



Digital Accessibility Audit Checklist: A Step-by-Step Review



Define a structured, evidence-based evaluation against WCAG 2.2 standards



Role-specific tasks for designers, developers, content publishers, team leads



Checklist turns broad requirements into executable review steps



Course covers: prepare, inspect, report, remediate, verify, sustain



Outcomes: prioritized findings, shared language, clear remediation roadmap



Why Unstructured Audits Fail



Unstructured Audit

Issues missed. Barriers remain.



Checkout

user123

.....

Shipping info

123 Main St

Apt 4B

???

Go



Structured Audit

Issues found. Barriers removed.



Checkout

Username

user123

Password

.....

Shipping address

Address line 1

123 Main St

Address line 2

Apt 4B

Continue



- Inconsistent coverage and subjective judgments miss critical barriers



- Untracked regressions allow accessibility issues to return silently



- Automated-only audits catch only 30-57% of issues; W3C requires human review



- Audit before releases, after major changes, and on a regular cadence

The WCAG 2.2 Baseline and Common Conformance Targets

- WCAG 2.2 has 86 active success criteria under four principles
- Level AA combines 55 A/AA criteria; the common legal target
- Six new A/AA criteria added: focus, dragging, target size, help, redundant entry, authentication
- Update 2.1-era checklists; drop obsolete 4.1.1 Parsing

FOUR PRINCIPLES OF WCAG



Perceivable



Operable



Understandable



Robust



SIX NEW A/AA CRITERIA IN WCAG 2.2



Focus



Dragging



Target Size



Help



Redundant Entry



Authentication

Roles and Responsibilities Across the Audit

Designers

contrast, focus indicators, target size, interaction patterns



Developers

semantic structure, ARIA, keyboard behavior, forms



Content

alt text, headings, captions, transcripts, documents



Team leads

accountability, scheduling, preventing dropped work



- Clear ownership prevents “everyone's job” from becoming no one's



Scoping the Audit and Selecting Representative Samples



Choose representative pages, flows, and templates, not URL counts



Define scope: templates, embeds, mobile views, documents, media



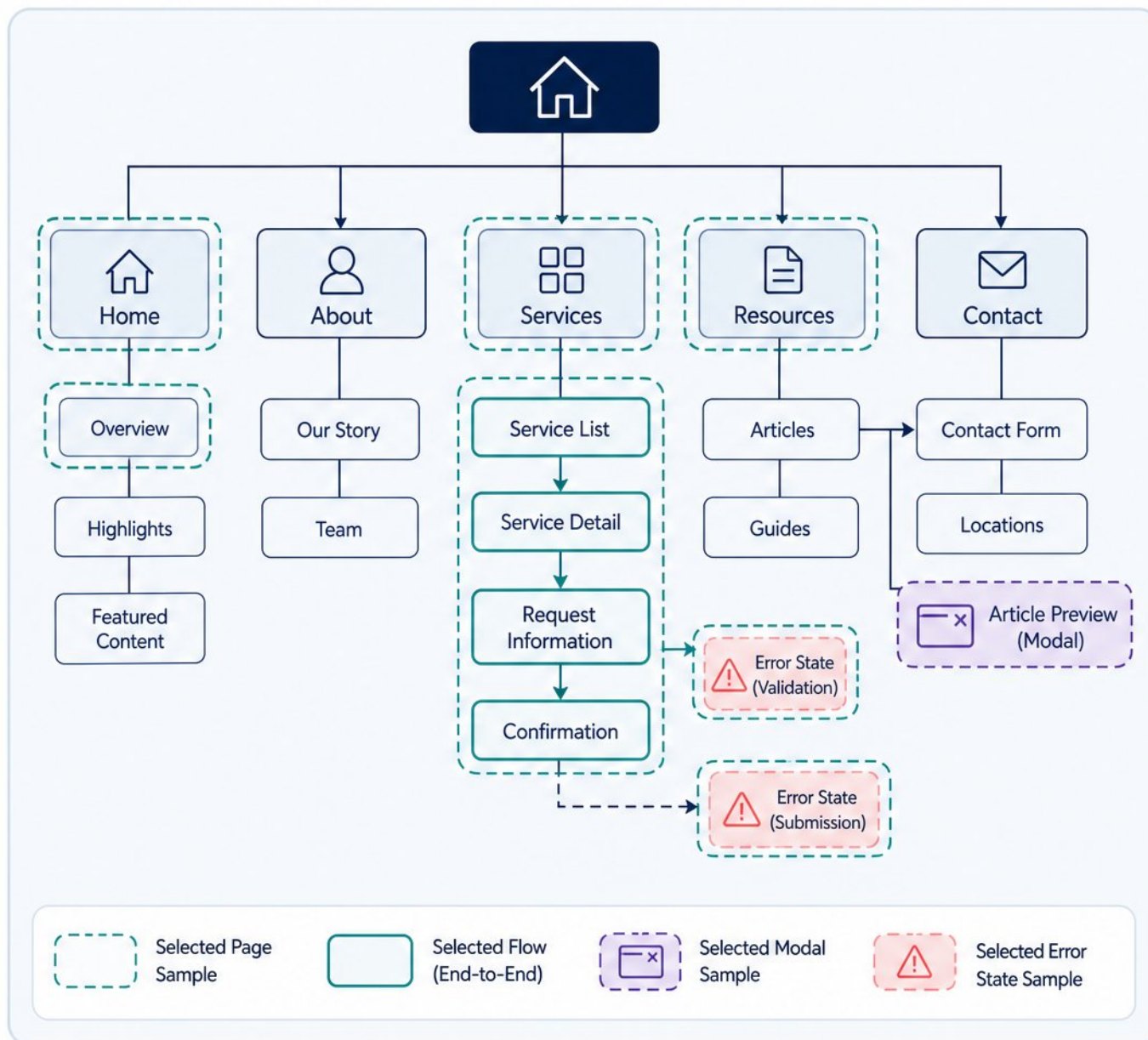
Document assumptions and exclusions before starting



Include full multi-step flows and error states



Balance depth with resources; record sample reasoning for defensibility



Automated Checks: The First Pass



- Detect missing alt text, contrast failures, heading order issues, labels, and invalid ARIA



- Run in browsers, code environments, and CI/CD pipelines



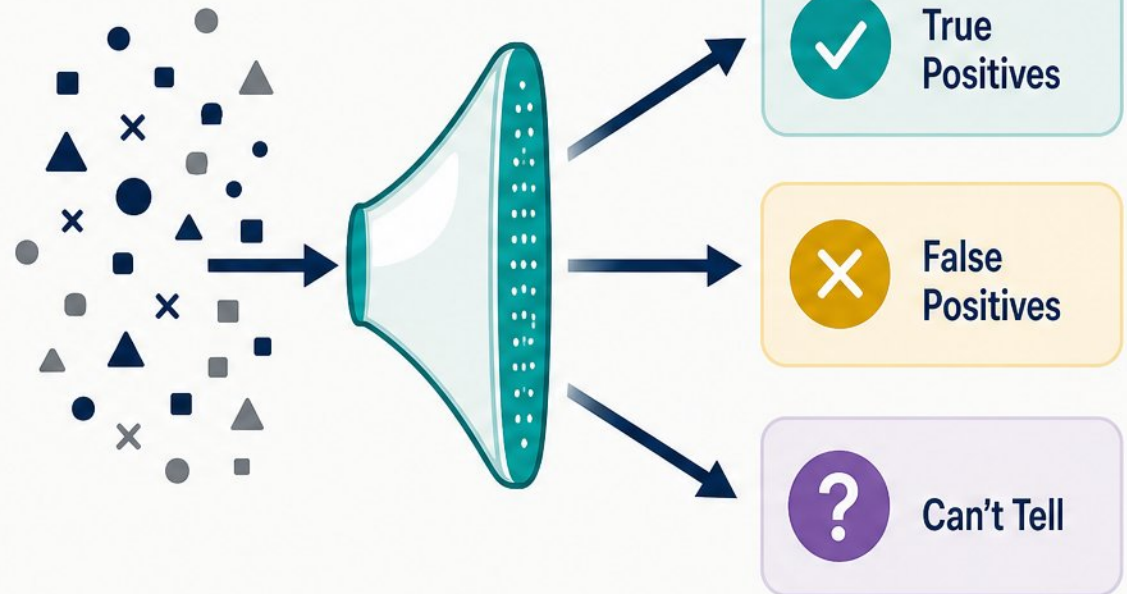
- Filter false positives; triage “incomplete” results—“can’t tell” is not a pass



- Cover only 30–57% of issues by volume—not the final verdict



- Clear high-volume defects cheaply; reserve humans for judgment-based checks





Keyboard and Focus Manual Checks



- Tab through every page without a mouse; verify logical order, no traps, and complete task flows.



- Check skip links, visible focus indicators, and that sticky headers or overlays do not obscure focused elements.



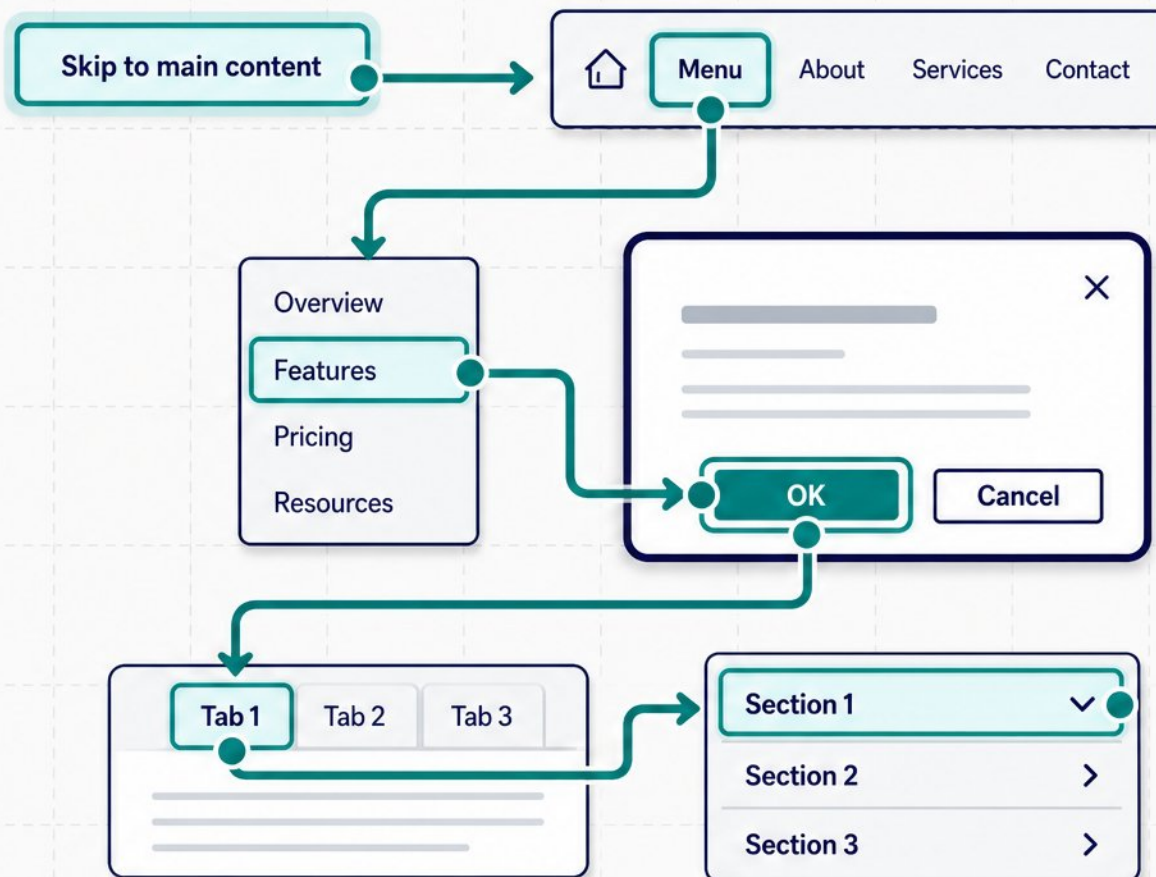
- Test modal dialogs, menus, tabs, accordions, and other custom widgets for correct focus management.



- Verify keyboard alternatives for drag-and-drop and pointer-only interactions.



- Test with operating system accessibility settings enabled, such as zoom and contrast modes.



Screen Reader and Content Manual Checks



- Test with NVDA, VoiceOver, TalkBack: verify names, roles, states, order



- Headings, landmarks, links, forms, dynamic announcements meaningful



- Alt text conveys purpose, not just presence

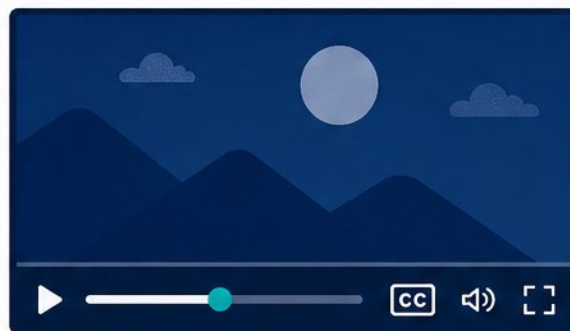


- Captions, transcripts, audio descriptions for media



- Combine keyboard and screen reader testing

News Article



SCREEN READER ANNOUNCEMENT

“Heading level 1 News Article link
main landmark
video
play button
Captions button
text ...”



Visual, Cognitive, and Responsive Checks



- Manually verify contrast on images, gradients, and overlays



- Test text spacing, reflow at 320px width, and zoom at 200%/400%



- Check for content clipping or loss during reflow and zoom



- Review instructions, errors, and navigation for clarity



- Ensure no color-only cues and targets are at least 24x24 px





Recording Findings and Assigning Severity



- Write reproducible findings: title, location, steps, expected vs. actual result, evidence



- Assign severity: user impact, frequency, workaround availability



- Link each finding to WCAG success criteria and affected role



- Use consistent templates or issue trackers for comparability



- Separate confirmed findings from tool candidates with documented reasoning



Finding Evidence Card



Title

Link text is not descriptive



Location

Main navigation menu



Steps to Reproduce

- Navigate to the homepage.
- Use keyboard to move to the main navigation.
- Focus on the "Learn more" link.



Expected Result

The link text describes the destination or purpose.

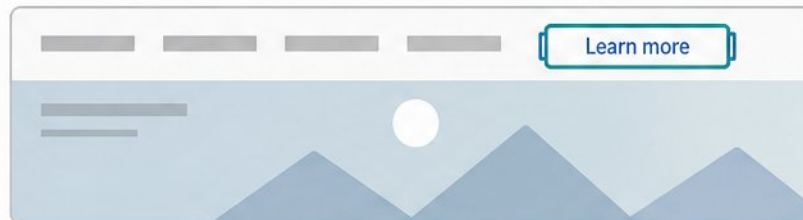


Actual Result

The link text says "Learn more" which does not describe the destination.



Evidence



WCAG Success Criteria

Link Purpose (In Context)



Affected Role

Keyboard and screen reader users



Prioritizing and Planning Remediation



- Prioritize: user impact, legal risk, frequency, effort



- Group issues by component or template for leverage points



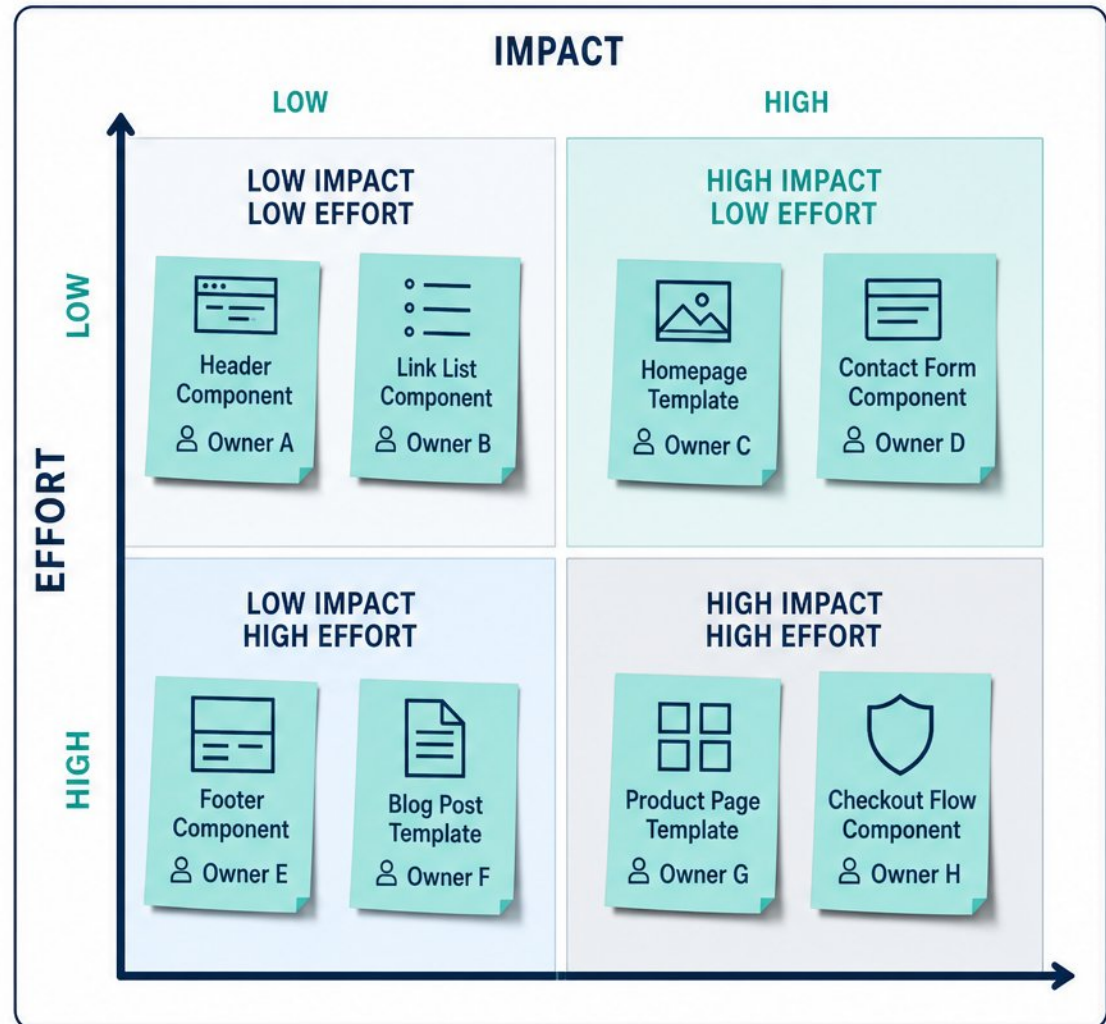
- Quick wins: high impact, low effort — fix immediately



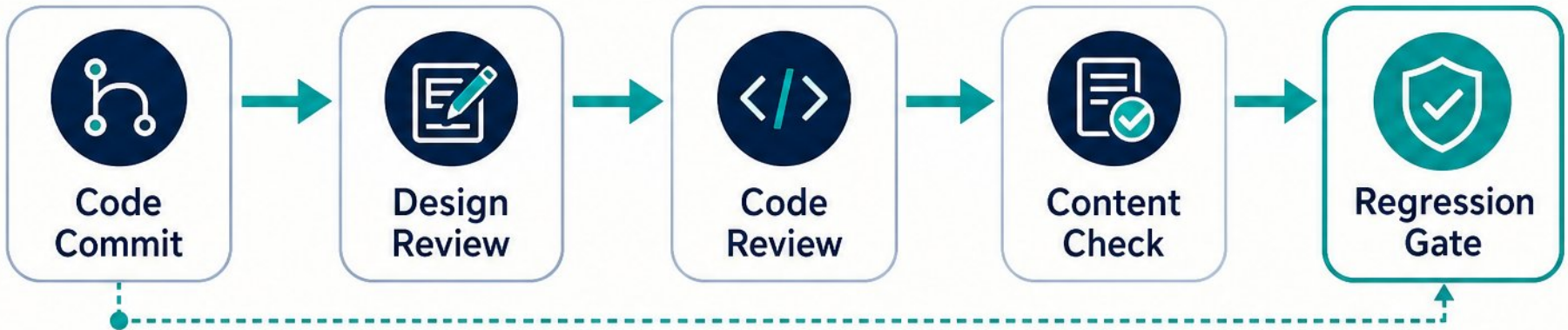
- Assign owners, deadlines, and review checkpoints



- Use an impact-effort matrix for a defensible order



Re-Testing and Regression Prevention



- Re-test fixed items against original reproduction steps



- Embed checks in design reviews, code reviews, and CI/CD



- Baseline accepted violations; gate only on new regressions



- Run scheduled full audits to catch untouched-page drift



- Maintain a living checklist as components and standards evolve

Adoption, Metrics, and Sustainability



- **Start small:** pilot the checklist on one product, template, or content type



- **Track what matters:** open high-severity findings, time to fix, repeat issues, and coverage



- **Use trend-based metrics** (health score, new/reopened issues) to demonstrate progress



- **Embed checks** into design reviews, code reviews, content workflows, and CI/CD



- **Build a shared vocabulary** and a long-term roadmap that integrates accessibility into workflows



Resources and Next Steps

- ✓ Start with the W3C WCAG 2.2 Quick Reference and evaluation methodology
- ✓ Continue learning through training and testing with disabled users
- ✓ Build a concrete action plan: choose a standard, audit a pilot, record findings, retest
- ✓ Develop internal expertise to reduce reliance on external consultants
- ✓ Next step: run your checklist against a real page or flow



Next step: run your checklist against a real page or flow

PersonWise.

PersonWise • AI digital-human course platform



Scan the code or click anywhere to watch this course live, with voice Q&A

personwise.ai/t/aeae411e/public