

Digital Accessibility Design & Development Engineering

Poornima Badhan Subramanian

IT Digital Accessibility, Holland America Group, Seattle, USA

Abstract The human-centric thought process is an important and essential aspect of product development. While user experience in product development is now a widespread concept, the fact of designing for inclusive design and development practice is still considered a discrete feature. The inclusive design ensures the product is accessible to everyone which means people of all abilities. The diverse population access digital products utilizing computer settings, assistive technologies, and handheld devices. This article covers thorough aspects and best practices of digital accessibility to consider in product development in both design and development along with satisfying Web Content Accessibility Guidelines universal standards for regulatory compliance.

Keywords Digital accessibility, WCAG, Accessibility engineering, Product development, UX design

1. Introduction

Product development comprises Planning, Design, Development, Testing, and Maintenance. With the introduction of the Agile implementation framework in many organizations, integration of each requirement becomes challenging with rapid sprint timeframes. The essential aspect for successful product development lies in educating, understanding, and learning the design and development principles well ahead and during each refinement practice. This article covers the topic of integrating the digital accessibility process into the product development life cycle, aligning the inclusive User Experience design standards, and adapting to the accessible development techniques to create accessible products that will lead to the creation of a digital product that can be accessed by everyone regardless of abilities.

2. Basis of Product Development

Understanding digital users' real needs in adapting and accessing technology: Designing a website accessible for everyone will be the motto of each product development solution, but the question is "Does the company or the team working on the product learn about their product's audience and curve out the possibilities of their needs?". Nowadays, the user experience, shortly known as UX came out with various trending frameworks to make digital websites appealing to a user, yet are they all usable and accessible? A major challenge around digital accessibility concerns the

present inability of technology to cover the diverse types of abilities. The digital technical solution must address the various spectrum of challenges faced by people with disabilities. For example, people with cognitive impairments like autism, or memory syndrome do not look for an appealing design only, but essentially a navigable and understandable way to complete a transaction with not losing the current step in the process, resume from where they left off, and know the suggestions relevant to their search, and more. The technology is developing rapidly now, but with more reactive approaches towards seeing digital accessibility as compliance rather than user experience. Another example is a senior user looking for ways to access the content through speech recognition, unlike a younger user who may look for a magnifying tablet to access the same content.

3. Web Content Accessibility Guidelines: What's in and can be Included?

Web Content Accessibility Guidelines, shortly known as WCAG, is a universal standard that provides measurable criteria to scale digital accessibility requirements. WCAG is framed into four different sets of product accessibility factors Perceivable, Operable, Understandable, and Robust. The guidelines offer more technical conformance level recommendations which may be challenging for every designer and developer to get to the core of it. This does not mean that the guidelines are not implementable, but with little education and with a Digital Accessibility specialist interpretation, the implementation becomes much easier to adapt into the product development life cycle. As now many organizations adapt to WCAG standards and start believing conforming to the standards will make their application accessible, there are considerable changes the W3C WCAG standards team to bring in to address the wide spectrum of

* Corresponding author:

pbtanisha@gmail.com (Poornima Badhan Subramanian)

Received: Jun. 28, 2022; Accepted: Jul. 10, 2022; Published: Jul. 15, 2022

Published online at <http://journal.sapub.org/web>

disabilities challenges a digital user could face from day-to-day.

WCAG standards currently cover the following

- Perceivable
- Operable
- Understandable
- Robust

What can be covered in upcoming WCAG standards –

- Predictable (for users with cognitive disabilities)
- Speech intelligible (for users of aged / motor disabled)
- Learnability / Memorability (for users with autism)

4. Digital Accessibility Integration into the Product Development Life Cycle

The step-by-step process in the Digital Accessibility life cycle is -

Requirements analysis: Create specifications to include Digital Accessibility standards in each product requirement to classify knowing the user base of the product.

Planning: The current and trending agile methodology includes Digital Accessibility guidelines mapping the product's design.

Product design (UX): Each product component requires to satisfy its relevant accessibility guidelines. For example, the form fields are expected to show visual labels instead of showing only placeholder labels.

Product Development: Learning and integrating the development best practices, and ARIA frameworks to adhere to Web Content Accessibility Guidelines is an enough step in developing accessible products.

Continuous Integration: An important factor in the product development life cycle is to maintain the consistency of its features for any new developments and changes planned for the product for the user experience to be accessible by everyone!

5. UX Design and Accessibility

A user interface application is a multi-dimensional combination of elements and components. The main 5 key factors of usability are

- empowering – it does what I want (functionality)
- efficient – it doesn't waste my time (action powered)
- easy – it's easy to learn and use (intuitive)
- engaging – it's a pleasant experience (articulate)
- trustworthy – don't feel a concern (feeling secured)

While the UX design trying to achieve these key factors, it is also important to know that these factors are applicable to each customer of the product. A user must be able to access the product how they could irrespective of using assistive tools and technologies. For example, I can use the mouse to scroll through the webpage and click on any elements, but another user may use the only keyboard to access the website

due to loss or inactivity of his or her limbs.

5.1. Universal Design Factors of UX Components

In brief, Universal design means the design can be accessible by everyone on the planet which is also sometimes referred to as 'Design for all'. Some would argue that universal design may not be an appropriate term, as the product audience may vary based on what it offers. However, the principles of universal design will never be a useless thing irrespective of the product's portfolio, as it defines the best practices of each and every interactive element of User Interface (shortly known as UI) that makes it accessible by everyone, including people with disabilities. Here listed are universal design and digital accessibility recommendations for the key (commonly used) user interface components -

5.1.1. Buttons

Show the buttons distinctly from other content with a good color contrast ratio. Position the buttons that help to relate to the context of the web page.

5.1.2. Links

Show the links distinctly from other content with a good color contrast ratio. Keep the text of the link short and clear to understand its purpose. For repeated links text, add context to it. Example: For 3 Quantity drop-down shown for different products, add the product name to the Quantity label e.g. "Quantity for Product 1". For generic links text, add context to it. Example: 'Read more' can be provided with the label "Read more about deals and promotions". Always remember to "match the visual text: Label the buttons as how it is shown visually e.g. 'Next Step', 'Submit', 'Cancel'.

5.1.3. Form Fields

Mark the required fields clearly. Show the visual labels that show on screen all the time, even while in focus. Display good color contrast for the form field borders that makes a lot of difference to distinguish. Limit to one column as much as possible which helps while zooming in. Highlight the error fields showing visual error text and error icons, not just by color. Place the 'Help' icon or link that supplies additional information next to the field. When possible, provide dynamic error messages visually (for missing or incorrect entries) when moving from one field to another field.

5.1.4. Menus

Keep only one (or two if needed) navigable menu for short housekeeping items, limiting to a maximum of 12 items in it e.g. 'About us', 'Contact us', 'Plan'. Create a good target area/space between each sub-menu item. As much as feasible, add a Close icon/link to close the menus.

5.1.5. Text

Always make sure the color contrast of the regular text (≥ 24 px) with its background color meets at least greater

than a 4.5:1 contrast ratio. For large text (≥ 24 px), the color contrast ratio with its background color expected to meet at least greater than 3:1. Create clear and sufficient spacing between letters (at least 0.12 times the font size), words (at least 0.16 times the font size), paragraphs (at least 2 times the font size), and lines (at least 1.5 times the font size).

5.1.6. Navigation

Include 'Skip to main content' navigation links as a first element of the page that helps users using screen readers to skip the repetitive menus and content at the top of the page which is present across all the pages and jumps directly to the main content of the web page. As much as possible, limit the navigation sections to only 'header' and 'footer'. Avoid adding side navigation bars unless it is really needed for the specific purpose of the design.

5.1.7. Focus Ring for Interactive Elements

Show the focus ring consistent and clear for all the interactive elements (buttons, links, menus, form fields, radio buttons, checkboxes, image links). The color contrast ratio of the focus ring with its background color must pass at least a 3:1 contrast ratio. Avoid showing a focus ring for the non-interactive elements (e.g. plain text, headers, error messages).

5.1.8. Multimedia

Include Captioning & Transcripts for the multimedia content (video, audio). Wherever applicable, include an Audio description to describe the multimedia content. The Audio description will be required for the multimedia content where scenes conveying emotions, backgrounds, and situations are included.

6. Coding Techniques for Creating Accessible Components

Before heading to the coding techniques, a few main principles of healthy coding are

- Use HTML roles, tags, and attributes in all ways (which works perfectly with many of the screen readers)
- Remember that the container elements `<div>` and `` used in the modern application frameworks can only be used for layout purposes. These container elements do not hold any meaning to it by default and require additional markup like the ARIA framework to make your coding accessible
- Remember '*No ARIA is better than Bad ARIA*'. It is important to carefully choose required ARIA roles and attributes for the correct interpretation of user interface elements' structure

6.1. UI Buttons Coding Techniques

Make sure the role of the button is coded (for the screen readers to announce as e.g. button, link, text field).

- Use HTML roles `<button>` as much as possible (e.g. `<button> Hello Coding </button>`)
- In modern applications, the usage of custom components and third-party custom libraries utilizing `<div>`s and ``s are more in favor in place of using HTML components. For those kinds, the same button can be marked up in a different way using ARIA (e.g. `<div role="button"> Hello Coding </div>`)

For additional context required for the buttons in case of same button text is shown for different items e.g. 'Save'

- Use HTML technique of 'hidden text' as much as possible (e.g. `<button> Save Product 1 to favorites </button>`)
- The same button can be marked up in a different way using ARIA `<div role="button" aria-label="Save Product 1 to favorites">Save</div>`

6.2. Web Semantics Techniques

Semantics convey the structure of a web page and how it is organized. Structural, semantic HTML is the key starting point toward good accessibility practices. Assistive technologies such as screen readers (used by people who are blind or have low vision), or scanning devices manipulate the structure of the web page from the semantic cues in the DOM order.

6.2.1. Headers

Code the headers in a logical order

- Limit one `<h1>` for a page to denote the main header of the page
- There can be multiple `<h2>` through `<h6>`s as to how it best describes the structure of the web page

Use HTML heading tags `<h1>` through `<h6>` e.g. `<h1> Heading Level 1</h1>`. For custom components using `<div>` containers, assign the heading tags through ARIA headers. e.g. `<div role="heading" aria-level="1">Heading Level 1</div>`.

6.2.2. Landmarks

Add landmarks to the different sections of the page

Use HTML landmarks tags as

- `<header>` define the header section of a web page. There can be only one `<header>` content in a document
- `<nav>` points to the navigation sections of a web page e.g. Menus
- `<main>` define the start of the main content of a web page. There can be only one `<main>` content in a document
- `<aside>` refers to the complementary sections in a web page e.g. Promotion or ad section present at the right corner of the page
- `<section>` refers to different sections of the page e.g. Products, Offers, Contacts
- `<footer>` refers to the footer section of a web page.

There can be only one <footer> content in a document

- <form> represents the collection of form elements

For applications utilizing custom components, the HTML equivalent form of ARIA landmark roles as

- role="banner" for header section
- role="main" for the main content
- role="complementary" for aside
- role="region" for section
- role="navigation" for navigation sections
- role="application" for widgets
- role="form" for a form section

Note: There are resources like SaIL: saliency-driven injection of ARIA landmarks [9] to inject landmarks into the web document for the use of screen reader users with keyboard shortcuts

6.2.3. Reading Order

The standard reading order of web documents is as below which can be achieved with DOM order

- Left-to-right
- Top-to-bottom

6.2.4. Table Structure

For the tables used to represent data structure visually, the relationship of headers and values also required to be conveyed for screen reader interpretation.

Use HTML table tags and attributes as e.g. <tr> <th scope="col">Table Column header <th scope="row">Table Row header </tr> <tr> <td> Table value 1 <td> Table value 2 </tr>.

"Using <scope> attribute is recommended for most of the data tables, except for complex tables, the 'id' and 'headers' may work best with screen readers".

For modern applications utilizing <div> and , use ARIA roles and attributes as e.g. <div role="table" aria-label="ARIA table" aria-rowcount="4"> <div role="rowgroup"> <div role="row"> ARIA Role </div></div> <div role="rowgroup"> <div role="row"> header h1 </div> <div role="row"> header h2</div></div>

6.2.5. List Items

It is required for the list items to be conveyed as lists for screen reader interpretation to understand the structure and meaning.

Use HTML list items coding using unordered list , ordered list and definition list <dl> e.g. <p> list item 1 <p> list item 2 .

For modern applications utilizing <div> and , use ARIA roles and attributes as e.g. <section role="list"> <div role="listitem">List item 1</div> <div role="listitem">List item 2</div> <div role="listitem">List

item 3</div></section>.

6.2.6. Group of Links

Use <div role="group" aria-label="Group heading"> <a>link 1 <a> link 2 </div>.

6.3. Form Elements Coding Techniques

Avoid using <div> and for the form fields

Code appropriate roles for the form elements e.g. 'input for edit field', 'radio for radio buttons', 'select for drop-down field'. For custom widgets combining two functions (e.g. Combo-box having both Edit field and drop-down) together, add proper instructions to the field for the screen reader to announce its purpose and operation.

7. UI Elements Benefit from the ARIA Framework

7.1. Menus

A couple of menu variations are:

1. **Fly-out menus** are most common in the form of primary navigation, search menus, hamburger menus
2. **Application menus** are rarely used either within the sections or as application controls of the web document

7.1.1. Coding Techniques for Menus

- Menus utilizing HTML + ARIA will make the best bet for screen reader interpretation
- Menus and sub-menus are keyboard accessible
- Coding parent menu as a toggle element to show/hide the sub-menus
- Avoid showing the sub-menus on keyboard focus
- When there is a long list of sub-menus present, split up the sub-menus under different (related) section headers
- Provide Close icons to close the sub-menus
- Label the menus when are more than one menu structure present in a web page (e.g. Main menu, Search menu)

7.1.2. HTML + ARIA Specifications for Menus

Though it is recommended to use most of HTML for flawless coding practice, some of the required/supporting ARIA roles and attributes (menu, menubar, aria-expanded, aria-haspopup) also help the screen readers to interpret menu structure appropriately.

```
<nav aria-label="Main">
<ul> <li><a href="#">Home</a></li>
<li><a href="#">Shop</a></li>
<li class="has-submenu" aria-haspopup="true">
<a href="#" aria-expanded="false">Space Bears</a>
<ul><li><a href="#">Space Bear 6</a></li>
<li><a href="#">Space Bear 6 Plus</a></li>
</ul>
```

```
</nav>
```

7.2. Tab Structure

One another custom user interface widget portrayed as 'tab structure' utilizes both HTML and ARIA roles and attributes to convey its structure – below is the sample coding snippet that utilized ARIA roles 'tablist', 'tab', 'aria-selected', 'aria-controls' to refer to the tab/tab contents.

```
<nav class="row tab-nav-row" aria-label="Alaska Cruise Ports">
  <div role="tablist">
    <a href="#" role="tab" aria-selected="true"
      aria-controls="302951-tab" tabindex="0">
      <span class="tab-nav-text">College Fjord</span></a>
    <a href="#" role="tab" aria-selected="false"
      aria-controls="303069-tab" tabindex="-1">
      <span class="tab-nav-text">Endicott Arm</span></a>
    </div>
  </nav>
```

7.3. Aria-label vs Aria-labelledby vs Aria-describedby

7.3.1. ARIA-label

Assigns the label for a UI element for one text element at a time.

- As the ARIA-label overrides the actual label for screen readers, always add the full label text as 'aria-label' e.g. `< a href="#" aria-label="View details of Alaska cruises">View details`
- Remember ARIA-label will not work on non-interactive tags e. ``, `<p>`
- Use ARIA-label on interactive roles & containers e.g. `<div>`, `<a>`, `<button>`

7.3.2. ARIA-labelledby

It is similar to aria-label for labeling purposes, except aria-labelledby can refer to more than one text element

- Use only when HTML `<label for>` attribute usage is not possible
- Primarily used to associate the label with the form field which is similar to HTML `<fieldset>` attribute
- ARIA-labelledby creates the relationship for the label, supporting text and the form fields e.g. 'input', 'select'

Example coding snippet:

```
<label id="text1" for="f3">Notify me</label>
<select name="amt" id="f3" aria-labelledby="11 f3 l2">
  <option value="1">1</option>
  <option value="2">2</option>
</select>
<span id="l2" tabindex="-1">days in advance</span>
```

7.3.3. ARIA-describedby

ARIA-describedby is similar to ARIA-labelledby, except it refers to the descriptive text

```
<button aria-label="Close"
  aria-describedby="descriptionClose">
```

```
  onclick="myDialog.close()">X</button> ...
```

```
<div id="descriptionClose">Closing this window will
  discard any information entered and return you back to the
  main page</div>
```

7.4. Custom Widget: Combo-box

Combination of input (edit) field and drop-down (list items) together

```
<label id="combo1-label" class="combo-label"> Favorite
  Fruit </label>
<div class="combo js-select">
  <div aria-controls="listbox1" aria-expanded="false"
    aria-haspopup="listbox" aria-labelledby="combo1-label"
    id="combo1" class="combo-input" role="combobox"
    tabindex="0"></div>
  <div class="combo-menu" role="listbox" id="listbox1"
    aria-labelledby="combo1-label" tabindex="-1">
    <!-- options are inserted here -->
    <!-- e.g. <div role="option" id="op1">option text</div>
  </div>
</div>
```

7.5. Range Widget: Slider

The slider role defines an input where the user selects a value from within a given range. The slider elements are used infrequently to choose values from the range of minimum value to the maximum value. This widget can be coded using different approaches HTML 5 slider, ARIA role="slider", ARIA role="application"

HTML 5 slider 'range' utilizes HTML attributes 'min', 'max', 'value' along with few of ARIA attributes 'aria-orientation', 'aria-label', 'aria-valuetext'

For applications using containers like `<div>`s, the ARIA role="slider" utilizes ARIA attributes aria-valuemin, aria-valuemax, aria-valuenow along with 'aria-orientation', 'aria-label', 'aria-valuetext' provide the same interpretation.

Example coding snippet –

```
<div class="deque-slider">
  <p>Amount you would like to pay today</p>
  <div>
    <input class="deque-slider-widget" type="range"
      min="50" max="5000" value="100" step="10"
      aria-label="Amount you would like to pay today"
      aria-valuetext="value:100">
  </div>
```

8. Cost Impact Analysis of Digital Accessibility Legal and Audits

A simple and quickly settled digital accessibility lawsuit would cost the defendant an estimated legal cost of up to \$350,000. But it does not stop there, the organization will also be brought under the Department of Justice agreement to ensure the defendant's digital content is made accessible within the specified timeframe that involves a large amount of audit and remediation efforts which would then add up to

approx. \$225 an hour of labor work that multiples based on the size of the application, number of project resources, and volume of issues identified. An estimation says the low end is around \$10K for a basic information/marketing website, an e-commerce site will fall into the \$25K-\$35K, and large enterprise sites can easily rise into the six figures zone. For public-facing websites like e-commerce apps or large enterprise applications, the cost impact analysis makes a significant difference considering the proactive approach which reduces the cost by up to \$5K whereas the reactive approach could lead to \$100K.

9. Conclusions

Digital accessibility is an inclusive practice in the product development life cycle that can be achieved with planning from its initial phase. The proactive approach of the design and development process considering digital accessibility best practices will not only make the product accessible for people experiencing disabilities but also usable for everyone! Think about the pinch zoom feature in mobile almost used by everyone nowadays, and the websites with good color contrast feature earning its user rewards, especially for the aging population. It is important for the business team to know the user base and consider digital accessibility from the requirements phase which will then be carried out to design and development teams for implementation. The UX best practices outlined in this article along with the coding techniques are the foundation for creating an accessible product that can be equally enjoyed by everyone. Keep your spirits towards universal access while designing or developing your product!

REFERENCES

- [1] Digital accessibility: Challenges and opportunities, Mphasis Chair for Digital Accessibility and Inclusion, Organizational Behavior and Human Resources Management, Indian Institute of Management Bangalore, Bangalore, Karnataka, India <https://www.sciencedirect.com/science/article/pii/S0970389617301131>.
- [2] Book 'Ensuring Digital Accessibility through process and policy', Jonathan Lazar, Daniel Goldstein, Anne Taylor, ISBN: 978-0-12-800646-7, Elsevier.
- [3] Web accessibility standards and disability: developing critical perspectives on accessibility, Sarah Lewthwaite, Pages 1375-1383 | Received 02 Dec 2013, Accepted 20 Jun 2014, Published online: 10 Jul 2014 Sarah Lewthwaite.
- [4] Web accessibility and usability: limits and perspectives, Luigi Tateo, <http://ceur-ws.org/Vol-2817/paper35.pdf>.
- [5] Digital Accessibility Guide for Aging Population, International Journal of Computer Trends and Technology, 2022 by IJCTT Journal, Volume-70 Issue-2, <https://doi.org/10.14445/22312803/IJCTT-V70I2P102>.
- [6] Book 'Web accessibility, practical advice for the library and information professional', edited by Jenny Craven, ISBN 978-1-85604-625-1.
- [7] Web Accessibility and Universal design, A primer on Standards and Best practices for libraries, by Debra A. Riley-Huff, <https://journals.ala.org/index.php/ltr/article/view/4687/5574>.
- [8] ARIA framework coding library <https://www.w3.org/TR/html-aria/>.
- [9] SaILL: Saliency-Driven Injection of ARIA Landmarks.
- [10] What makes web data tables accessible? <https://dl.acm.org/doi/pdf/10.1145/3491102.3517469>.
- [11] HTML static code analysis on table tags <https://rules.sonarsource.com/html/RSPEC-1102>.
- [12] Book 'UX for the Web' by Marlie Ritter, Cara Winterbottom.
- [13] ARIA Slider role https://developer.mozilla.org/en-US/docs/Web/Accessibility/ARIA/Roles/slider_role.
- [14] Marion A. Hersh, Michael A. Johnson, Assistive Technology for Visually Impaired and blind people.
- [15] The Cost Range of WCAG Auditing <https://www.accessibility.works/blog/web-accessibility-ada-compliance-costs-budgeting-guide/>.