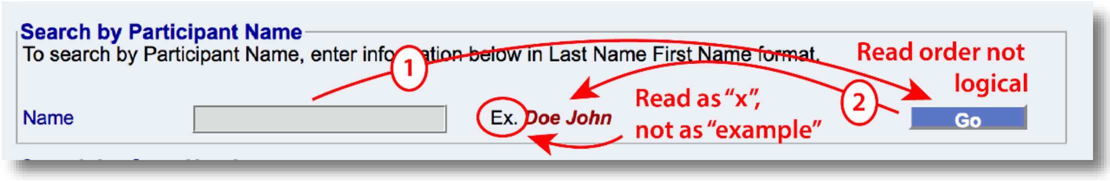
Sec. 6 - Manual Checks

1. ***P2*** – The images on the Courthouse Maps page (http://county-circuit-clerk/courthouse-maps/) and subsequent JPEGs do not have adequate ALT text. The thumbnail images of the courthouse should say something like, “Link to first floor map”. (Reference WCAG 2.0 1.1.1 Non-text Content)

Figure 1 – Issue example

Figure 1 shows the following:

1. **Prominent priority ranking.** Findings should be ordered to flow from highest to lowest priority. The severity of an issue sets its priority. Since showstoppers are the most severe, they come first in the list of issues in each section, followed by less severe problems. This reinforces the idea that these issues are critically important. Note: The severity ranking only applies to the finding at hand, not the site as a whole. If a showstopper is found, it deals only with a specific issue.
2. **Reference to the page where the issue was found.** Nothing is more frustrating than coming back to your report to check a finding that you can no longer locate. If you reference the problem page, reconnecting is easy for you and for your client. Note: Generic, template-based issues do not require a specific page reference, but it’s still a good idea to provide enough information so the client can find examples of the issue.
3. **Include Screen shots where needed.** Yes, they can take time and be a pain to produce, but a screen shot crystalizes your finding. It also demonstrates a universal design approach because you’re making the point in more than one way. Figure 2 is an example that illustrates a couple of issues.



Search by Participant Name

To search by Participant Name, enter information below in Last Name First Name format.

Read order not logical

Read as "x", not as "example"

Figure 2 - A screen shot makes the issue clear.

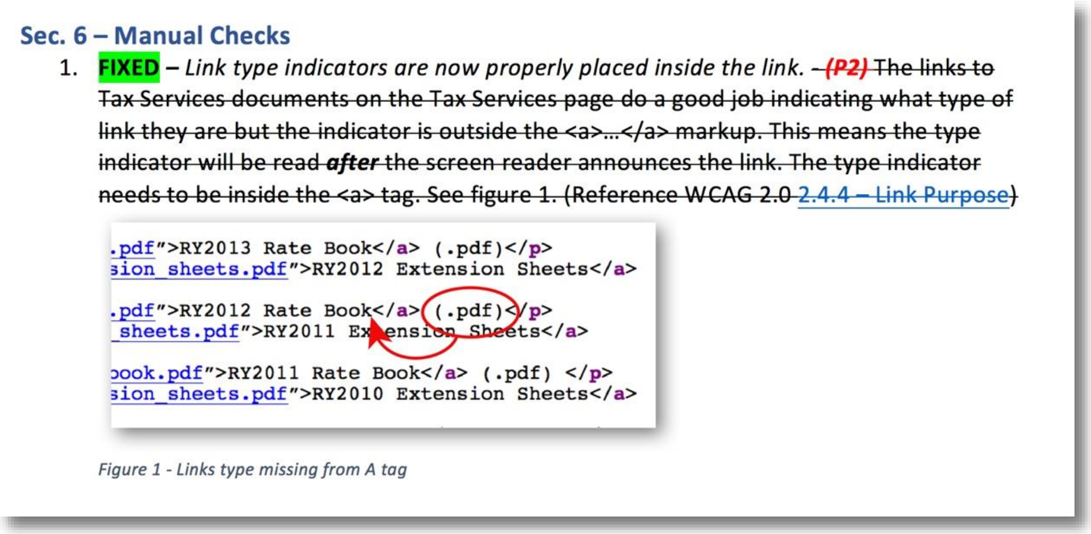
4. **Map your findings to the specific WCAG 2.0 guideline.** Mapping to the W3C shows that you're not making things up, and it lends an official note to your findings. If you're not certain which WCAG Guideline applies, automated tools sometimes reference this connection for you. You can add benefit by linking the reference to the "How to Meet" or "Understanding" pages provided by the W3C. There may be times when it's difficult to tie your finding to a specific standard. In those cases, a generic statement of meeting "best practices" should suffice. This is typical of usability findings.
5. **Include helpful references to other accessibility sites.** It's a good idea to link to sites like WebAIM and accessibility blogs that further explain and illustrate the issues you've flagged. Remember, you're trying to educate the client.

Follow Up

Once the report has been finalized and submitted, the plan is for the developer to perform the necessary remediation and then resubmit the site for a follow-up review. When you get the go-ahead to perform another review, of course it should be based on the initial report. This is because it's important to follow the same numbering order and structure, so you're not introducing confusion. In can be useful to work from a copy of the original report. New issues that come up can be placed in a new section or appended to an existing related finding. Be certain not to disturb the numeric identification of existing findings, otherwise the report will become disorganized.

As you go through the report and highlight various issues as "FIXED", be sure to provide a brief description of *how* each item was fixed, even if it's obvious. If you just say FIXED and then have to revisit the issue, there's no information on why it met your approval. Additionally, don't eliminate resolved issues. It's better to simply strike-through the old issue; that way you still have the original information in case it's needed for future reference.

Figure 3 shows a follow-up review and how the status of the issue is highlighted for easy identification.



Sec. 6 – Manual Checks

1. **FIXED** – *Link type indicators are now properly placed inside the link.* ~~– ***(P2)*** The links to Tax Services documents on the Tax Services page do a good job indicating what type of link they are but the indicator is outside the <a>…</a> markup. This means the type indicator will be read ***after*** the screen reader announces the link. The type indicator needs to be inside the <a> tag. See figure 1. (Reference WCAG 2.0 2.4.4 – Link Purpose)~~

```
.pdf">RY2013 Rate Book</a> (.pdf)</p>
sion_sheets.pdf">RY2012 Extension Sheets</a>

.pdf">RY2012 Rate Book</a> (.pdf)</p>
_sheets.pdf">RY2011 Extension Sheets</a>

book.pdf">RY2011 Rate Book</a> (.pdf) </p>
sion_sheets.pdf">RY2010 Extension Sheets</a>
```

Figure 1 - Links type missing from A tag

Figure 3 - Styling a FIXED issue.

Issues that are still outstanding need to be marked up similarly with text that says, "**NO CHANGE**". The highlighting is critical for showing that work still needs to be done.

Mobile Testing

As noted above, mobile/tablet testing needs to be in its own section, so that you can enlarge upon issues common to these environments without getting them confused with desktop issues. With the advent of Responsive Web Design, it is almost a foregone conclusion that a web site will be built with a sensitivity to the smaller screen. Unfortunately, without incorporating accessibility into the responsive design, that sensitivity does not carry forward for the user of assistive technology. For instance, the "hamburger" or collapsed menu does not announces its status for assistive technology without the use of specific techniques for accessibility. Because of this, users of the built-in assistive technology present on Android and Apple phones typically have no clue whether they're reading a menu or the page's contents.

Developers, at least the smart ones, are beginning to understand that the mobile/tablet environment is an entirely different interface paradigm with new and unique affordances. Manufacturers have done a good job building assistive technology into their mobile operating systems, but sadly, these interfaces change on such a fast life cycle that it's difficult to keep up. And not all version updates are steps in the right direction. Occasionally, a favorite accessibility feature disappears or ceases to work with a new update. Comparisons of the mobile market to the Wild West are not unfounded.

Unfortunately, the W3C doesn't help clarify the situation because there are no specific guidelines singling out a mobile interaction pattern. In fact, the W3C states,

> "Mobile accessibility is covered in existing W3C WAI accessibility standards/guidelines, particularly WCAG and UAAG, […]. There are not separate guidelines for mobile accessibility." (WC3 WAI, 2016)

This leaves one with a bit of a fuzzy feeling because evaluation is placed on a generic foundation.

All of this makes our job as accessibility evaluators even more challenging by needing to stay abreast of the mobile world. As an evaluator, it's important that you have, at least, a passing awareness of the accessibility features on the device. Be sure you utilize them to gain a full picture of the mobile experience. Despite the difficulties that mobile environments present, testing with the available assistive technology should give us enough ground to determine if there are accessibility issues present.

Assistive Technology (AT) Testing

Screen reader users who are blind debate whether sighted evaluators, with minimal experience using the software, should even attempt testing with AT. Their reasoning is that sighted users, aside from being inexperienced with screen reader software, often have difficulty overcoming the cognitive bias inherent with sighted use of a computer and do not understand how blind users actually use screen readers. Further, blind users are typically much better at deciding if a particular issue is a showstopper or may simply require a few extra steps; while novices may report items that are false positives because they didn't understand how to get the most out of the software. The same is true for other types of assistive technologies as well.

In spite of not being an expert, testing with assistive technology should not be overlooked. It's too valuable a tool to avoid, and it can give you firsthand experience as to why an interface may be inaccessible that cannot be gained any other way. The trick is to acknowledge when you suspect you're in over your head. Don't hesitate to be upfront and say an issue may be solvable for more experienced AT users. Skillful screen reader users can do amazing things, but there are always levels of expertise. Not everyone will be able to surf through a site in the same fashion, so even the novice assistive technology user should be able to gain experience and point out issues that are noteworthy.

One final important thing to keep in mind when doing an evaluation – don't rely solely upon one assistive tool. Each AT – including differing screen readers - does things differently, sometimes very differently, so it's important to acknowledge that they do not all function equally; test with at least the most popular tools. For example, one consequence of how different screen readers, such as JAWS and VoiceOver, work is that what seems like an issue in one screen reader may simply be a difference in how it works compared to another one. It is not always possible to create an identical user experience between different screen readers. The same goes for browsers; don't rely upon one for your evaluation, assuming it's giving you a universal experience. This doesn't mean that all cross-platform issues need to be resolved, but rather that the evaluator must at least be aware of what's going on and offer remediation suggestions where possible.

Formatting

A word about how to format your report. There are a few factors to be considered:

- Does the format lend itself to readability? Part of being consumable is the ease with which one can read through a report without getting lost or confused.
- Can you utilize screen captures, tables or other "non-textual" data in an easy way that allows you to add alternative text? Don't assume your audience does not include people with disabilities.
- Is it easy to track the status of an issue? Can you highlight and make remaining issues standout?
- Can the final report be put into an accessible format that is easy to distribute?

Using Microsoft Excel, or spreadsheets in general, is a popular format with some definite pluses. It's easy to itemize and categorize issues in a spreadsheet. You can also easily indicate the status of each issue. But there are also definite drawbacks that need to be considered. For instance, readability suffers drastically when using a spreadsheet; only statisticians like to read them. Adding screen shots, while not impossible, can be cumbersome. Finally, translating the final spreadsheet into an accessible and readable format will require some work.

Microsoft Word is a good option because it accommodates all of the factors mentioned above. Additionally, Word files readily convert to PDF, which can be made accessible for easy distribution.

Conclusion

Accessibility evaluation is often viewed as the process of *finding problems* with a given product. Viewed that way, we're immediately in negative territory where the developer feels pinned down, with accusatory fingers pointing at him or her, saying, "You did this wrong." To help dispel that kind of environment, it's best to take advantage of every available tool and method to help you succeed. Creating a consumable, easy to understand and clear report is a big step towards establishing the kind of win-win relationship you want with your client. After all, you want them to see you as being on the same team, working to create the best possible product.

There are multiple ways to successfully communicate the issues; this paper offers one of those ways, and the author hopes that it provides additional food-for-thought on how you may be able to improve your preferred method.

References

W3C Web Accessibility Initiative (2016). Retrieved from https://www.w3.org/WAI/mobile/

Basic Accessibility Checklist (2014). PDF retrieved from http://itaccessibility.illinois.edu/sites/itaccessibility.illinois.edu/files/BasicAccessibilityChecklist.pdf

Minimal Functional Accessibility Requirements for Software Candidates (2014). PDF retrieved from http://itaccessibility.illinois.edu/sites/itaccessibility.illinois.edu/files/Purchasing_Accessibility_Checklist.pdf

Copyright notice

Why Usability Testing Should be Part of your Accessibility Testing Strategy

Peter McNally

User Experience Center
Bentley University
Waltham MA, USA
pmcnally@bentley.edu

Abstract

Usability testing is an accepted process from the domain of user experience (UX) to gauge how easy to learn or use a product is. Users are asked to work on typical scenarios or tasks on the product with the goal of trying to identify challenges in the design of the user interface, interaction or content. In this paper I will discuss the benefits of usability testing in conjunction with accessibility testing and how they can work together. In particular, I will review some recent usability testing projects where we had people with disabilities participate and demonstrate how these findings can augment and support accessibility testing efforts.

Introduction to Usability Testing

At its core usability testing is a method where we ask an individual to try and use something and learn from that experience. Usability testing can be performed on anything that a human has to use including digital products such as a web site or a mobile app. There are several different approaches to usability testing, but the most common one used in the UX industry follows a *qualitative* approach; that is between 5 to 12 participants are tested over one or two days, each session lasts between 30 to 90 minutes. Participants are given multiple tasks or goals to accomplish during the session. Tasks types may range between open ended and specific. Table 1 shows three examples of usability testing tasks.

Table 1 – Example of usability testing tasks

Product Type	Task Type	Example Usability Testing Tasks
Electronic component mobile app	Completely Open Ended	Let’s say you are working on a project and need some information about a product related to what you described earlier. How would you go about learning more about it from the app?
Career Web site	Less Open Ended	Now, I’d like you to imagine that you’re satisfied with what you have seen so far and you are interested in working at this company. Go ahead and see if the site has a position you would want to apply for.
Clothing web site	Specific	You saw a friend’s daughter wearing this *Summer Green Surf Floral Swimsuit* and are considering getting one for your daughter. Please, find it and add it to your cart

Each participant conducts a test session one-on-one with a usability testing moderator. Participants are asked to think-aloud as they work through the tasks. The goal is to see if people can use the product "cold" such as the use of most consumer web sites where users are not expected to learn how to use it beforehand. Participants in usability testing should have a desire, need or goal in using the product tested. Anyone fitting this definition including people with disabilities are appropriate candidates to participate in usability testing.

After working with a relatively small number of participants it is typical to identify many common issues. Usually the usability testing moderator or team running the usability test will produce a report outlining the main issues and recommendations to improve the product tested.

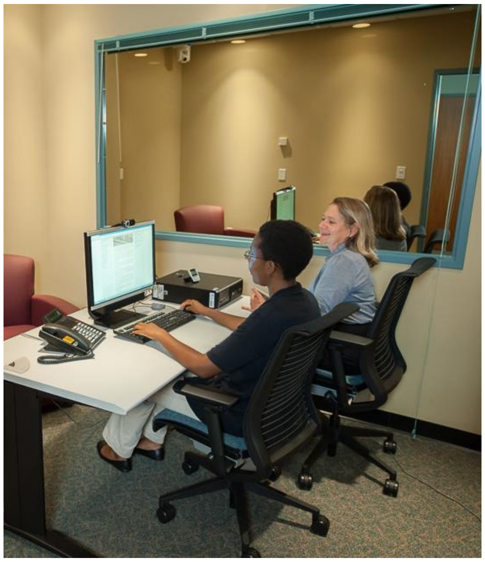


Figure 1 – Typical usability testing set up in a Bentley UXC lab.

Some Real World Examples

The User Experience Center at Bentley University is a consultancy that provides UX services including usability testing to our clients. We try to bring in wide range of participants in all our user research. Recently, several of our clients have explicitly asked us to work with people with disabilities when evaluating the user experience of their web sites. These client examples are all well-known large US based corporations in industries ranging from banking to publishing. Depending on the client requirements we have worked with individuals with a range of various and/or multiple disabilities including visual, motor, and learning disabilities.

Participants with Disabilities

Table 1 below provides a summary of the disability types of 37 participants in four usability testing projects. The majority had a visual impairment (20) with blind individuals mostly using JAWS. Individuals with low vision either used ZoomText, relied on default browser enlargement, or a combination of ZoomText and JAWS. Individuals with motor disabilities (6) included people using Dragon Naturally Speaking, an adaptive pointing device, and/or a wheelchair. Individuals with learning disabilities (10) included people with Dyslexia, ADD or

ASD. Note: while some participants had multiple disabilities, for simplicity of the table only one disability is represented per participant.

Table 1 - Overview of participants' disability categories in four usability testing projects

Industry/ Disability	Blind	Low Vision	Color Blind	Motor	Learning/ Cognitive	Total
Banking	4	1		3	2	**10**
Publishing	5	3				**8**
Chemical	4	1	1		4	**10**
Pharmaceutical	1	1		3	4	**9**
Total	**14**	**6**	**1**	**6**	**10**	**37**

Issues Found During Usability Testing

It would have been ideal if all major accessibility issues were resolved prior to usability testing. All four web sites had some level of accessibility testing prior to usability testing, however as an external consultants we was not privy to either the process or outcome of prior accessibility testing. Therefore my team observed participants encountering both accessibility and usability issues during usability testing. Accessibility issues were defined as issues that should have been found in a thorough accessibility review such as violations of WCAG 2.0. Usability issues are defined as not accessibility related, but have a negative impact on the overall experience.

Table 2 – Summary of issues discovered during four usability testing projects

Issue Category	Accessibility	Usability	Total
Content/readability/Page Layout	12	26	**38**
Search	21	11	**32**
Forms/interactive pages	16	10	**26**
Process/Understanding Concepts	0	19	**19**
Navigation/Menus, Orientation	7	10	**17**
Registration/Log in/Captcha	5	7	**12**
Error handling **Required fields**	5	4	**9**
General/Other	5	3	**8**
Information Architecture/Site Structure	0	4	**4**
Total	**71 (43%)**	**94 (57%)**	**165**

Table 2 shows that there were slightly more usability issues than accessibility issues discovered. Since the goal of usability testing is not to discover accessibility issues it may seem unexpected that 43% are accessibility related. The majority of accessibility issues identified ranged from simple, such as inappropriate alternative text on images to more complex issues, such as applying filters on search results automatically without providing appropriate feedback.

Many of these accessibility issues would be familiar; as you probably would have seen similar examples countless times in your testing. What is the more interesting are usability issues that while not strictly accessibility related are more likely to be found in context of usability testing with people with disabilities. Finding issues like these are one of the several benefits of conducting usability testing with people with disabilities. Two such examples are discussed below.

Benefits of Usability Testing for Accessibility

Since usability testing focuses on scenarios of use, users are asked to focus the complete end-to-end process to accomplish their goals. In doing this they may bypass what are thought of as critical accessibility issues or get tripped up on something that was thought to be minor. As I have seen through my usability testing, access does not always mean useful or usable. A web site may be judged accessible, but still present challenges for people trying to accomplish their goals. Usability testing can provide many benefits including:

Focusing Design and Development Resources

All accessibility issues should be fixed, however design and development resources are scarce and providing a team with a long list of issues to address can be overwhelming. Usability testing can prioritize the order of addressing issues around what really hiders users in accomplishing their goals – what they need to get done with the product. In the case of the usability testing conducted on these four sites clear patterns emerged on critical areas to focus on based on the tasks we asked participants to work on, such as:

- Search – all sites required users to search for some content such as a job or a publication. If users can't search effectively then they may give up and go elsewhere.
- Content – this is web page that contained the information users were looking for. Issues included its readability and accessibility such as appropriate use of alt text and headings for good structure.
- Navigation – moving around different pages using the sites' menus and other wayfinding elements was sometimes not clear or not accessible to screen readers.
- Forms – filling out online forms was sometimes challenging from an accessibility and usability perspective including inappropriate use of field labels.
- Process – having enough information to understand where they were in the process is a key usability attribute. In several cases participants did not know where to go next.
- Error handing and required fields – providing poor feedback on which fields are required and good error handling. Good design should avoid errors in the first place and helping users realize that there were errors and how to fix them.

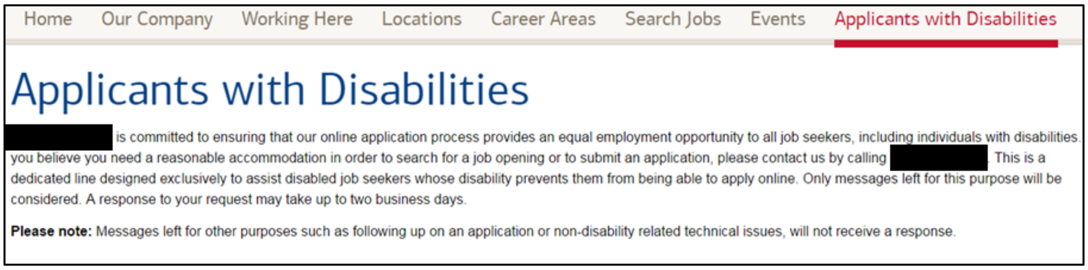
Discovery of Problems Learned in Context of Use

In addition to focusing efforts on fixing what is on the critical user path usability testing can identify accessibility related problems that may only be seen in the context of use. A real user may do things that were never thought of in any technical test case. Furthermore, by talking to real users you can understand the “why questions” about why the issue is important for them and how it impacts their over experience and perception of the organization. Here are two examples from usability testing.

Example 1: Bank site with page for applicants with disabilities

On the bank web site, participants assumed when they say a section that called out “Applicants with Disabilities” the page would be customized for people like themselves. However when they clicked on the link they were disappointed as one participant attests:

> “Oh no, really?! …I thought I was going to go to a page that would have a screen reader friendly page or TTY number for people who are deaf…feels like I was tricked” — *bank web site usability testing participant.*



Applicants with Disabilities

[redacted] is committed to ensuring that our online application process provides an equal employment opportunity to all job seekers, including individuals with disabilities. you believe you need a reasonable accommodation in order to search for a job opening or to submit an application, please contact us by calling [redacted]. This is a dedicated line designed exclusively to assist disabled job seekers whose disability prevents them from being able to apply online. Only messages left for this purpose will be considered. A response to your request may take up to two business days.

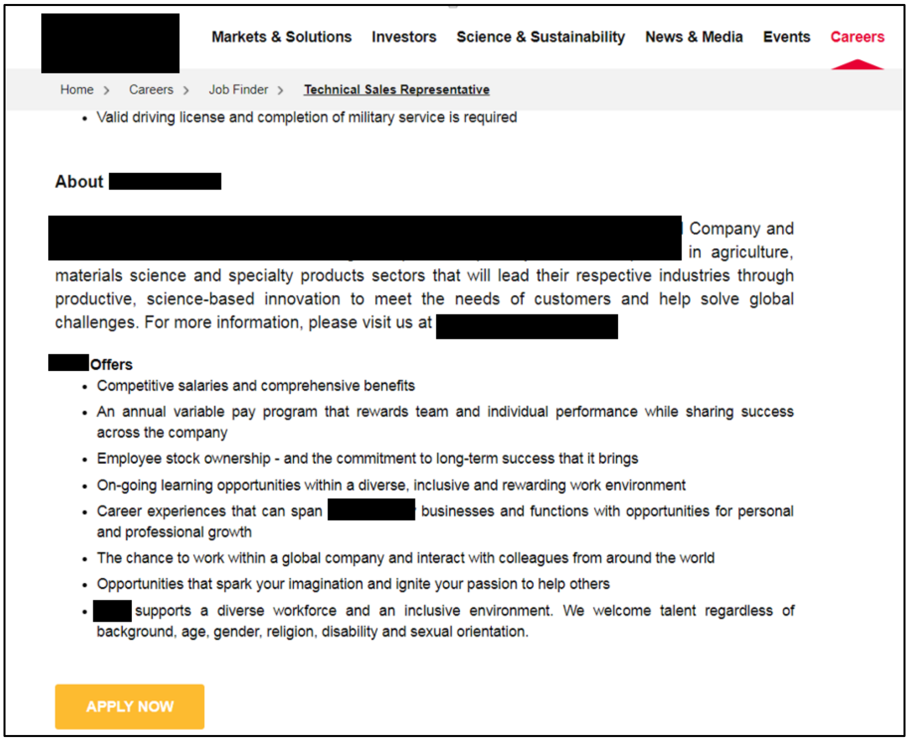
Please note: Messages left for other purposes such as following up on an application or non-disability related technical issues, will not receive a response.

Figure 2 – Bank site with page for applicants with disabilities

Example 2: Chemical Company Job Description

On the chemical web site, for one task participants had to review a job description to see if would be something they were interested in. Due to the heading structure some participants using screen readers missed some the job description because when they heard the “About <Company>” information they assumed there were at the end of the job description and did not continue on, therefore missing important information about company benefits, for example one participant mentioned:

> “Right now it’s a big long list and it’s pretty extensive. … You don’t know if you’re missing part of the content…right now once you reach Qualifications, and you hit H, it throws you back at the top and you don’t know what you’re missing.”- — *chemical company web site usability testing participant.*



[redacted]

- Valid driving license and completion of military service is required

About [redacted]

[redacted] Company and in agriculture, materials science and specialty products sectors that will lead their respective industries through productive, science-based innovation to meet the needs of customers and help solve global challenges. For more information, please visit us at [redacted]

[redacted]**Offers**

- Competitive salaries and comprehensive benefits
- An annual variable pay program that rewards team and individual performance while sharing success across the company
- Employee stock ownership - and the commitment to long-term success that it brings
- On-going learning opportunities within a diverse, inclusive and rewarding work environment
- Career experiences that can span [redacted] businesses and functions with opportunities for personal and professional growth
- The chance to work within a global company and interact with colleagues from around the world
- Opportunities that spark your imagination and ignite your passion to help others
- [redacted] supports a diverse workforce and an inclusive environment. We welcome talent regardless of background, age, gender, religion, disability and sexual orientation.

Figure 2 – Chemical company job description

Focusing on the user workflow rather than specific technical issues out of context shows how a low priority accessibility issue can mess up an entire user goal or process.

Supporting Individuals with learning or cognitive disabilities

Many issues that are considered usability issues are similar to issues that individuals with learning or cognitive disabilities may encounter such as having to remember things from page to page, unclear language/terminology, etc. While issues like these can be challenging for everyone, for people with learning and cognitive disabilities they present greater challenges based on the individual, the environment, or their context of use. Accessibility guidelines such as WCAG 2.0 tend to focus less on these issues since that are not based on code. However, issues found through usability testing and applying design principles such as Jakob Nielsen’s usability heuristics, e.g., “Recognition rather than recall” and plain language guidelines can provide a solid foundation to move in the right direction in conjunction with WCAG.

Getting the word out about accessibility

Having stakeholders (clients, developers, managers, etc.) watch usability testing sessions puts a face on accessibility for some who were not aware (or didn’t care). Inviting developers to watch usability testing can help them understand the importance of their impact on successful user experience in general, but in particular for their users with disabilities.

Providing a good way to "jump start" an accessibility program

For organizations that need a "kick in the pants". There is nothing like seeing an actual user or potential user struggle during a usability test for starting a fire to move efforts in the right direction. As standard practice we always invite our client to observe first hand either in person in our observation room or observing remotely watching via live stream.

Figure 3 – People observing a usability test at a Bentley UXC lab

Recommended Approaches

Continue to conduct full accessibility reviews by using automatic and manual processes. Once you are satisfied the big issues are resolved then plan on conducting usability testing with individuals with disabilities. Usability testing can be conducted in a conference room or specialized lab. It is best if participants can bring their own AT to the usability testing location. Conferencing tools such as Zoom offer the promise of an accessible experience for remote usability testing so participants don't have to travel, therefore avoiding logistical hurdles.

There are several different approaches for recruiting participants. Work with your UX team or client, partners (e.g., local disability advocacy group) in bringing in the right participants. Initially it may make sense be bring in a range of participants with different disabilities: persons with visual disabilities, persons with learning disabilities, deaf persons, etc. This will help you see if any one group has issues you didn't expect. Subsequently, usability testing could focus on specific groups such as persons with learning disabilities to focus efforts. Regardless of the mix of disabilities you recruit, the participants need to match the profile of the target user for the product so the scenarios make sense. For example, if you are testing a retirement site from a financial services firm, then all the participants might need to be active participants in a retirement plan and have a level of financial knowledge that match the user profile/persona.

Moving Forward

This process of conducting usability testing after accessibility testing assumes we are testing a finalized product or least a product that has been coded. However, we know from UX design that in most cases once coding starts some deep usability problems are baked in. It may be too late or too expensive to change. Just as it is important to consider usability testing after accessibility testing it is equally important to consider accessibility during the UX design process. Accessibility testing professionals should be involved in design before coding starts, such as developing user personas and reviewing wireframes, e.g., asking how will someone only using the keyboard or a screen reader access the primary navigation menu, etc. In addition, many UX professionals do not have detailed knowledge on accessibility or are intimidated by the level of technical knowledge required. Consider proving informal training such as “lunch and learns”, etc., to provide your UX team more knowledge on accessibility. By working together we can make a great experience for all!

References

Five Steps to Plain Language. (2017). Retrieved from http://centerforplainlanguage.org/learning-training/five-steps-plain-language/

Lewis, C. (2010). *ICT and People with Cognitive Disabilities: Variations in Assistive Technology*. User Experience Magazine, 9(2). Retrieved from http://uxpamagazine.org/variations_assistive_technology/

Henry, S.L. (2007). *Just Ask: Integrating Accessibility Throughout Design.* Retrieved from http://www.uiaccess.com/accessucd/ut_conduct.html

Horton S., Quesenbery W. (2013). *A Web for Everyone: Designing Accessible User Experiences*. Brooklyn, NY: Rosenfeld Media.

Nielsen J. (1995) 10 Usability Heuristics for User Interface Design. Retrieved from https://www.nngroup.com/articles/ten-usability-heuristics/

Rubin J. (1994). *Handbook of Usability Testing*. New York NY: John Wiley & Sons.

Copyright notice

Accessibility: WCAG, Usability, and Design

Irfan Ali

Principal Research Systems Specialist
Educational Testing Service (ETS)
660 Rosedale Road, MS 03-C, Princeton, NJ-08541, USA
iali@ets.org

Jason White

Associate Research Scientist
Educational Testing Service (ETS)
jjwhite@ets.org

Jennifer Grant

Senior Research Systems Specialist
Educational Testing Service (ETS)
jgrant@ets.org

Abstract

Developers of applications and web sites are familiar with the concepts of design, usability, and increasingly accessibility. Accessibility testing can reveal WCAG (Web Content Accessibility Guidelines) conformance issues, which can be corrected by developers, but the issue of usability implications for assistive technology users is less clear. Technical conformance to standards is not enough to ensure a usable, accessible experience. Discovering usable accessibility issues after the design and development work has already been completed can result in a sub-standard experience for a user who has a disability, and retrofitting usability can be costlier than simply retrofitting missing alt text or labels. This paper provides recommendations to developers, designers, and architects with the goal of producing a truly usable experience for all users, including those who have disabilities, while simultaneously conforming to the WCAG standards.

Introduction

Testing web sites and applications for accessibility, particularly those related to educational content, is a complex process that involves not only evaluating the strict application of WCAG and WAI-ARIA standards but also determining whether an interface is actually usable by someone with a disability, taking account a range of disabilities and assistive technologies. With educational content, for example, Science, Technology, Engineering, and Mathematics (STEM), interaction and content complexity can go well beyond what is typically evaluated in commercial or governmental web sites. Rather than making simple recommendations to retrofit an existing application to bring it into conformance with WCAG, for example, we must evaluate the application as a whole, bearing in mind the overall design goals, for example, in education, to demonstrate, interactively, a principle of mathematics or biology. The combination of the content and the interface used to interact with it can be significantly more challenging when factoring in the use of a screen reader, for example.

One challenge can be timing of accessibility testing. When accessibility testing is introduced late in the development process, well after major architectural and development approaches are decided, discovery of accessibility or usability issues can be more difficult to resolve, especially if significant software changes are required. At this point, there may be no magic wand solution for making the content accessible. While WAI-ARIA is often seen as a way to make code accessible, applying it in complex interactions can be time consuming and potentially costly depending upon the structure of the underlying code, and it may fall short in addressing all identified obstacles. More difficult still is making changes to the overall interaction design or visual presentation, if accessibility testing reveals fundamental problems there. The time to think about accessibility is at the beginning of the process – at the design stage, and to evaluate accessibility throughout the software development lifecycle.

An additional challenge to accessibility evaluation is the myriad of frameworks and different sets of technology-stacks that are being used by development teams. User interface frameworks, whether commercial or open source, have varying levels of accessibility support. Accessibility needs be considered up front in the design process, and decisions revolving around which framework to use must be made with accessibility in mind. Claims that libraries or frameworks support accessibility or have specific accessibility features should be verified carefully, with respect to the version of the software to be used. Further, the target environment for the application, for example, which browsers and assistive technologies are required to be supported, will influence tool and design selection. It can be argued that accessibility testing should be conducted on the software development frameworks themselves to ensure that they can be successfully used to develop accessible applications.

Accessibility Tips for User Interface Designers

To create an optimally accessible experience, start early, in the application user interface and content design process. Here are some basic guidelines for every designer:

- **Plan Heading and Semantic Structure Early:** Ensure that all content and design fits into a logical heading or region structure.
- **Consider Reading Order:** The reading order should be the same as the visual order
- **Understand the challenge of navigating non-**visually: Ensure the structure supports efficient navigation between critical areas which a user may have to refer to in order to complete.
- **Provide Good Contrast:** Be especially careful with light shades of gray, orange, and yellow and ensure that selected colors meet the ratio mandated by the WCAG guidelines. See https://www.w3.org/TR/UNDERSTANDING-WCAG20/visual-audio-contrast-contrast.html
- **Avoid Images of Text:** Use actual text, rather than an image of text. It enlarges better, loads faster, and is easier to translate. Use CSS to add visual style to the text.

- **Use Caution When using Capital Letters:** All caps can be difficult to read for some users and can be presented incorrectly by screen readers
- **Use Adequate Font Size and Remember Line Length**
- **Make Sure Links are Recognizable:** Differentiate links in the body of the page with underlines or something other than color alone
- **Design Focus Indicators:** Ensure that interactive controls, such as buttons and links, have a clearly discernable focus ring. Sighted keyboard users rely on focus indicators to identify location. Never consider a design where you disable the default focus indicator.
- **Design a "Skip to Main Content" Link:** Place a keyboard accessible link for users to skip navigation at the top of the page
- **Do Not Convey Content with Color Alone:** Some users can't distinguish colors or may override page colors
- **Design Accessible Form Controls:** Ensure form controls have descriptive labels, instructions, and validation/error messages
- **Provide a design for responsive layout:** Ensure that you provide a design to accommodate content in a responsive layout, so that a user who needs to enlarge text on the page can do so without having to scroll horizontally to read the end of each line.

Accessibility Tips for Developers

We had identified some basic tips for developers that focus on factors that can influence overall accessible user experience. Bringing accessibility expertise into the development process and to have subject matter experts embedded or readily available can avoid many issues.

Tools and Frameworks

Selecting the Right Framework/Widget Library is critical. Common frameworks may have components and widgets with built-in accessibility, but we have found cases where documentation did not accurately describe accessibility support or did not provide adequate documentation on what developers would be required to do to implement accessibility features effectively. Further, stand-alone components may be accessible in a test case, but when integrated into an application can introduce accessibility issues. When framework issues are found, it is critical to file a bug with the vendor or open source project owner. But if a fix is not forthcoming, seriously consider whether the framework itself is the right choice. We highly recommend identifying and using accessible widget libraries with robust components.

Performance

Application performance can seriously affect accessibility. A poorly performing application can cause lag time that may be confusing to some users, particularly those who rely on screen reading software. Backend developers should tune up performance, and performance should be managed properly on both the back end and front-end layers to ensure low latency for application loading and functions. Here are some basic performance guidelines for developers:

- **Browser Caching:** Leveraging the browser cache is crucial for assets that are rarely changing. In such cases, a maximum-age of 7 days is recommended.
- **Reducing the HTTP requests:** Developers should minimize the number of HTTP requests
- **Minify CSS and JavaScript:** Minification of resources means removing unnecessary characters from your HTML, JavaScript, and CSS, reducing load time.
- **Error Handling:** In case of an error while accessing the application, provide the user with a link or proper information that, for example, provides contact information for customer/end-user support
- **Database Optimization:** Whether it is a cleaning out old unused tables or creating indexes for faster access, there are always things that can be optimized

Dynamic Content

As a general rule, when content within the page is updated dynamically, the user should be given control over those content updates, for instance by activating a link or button. If content updates automatically without user intervention, there is a risk that users of assistive technology will miss that changing information or be confused by it. That is, if an element that currently has focus is removed or significantly changed, keyboard focus may be lost, sometimes reverting to the top of the page, thus disorienting a screen reader user.

Status updates as an example, if a page displays updated data every 10 seconds, it’s quite possible that the automatic content updates will not be accessible to screen reader and keyboard users. And worse, the rest of the page content may also be inaccessible because the user cannot reach it without first encountering the dynamically updating content.

Using WAI-ARIA, the developer can identify content that dynamically change as a live region. A live region allows for the presentation of content updates in a way that is appropriate for a screen reader user. It allows the developer to control the level of interruption (assertive or polite, etc.), the kind of information that is presented (additions, deletions, etc.), and how much of that information to present.

When using live regions, early testing with screen readers will be essential to confirm that the updates are announced as intended.

Conclusion

Accessibility Testing and WCAG conformance are only part of the answer when developing applications that are accessible and usable by your target audience. The design and development effort must take accessibility into account at the earliest stages, from initial design to selection of development tools and frameworks to be used in implementation. Late stage accessibility testing is not the time to determine that you have fundamental design issues impacting the ability for users of assistive technologies to successfully use your system.

Copyright notice

Why Test: Getting Value from Accessibility Testing

Preety Kumar

Deque Systems
Preety.kumar@deque.com

Abstract

This session will focus on the role of testing for WCAG 2.0 checkpoints and techniques throughout the software development life cycle. While a lot of focus has been placed on understanding and debating the technical requirements, organizations still struggle with understanding what testing can be automated, how much manual testing must be performed, by whom, and how often. If an organization has made accessibility testing a mandate, the challenge of how to effectively take action on the test results to improve accessibility still remains. I will walk you through what you can do to ensure your test results don’t go into a black hole.

Summary

An effective accessibility program proactively defines not only what, when, and how to test – but also **why** to test. This insight will help the organization clarify:

- What metrics are most important to baseline and track – and how will testing support this initiative?
- How will test results be consolidated, tracked and reported? To whom?
- What decisions will be made? By whom? To what end?
- How can automated testing be augmented by specific, targeted manual testing in a way that nets quick, actionable findings?
- How can accessibility test results justify funding of priority remediation activities?
- Which test results can help define cross-team training plans?
- How can test results be structured to not only monitor the organization’s accessibility maturity level – but to actually improve it?
- In what ways can testing results be tied to governance?

Running an effective accessibility test program requires some planning and technical management to ensure tests are executed efficiently, consistently, and that the results are tracked successfully to completion (i.e. code is actually fixed to address defects coming out of the tests). Below are some topics this session will include, wherever possible using real life case studies and examples:

1. How to perform a comprehensive test without breaking the bank? Layering in accessibility at the right time in the dev lifecycle while tracking progress is a challenge,

especially in the Agile world of development. Getting your arms around the scope is often the first step. Understand how to scope and prioritize based on risk.

2. How to plan and structure tests for effective results? There is no substitute for planning. Treat accessibility testing as you do any other testing. Align test goals with business users, invest in up-front automation, set up test cases intelligently, allocate dedicated time and resources, have project management in place and you will see how results from your accessibility testing efforts will give you required coverage and actionable results.
3. How do you create a good monitoring plan? Do you have a good monitoring and incident response plan in place? Plan for a review of how you respond to complaints and how you review the state of accessibility of all your digital properties. Understand how to use data from monitoring reports - where there is smoke, there is fire!
4. How to effectively communicate and align with the business? Review requirements and the accessibility plan with the business stakeholders and the IT team. Pre-test requirements will need cooperation from different people in your organization. The more you keep all concerned parties in the loop, the easier your accessibility test and remediation planning will go. Remember, no one likes surprises.
5. What should I do with the test results? After the test, make sure you provide the team with actionable results qualified by the right frame of reference. Most accessibility tests will be recorded in a standard report. Is that the best way for this data to be consumed? You may generate a summary report that provides context and background to the results and perhaps comparing them over time making sure they are an apple to apple comparison (same scope, same tests). However, the defects you report may be better consumed if you enter them in a defect tracking system where development managers can assign user stories/defects, understand the impact of story points on the development effort, and plan for what will be required to fully implement the accessibility backlog.
6. What should I report to management? We'll review best practices of separating the management (provides a summary) and technical reports (goes in depth), what a good mitigation plan should contain, and what metrics should be tracked over time. Separating out reports based on who can take action is also going to result in happier consumers of the reports. For example, if I cannot change content, don't tell me what is wrong with the content. If I can't touch code, don't tell me defects in the calendar widget. We will review best practices in reporting and how reports be used to impact change. Detecting patterns (are the same issues occurring over and over again) can be very powerful in informing training programs and addressing knowledge gaps.

Discussion Topics

Some topics for group discussion include:

- Is your organization currently testing? Why? What are the objectives for testing?
- How are reports produced? To whom are these delivered? How are the reports used?
- Does your organization use testing data to define training? (For developers? For content authors? For project managers? For others?)

- Are test results used to support funding requests? (For training? For remediation activities?)
- Does your organization plan to use trended test data to optimize practices?
- Is your organization utilizing accessibility test data to do something unique and actionable?

Ad hoc discussion topics may also develop, these are welcomed!

Acknowledgment

Many Thanks to Debra Martin.

Copyright notice

Symposium Committee

Chris M. Law (Chair)
Organizational Accessibility Consultant, Accessibility Track Consulting, LLC

Norman Robinson (Co-Chair)
Agency Section 508 Program Manager, Department of State

Laura Renfro (Marketing & Outreach Chair)
Senior Accessibility Lead, Renfro Consulting

John Rempel (Awards Chair)
Quality Control & Training Specialist, AMAC Accessibility Solutions & Research Center at Georgia Tech

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

Matt Feldman
Director of Services, The Paciello Group, A VFO Company

Katherine Eng
Accessibility Specialist, Assistive Technology Resource Center (ATRC), Department of Justice

Karl Groves
Founder and President, Tenon.io

Sunish Gupta
Founder, MIT – Easy Alliance

Allen Hoffman
Deputy Executive Director, The Office of Accessible Systems & Technology, Department of Homeland Security

Maureen Kraft
Transformation Lead, IBM Accessibility

Preety Kumar
Founder & CEO, Deque Systems, Inc.

Mark Lapole
Lead Product Manager, Accessibility, eBay

Anil Lewis
Executive Director, National Federation of the Blind Jernigan Institute

Christine Loew
Director, Accessibility for Digital Assessments, The College Board

Earl McJett
Section 508 Coordinator, Federal Deposit Insurance Corporation

Eduardo Meza-Etienne
President & CEO, Meza Consulting, LLC

Karen Moyes
Section 508 Coordinator, Westat

Andrew Nielson
Senior Program Manager, New Editions Consulting, Inc.

David O’Neill
Managing Director, Digital Inclusion for the American Foundation for the Blind, & CEO, AlwaysAccessible, Inc

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

Cyndi Rowland
Executive Director, WebAIM, Utah State University

Gian Wild
Founder, Owner & CEO, AccessibilityOz

Author, Panelist and Speaker Index