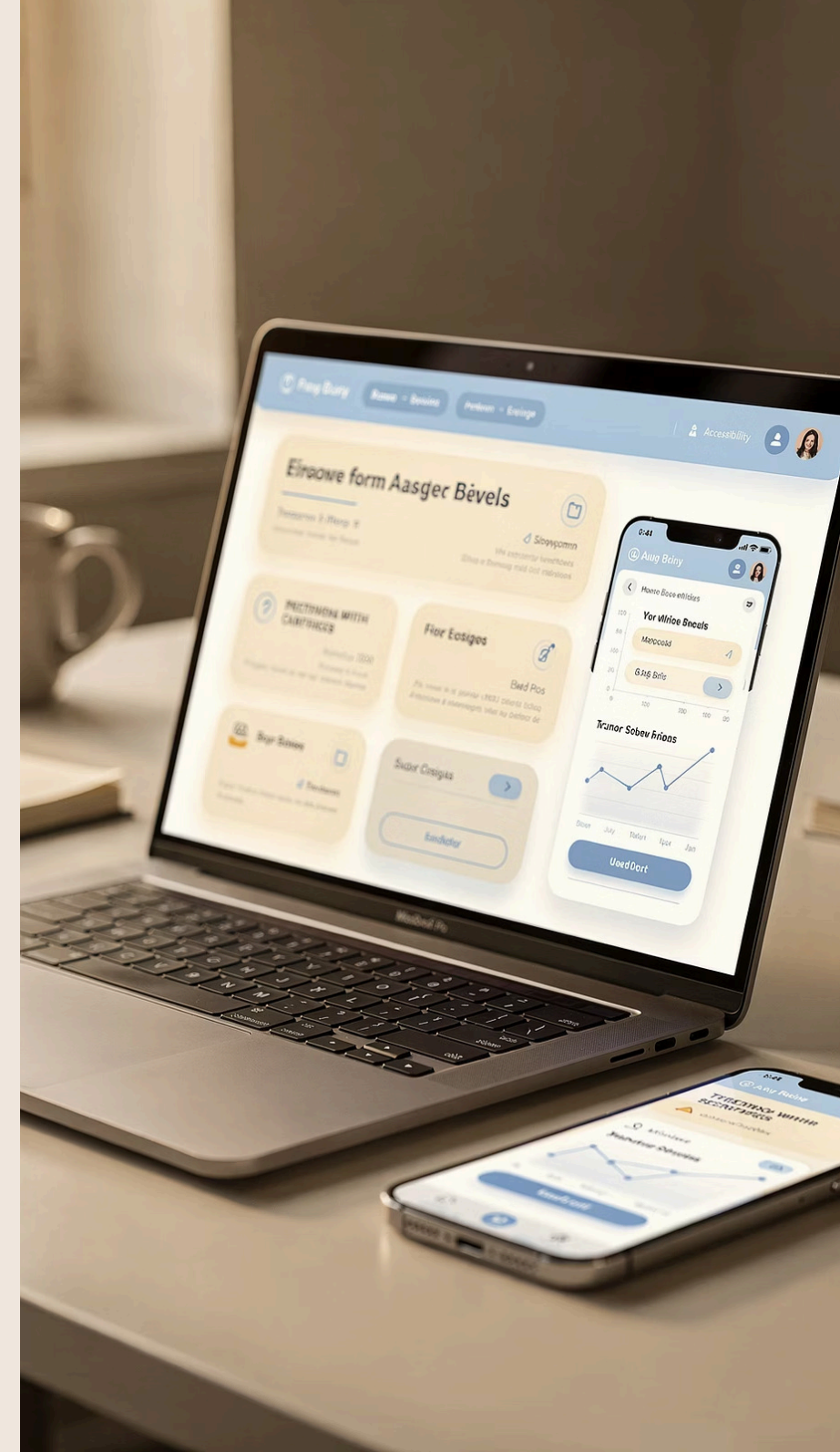


wA11y Accessible Component Vetting Guide

This guide defines a repeatable audit protocol for evaluating third-party UI components before production integration.

It is used to:

- Identify accessibility risks early
- Estimate remediation effort
- Support adoption decisions with evidence



The Three-Layer Vetting Process

The Shift-Left philosophy prioritizes identifying accessibility blockers before they enter the codebase. This systematic approach moves beyond marketing claims to verify real-world component accessibility through three distinct validation layers.

Layer 1: Documentation Review

Analyze VPAT or ACR Files

Begin with a thorough examination of Voluntary Product Accessibility Templates (VPATs) or Accessibility Conformance Reports (ACRs). Look for detailed remarks that explain conformance levels.

Critical Red Flag: A VPAT claiming "100% Support" for complex widgets without granular documentation indicates insufficient testing or transparency issues.

Layer 2: Automated Smoke Test

Diagnostic Scanning

Run automated accessibility audits using Axe, Lighthouse, or equivalent tools on the vendor's live demo environment to establish a baseline.

Automated tools detect structural violations but do not validate usability.

Key Detection Points

- Color contrast ratios below WCAG 2.1 AA standards (4.5:1 for text)
- Missing accessible names
- Landmark and role misuse
- Duplicate or missing ID attributes that break ARIA relationships
- Missing alternative text for meaningful images
- Form elements without associated labels

Layer 3: Manual Stress Test

Automated tools detect only 30-40% of accessibility issues. Manual testing validates interaction patterns, focus management, and screen reader compatibility that cannot be verified programmatically.

This layer targets high-priority gaps where automation fails: keyboard traps, dynamic content announcements, and context-dependent ARIA states.

Verify focus trapping in modals and overlays using the native `inert` attribute or specialized libraries; confirm that focus is programmatically restored to the trigger element upon dismissal.

Keyboard & Focus Navigation

This critical manual audit layer verifies essential interaction patterns that automated tools often miss. It ensures a robust and intuitive experience for users relying on keyboard-only navigation or assistive technologies.

01

Keyboard-Only Interaction

Thoroughly navigate all interactive elements using Tab, Shift+Tab, Enter, Space, and arrow keys. Confirm every interactive component is reachable and activatable without a mouse.

03

Screen Reader Compatibility

Test with primary screen reader combinations (e.g., NVDA + Chrome on Windows, VoiceOver + Safari on macOS) to validate correct announcements of roles, states, and names for all components.

02

Focus Indicator & Traps

Verify a clear, visible focus indicator appears on each interactive element. Crucially, confirm that no keyboard traps exist, preventing users from moving focus away from a component once entered.

04

Visual & System Preference Checks

Assess the component's behavior at 200% and 400% zoom levels. Activate prefers-reduced-motion to ensure animations are suppressed, and test high contrast mode (Windows) for visual integrity. All testing must occur in a production build environment.

Focus Indicator Standards

Effective keyboard navigation relies on clear and consistent focus indicators. These visual cues must adhere to specific standards to ensure usability for all:

Visual Thickness

Must be at least 3px visible thickness OR have a 2px outline with a 1px offset for optimal visibility.

Contrast Requirements

Maintain a minimum 3:1 contrast ratio against adjacent colors to ensure legibility.

Background Resilience

Remain clearly visible and distinct on all background color variants used within the component.

Color Independence

The focus state must not rely on color alone; additional visual cues are necessary.

Persistent Display

Cannot be removed using outline: none without a clearly visible and functionally equivalent replacement.

Technical Audit: Keyboard & Focus

The No-Mouse Challenge

Every interactive element must be operable via keyboard alone. Verify that Space or Enter triggers all buttons, links, and custom controls. Test tab order follows logical reading sequence.

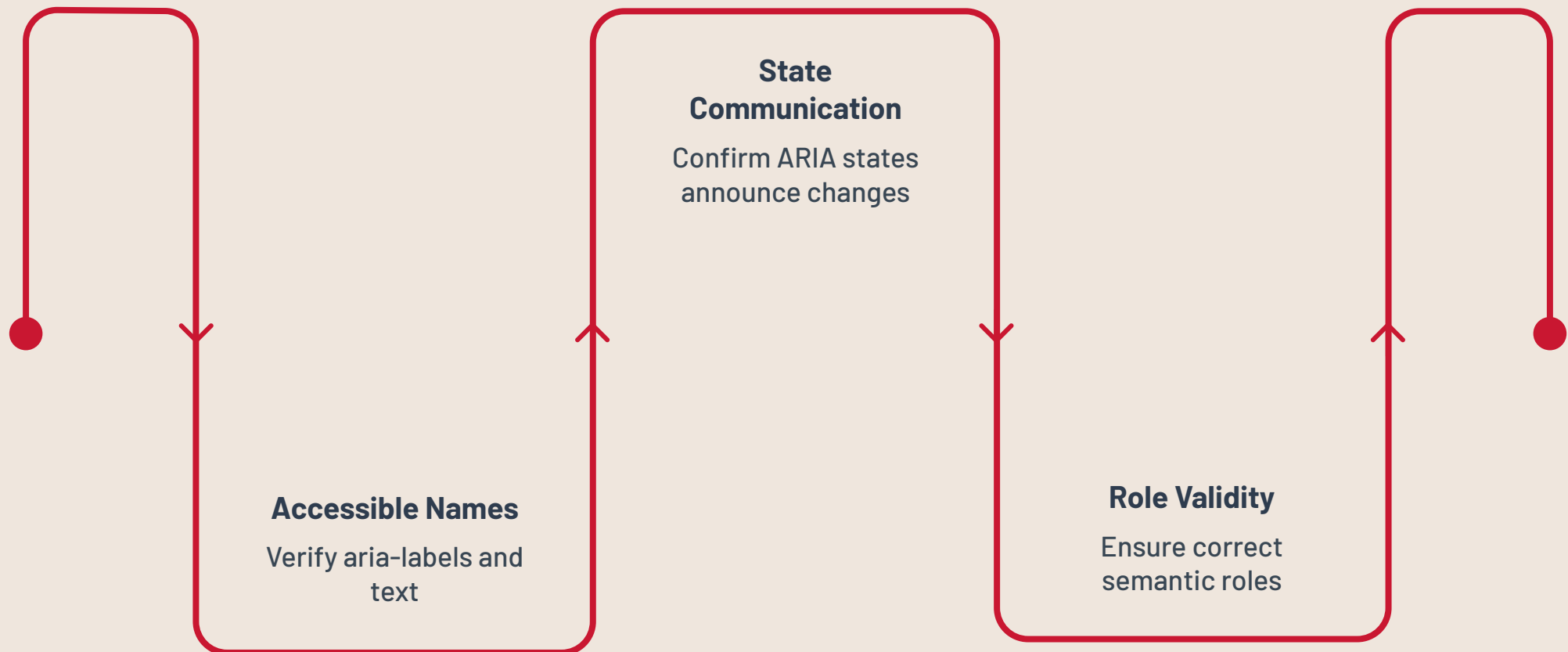
Focus Management

Verify focus trapping in modals using a library like `focus-trap` or native `inert` attribute; ensure focus returns to the initiating trigger upon dismissal.

Visual Persistence

Focus indicators must meet wA11y standard of 3px minimum thickness and remain visible at all times. Check for CSS properties like `overflow: hidden` on parent containers that clip focus rings.

Technical Audit: Screen Reader & Semantics



Screen reader compatibility requires semantic HTML and proper ARIA implementation for dynamic interfaces.

Accessible Names: Buttons without visible text must use `aria-label` or `aria-labelledby` to provide context.

State Communication: Validate that custom widgets utilize the correct ARIA Design Pattern (e.g., Tabbed interfaces must use `role='tablist'`, `tab`, and `tabpanel` with associated `aria-controls`)

Role Validity: Prefer native HTML semantics. Custom roles like `role="combobox"` must include all required child roles per ARIA specification.

Minimum Validation Requirements

Each component must:

- Use native HTML elements whenever possible.
- Avoid ARIA roles that override native semantics.
- Provide a programmatic name.
- Expose state changes (expanded, checked, selected, pressed).
- Maintain correct reading order in DOM.



If ARIA is required, the pattern must match the WAI-ARIA Authoring Practices specification

Design & Flexibility Audit

Resizing & Reflow

Test component stability at 200% text zoom. Verify responsive behavior supports reflow at 400% zoom without introducing horizontal scrolling, which violates WCAG 2.1 Success Criterion 1.4.10.

Customization Safety

Components must accept wA11y semantic design tokens without breaking internal logic. Assess whether custom styling can be applied to borders, backgrounds, and typography without compromising functionality.

Reflow & Zoom Requirements

Components **must**:

Reflow at 200% zoom without horizontal scroll (WCAG 1.4.10).


Remain usable at 400% zoom at a 1280px viewport width.

Preserve reading order.

Maintain visible focus.

Avoid content clipping.

Support responsive breakpoints without structural collapse.

 **Before You Buy:** Calculate the cost of accessibility remediation versus selecting an accessible alternative. Post-integration fixes typically cost 5-10x more than upfront vetting.

Issue Severity Definitions

1

Critical

- Blocks keyboard access or screen reader interaction
- Causes complete loss of functionality
- Violates WCAG Level A requirements
- **Remediation required before adoption**

2

Major

- Creates significant usability barriers
- Requires workarounds for assistive technology users
- Violates WCAG Level AA requirements
- **Requires documented remediation plan**

3

Minor

- Does not block use or AT access
- Creates friction, but not complete failure
- **May be scheduled for backlog resolution**

Vetting Verdicts: The Decision Matrix



PASS

Zero critical failures identified across all three audit layers. Minor defects are documented, reproducible, and addressable through component props or configuration without code modification.



REMEDiate

Accessible core logic exists, but implementation requires custom wrappers, ARIA overrides, or CSS adjustments to meet wA11y standards. Remediation effort must be estimated and documented.



FAIL

Structural failures that cannot be resolved without complete rebuild. Examples include div-based buttons without keyboard support, missing semantic structure, or focus management that cannot be overridden.

Comprehensive Accessibility Audit Checklist

This checklist provides a rapid assessment framework for evaluating component accessibility across key domains:

Structure

- Native HTML element used
- Accessible name present (aria-label, aria-labelledby)
- Valid role/state for custom components

Keyboard

- Fully operable via keyboard alone
- No keyboard focus traps
- Logical tab order for navigation
- Visible and consistent focus indicator

Screen Reader

- Correct content announcement
- State changes announced accurately
- No redundant or conflicting ARIA roles

Visual

- 3:1 minimum focus indicator contrast
- 4.5:1 minimum text contrast
- 200% text reflow support without content loss

Motion

- Honors prefers-reduced-motion setting

Decision

- PASS
- REMEDIATE
- FAIL

Audit Scorecard & Next Steps

Audit Category	Score	Critical	Major	Minor
Keyboard Navigation	__ / 5	--	--	--
Screen Reader Support	__ / 5	--	--	--
Visual & Design Flexibility	__ / 5	--	--	--
Documentation Quality	__ / 5	--	--	--
Total Score	__ / 20	--	--	--

Components scoring below 15/20 or with any critical failures require executive review before procurement. Reference the wA11y Governance Workflow for final approval procedures and escalation paths.