**WCAG 2.1 New Success Criteria**

WCAG 2.1 introduced **17 new success criteria** to address evolving technologies, especially touch interfaces, small screens, motion sensors, and other modern input methods.

Below is a deep focus on the most impactful ones in mobile and responsive design:

**1. Mobile Accessibility**

**1.4.10 Reflow (Level AA)**

**Definition**: Content must support **reflow** without loss of information or functionality, and **no horizontal scrolling** is required at 320px width (or 256px if vertical writing).

**Detailed Example:**

A public transport app shows departure times and fare options in a grid. Originally, on mobile:

* Users needed to scroll **horizontally** to read full content.
* Font sizes were small and columns overlapped.

**WCAG-compliant Fix**:

* Layout changes to a **single-column responsive design** at <768px.
* Flexbox is used for wrapping.
* Text wraps gracefully, no horizontal scroll needed.

@media (max-width: 768px) {

.grid-container {

flex-direction: column;

}

}

**Common Failures:**

* Fixed pixel layouts (e.g., width: 1024px).
* PDF documents embedded in full width without mobile responsiveness.

**Case Study – Government Tax Filing Site:**

**Issue**: The site forced horizontal scrolling and pinch-zoom for mobile users to submit forms.

**Solution**:

* Reflow implemented using CSS Grid + media queries.
* Inputs and buttons became stackable and resizable.

**Result**:

* 27% decrease in form abandonment on mobile.
* Increased usage among older users and those using screen magnifiers.

**2.5.5 Target Size (Level AAA)**

**Definition**: Interactive targets (e.g., buttons, links) must be **at least 44×44 CSS pixels**, unless there's sufficient spacing.

**Example:**

A university’s mobile menu includes links with 20px height. Students report frequent mistaps.

**Fix**:

* Increase tap targets to 44px.
* Add padding: 12px 16px on buttons and set line-height: 1.5.

**Mistakes:**

* Icon-only buttons with no accessible name or small touch area.
* Clustered form fields or action buttons that overlap on smaller screens.

**2. Input Modalities**

**2.5.1 Pointer Gestures (Level A)**

**Definition**: All functions that use multipoint or path-based gestures must have **simple alternatives** (e.g., tap or click).

**Example:**

A photo app uses **pinch-to-zoom** only. This excludes users with motor impairments or screen readers.

**Fix**:

* Add zoom-in (+) and zoom-out (–) buttons as alternatives.
* Ensure these are accessible via keyboard and screen readers.

**2.5.2 Pointer Cancellation (Level A)**

**Definition**: Function should not trigger just from pointer **down** event. Allow **abort** or **undo**.

**Example:**

A “Delete” button triggers immediately on finger press (mousedown), causing unintended deletions.

**Fix**:

* Require mouseup to confirm intent.
* Add confirmation dialog (e.g., “Are you sure you want to delete?”).

**2.5.4 Motion Actuation (Level A)**

**Definition**: Functions triggered by device motion (e.g., shaking) must also be available via **standard UI**.

**Example:**

A fitness app allows users to **shake the phone to reset workout timer**. This is inaccessible to users with motor impairments.

**Fix**:

* Add a visible "Reset" button.
* Include setting to **disable motion-based triggers**.

**Case Study – Fintech App**

**Problem**: A financial app allowed gesture-only navigation (swipe to confirm payment), which was unusable for screen reader or switch control users.

**Solution**:

* Replaced swipe-only with "Confirm" and "Cancel" buttons.
* Maintained gesture support but made it optional.

**Result**:

* App passed Android Accessibility Test Suite.
* Positive reviews from users with Parkinson’s and tremors.

**3. Orientation and Zoom/Responsiveness**

**1.3.4 Orientation (Level AA)**

**Definition**: Content must **not be restricted to one display orientation** (portrait or landscape), unless essential.

**Example:**

A museum app shows a map **only in landscape**.

**Fix**:

* Allow map viewing in **portrait mode**, optionally zoomable.
* Use orientation media queries to adjust design.

@media (orientation: portrait) {

.map {

height: 300px;

overflow: scroll;

}

}

**Bad Practice:**

* Locking orientation via JavaScript or meta tags unless for apps like digital levels or games.

**1.4.4 Resize Text (Level AA)**

**Definition**: Users must be able to resize text **up to 200%** without assistive tech and **without loss of content or functionality**.

**Example:**

On a banking website, enlarging text breaks the UI:

* Input fields overlap.
* Buttons disappear off-screen.

**Fix**:

* Use relative font sizes (em, rem).
* Avoid fixed heights on containers.
* Apply fluid layout techniques.

**Case Study – Retail App**

**Issue**: Zooming in on product descriptions caused text to overflow off-screen with no horizontal scroll.

**Fix**:

* Used vw, em units.
* Container widths set with max-width: 100%.

**Impact**:

* Improved readability for low-vision users.
* Reduced cart abandonment from 18% to 12%.

**Summary Table**

| **Success Criterion** | **Description** | **Level** | **Key Devices** |
| --- | --- | --- | --- |
| 1.4.10 Reflow | No horizontal scrolling at 320px | AA | Phones, Tablets |
| 2.5.1 Pointer Gestures | Multipoint gestures need simple alternatives | A | Touchscreens |
| 2.5.4 Motion Actuation | Motion-triggered features must have UI backup | A | Phones with gyroscope |
| 1.3.4 Orientation | Support both portrait & landscape | AA | All screens |
| 2.5.5 Target Size | Touch targets ≥ 44×44 px | AAA | Phones, Tablets |
| 1.4.4 Resize Text | Support 200% zoom | AA | All devices |