Mobile Web Accessibility Testing Methodology

Step 1: Identify what needs to be tested

Step 2: Test critical issues

Step 3: Test mobile-specific issues

Step 4: Test mobile assistive technology and feature support

Step 5: Test mobile and desktop relationship issues

Table of Contents

Mobile Web Testing Methodology

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

Identify what needs to be tested

What needs to be tested is dependent on the site type, variations of the page and chosen devices.

Identify devices

Recommended devices and browser combinations:

- iPhone, Safari
- iPad, Safari
- Android phone, Chrome

Identify the site type and variations of the page

- **Desktop web sites:** non-responsive sites that have only one site view or display, whether viewed on desktop or mobile or tablet device – a viewer would see the exact same thing;
- **Responsive web sites:** user views or experiences that change depending on the screen size or other feature as determined in break points by the developer – usually the mobile is simpler than the tablet than the desktop web site;
- **m.dot sites:** that have a particular display for mobile and tablet sites. The m.dot site must also be tested against the entirety of WCAG2, **in addition** to the standard www version of the site.

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

See below under Responsive web site testing – all versions of the site on the various devices will be required for comprehensive testing. The user experience and accessibility can vary greatly between these differentiators, never apply results from one situation to another.

Desktop web site testing

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

Recommended testing:

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

m.dot site testing

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

Recommended testing:

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

Responsive web site testing

It is important that **each variation of the page is tested** and that **all functionality is available on all variations of the page**. The testing methods for responsive web site testing are dependent on whether there variations of the page.

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

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

Recommended testing:

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

Test critical issues

Hover traps

Applies to: Touch users

Content must be able to be dismissed if activated on touch (often these are actionable items that are activated on mouse hover on a desktop). For more information see WCAG2.1 SC 2.5.1: Pointer Gestures.

Touch traps

Applies to: Touch users

User must always be able to scroll / swipe to move up and down the page

Screen reader swipe traps

Applies to: Screen reader users

Screen reader users must always be able to activate an item on the current page or move back to the previous page.

Onscreen keyboard traps

Applies to: Onscreen keyboard users

Onscreen keyboard must be able to be dismissed.

Zoom traps

Applies to: Touch users

Do not replace the entirety of the page with a feature that overrides standard mobile functions such as swiping and scrolling.

Text-to-speech traps

Applies to: Screen reader users

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

Swipe traps

Applies to: Screen reader users or other assistive technology users which capture the swipe

Any swipe gesture that is supplanted by a screen reader gesture must have an alternate gesture. For example, if you require your user to swipe right to complete a purchase, when the screen reader is on, the swipe right gesture moves you to the next focusable item and

doesn't complete the purchase. You must be able to perform the same action, by using a link, an up or down swipe, or some other gesture.

Headset traps

Applies to: Screen reader users, Headset users

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

Exit traps

Applies to: All users

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

Layer trap

Applies to: All users (but mostly encountered by screen reader users)

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

Test mobile-specific issues

Alternatives

Alternatives are provided for non-web mobile functionality that is mandatory (for example, recording a specific gesture by the camera, before functionality appears).

Important functionality and information is available in the Reader or Simplified view.

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

Audio cues have an equivalent visual cue

Horizontal or vertical swiping support touch gestures as a fallback (for more information see WCAG2.1 SC 2.5.1: Pointer Gestures).

Horizontal or vertical dragging support touch gestures as a fallback (for more information see WCAG2.1 SC 2.5.1: Pointer Gestures).

Toggle and slider elements support touch gestures as a fallback (for more information see WCAG2.1 SC 2.5.1: Pointer Gestures).

Active swipe elements support both horizontal scrolling and swipe gestures.

Actionable elements are triggered only on removal of touch (ON TOUCH START and ON KEY DOWN have not been used) (for more information see WCAG2.1 SC 2.5.2: Pointer Cancellation).

Display

Do not present new content on hover over actionable element (for example, do not have a top-level menu item that displays sub-items on hover, but also when tapped opens a new page). For more information, see WCAG2.1 SC 1.4.13: Content on Hover or Focus.

Size of touch targets is at least 44 by 44 CSS pixels (approximately 7 to 10 millimeters). For more information see WCAG2.1 SC 2.5.5: Target Size.

Touch targets have sufficient inactive space between them (Inactive space of at least 10 pixels should be provided around active elements).

Horizontal scroll bars do not appear at all when the page is resized. For more information, see WCAG2.1 SC 1.4.10: Reflow.

Pinch zoom is operable, unless an accessible font resizing feature has been included in the web site that allows the user to increase the size of content at least two times the size of the standard font size. For more information see WCAG2.1 SC 1.4.4: Resize text.

The system can be used in portrait mode (for more information see WCAG2.1 SC 1.3.4: Orientation).

The system can be used in landscape mode (for more information see WCAG2.1 SC 1.3.4: Orientation).

Actionable items

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

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

Functionality that can be operated by device motion or user motion can also be operated by user interface components, and responding to the motion can be disabled to prevent accidental actuation, except for certain situations. For more information see WCAG2.1 SC 2.5.4: Motion Actuation.

Infinite scrolling has not been used

Links

Link text should have a minimum color contrast ratio of 3.0:1 when compared with the surrounding body text. For more information see WCAG2.1 SC 1.4.11: Non-text Contrast.

Color alone should not be used to indicate links (if not underlined). A secondary method, such as underlines should be used, in addition to color. For more information see WCAG2.1 SC 1.4.1: Use of Color.

Navigational aids

Arrows and Next and Previous buttons have been used to indicate swipe or scroll areas (for more information see WCAG2.1 SC 2.5.1: Pointer Gestures).

Navigational aids such as back buttons, breadcrumbs, next and previous buttons are provided.

ARIA document landmarks have been used to appropriately describe document structure.

Audio and Video

All video and audio have an accessible transcript.

Forms

Field labels for all fields are positioned adjacent to the input field

The following HTML5 input type are used appropriately: EMAIL, TEL, DATE, DATETIME, MONTH, SEARCH

The user should be able to dismiss on-screen keyboards that display when the user focuses on an input field. Additionally, any pop-up dialog or warning must be able to be dismissed. For more information, see WCAG2.1 SC 1.4.13: Content on Hover or Focus.

Test mobile assistive technology and feature support

All actionable items can be accessed and activated by the following assistive technologies (or when the following feature is enabled):

- VoiceOver (iOS)
- Keyboard (iOS)
- Keyboard and switch (iOS)
- Zoom (iOS)
- Invert colors (iOS)
- Grayscale (iOS)
- Reader view (iOS)
- TalkBack (Android)
- Keyboard (Android
- Keyboard and switch (Android)
- Magnification (Android)
- Invert colors (Android)
- Grayscale (Android)
- Increase text size (Android)
- Simplified view (Android)

All important content can be accessed by the following assistive technologies (or when the following feature is enabled):

- VoiceOver (iOS)
- Keyboard (iOS)
- Keyboard and switch (iOS)
- Zoom (iOS)
- Invert colors (iOS)
- Grayscale (iOS)
- Reader view (iOS)
- TalkBack (Android)
- Keyboard (Android
- Keyboard and switch (Android)
- Magnification (Android)
- Invert colors (Android)
- Grayscale (Android)

- Increase text size (Android)
- Simplified view (Android)

Test mobile and desktop relationship issues

Item labelling between the mobile and main site is consistent

Links between mobile and full version of the web site have been provided

Users are not restricted to a particular version dependent on device (i.e. cannot use mobile version on desktop and vice versa)