

B2B Website Audit Workbook: reproducible checks instead of guesswork



AI concept illustration. Editorial content and working models: Patrick Wings.

This workbook turns “the website is not working well” into reproducible findings. You test selected visitor tasks, record actual behaviour and separate defects from assumptions about their impact. Every finding needs a page, starting state, steps and expected outcome. The company Wartungsblick and its demo journey are explicitly fictional working examples that illustrate the process, not measured results. This is a bounded marketing and usability review, not a complete accessibility assessment or a legal compliance determination.

Edition 17 October 2026 · English

Implementation support: [request a strategy call](#)

HOW TO USE THIS RESOURCE

For B2B marketing teams, website owners and delivery partners who need specific improvement assignments with verifiable acceptance.

Your practical outputs

- A documented test path from entry to confirmed enquiry.
- Reproducible findings for content, interaction and technical access.
- An action plan with impact, ownership and retesting.
- A defensible release decision with remaining uncertainties made explicit.

Contents

1. Scope the review around important visitor tasks	3
2. Specify starting conditions and evidence format	3
3. Test the service promise through a concrete task	4
4. Check navigation and orientation on smaller screens	4
5. Run a repeatable keyboard and form check	4
6. Check images, contrast and media deliberately	5
7. Test the complete enquiry rather than the click	5
8. Assess technical access separately	6
9. Investigate loading problems under comparable conditions	6
10. Prioritise by task blockage and strength of evidence	6
11. Run an audit workshop with a clear handoff	7
12. Release with retesting and known limitations	7
Editable finding and acceptance record	8
Application questions	9
Sources and context	9

1. Scope the review around important visitor tasks

Choose three tasks a suitable visitor should actually complete: understand a service, find relevant evidence and initiate a next step. Record the starting page and end state. Avoid random page inspections while the central enquiry journey remains undefined. A full inventory can follow later; the first round needs a testable focus.

The fictional Wartungsblick sells B2B maintenance-planning software. Its first test path starts on a service page and ends with a successfully confirmed demo request. A second path leads to implementation information and a third to a guide download. The team describes each end state without promising an improved conversion rate: visitors must be able to finish the task reliably.

2. Specify starting conditions and evidence format

Record browser, viewport width, login state, language and consent state. A session with saved form entries is not equivalent to a first visit. Use clearly identifiable test data and coordinate test enquiries with the receiving team. Avoid unnecessary personal information in screenshots or openly shared issue lists.

A useful finding contains the exact URL, numbered steps, expected and observed behaviour, and supporting evidence. Add whether the issue is repeatable or occurred only once. If something could not be checked, mark it “not tested” rather than “passed”. Implementation partners can then reproduce the issue without first reconstructing your entire session.

Test path	Starting state	Evidence and expected outcome
Mobile demo	Logged out, narrow viewport, empty inputs	Video or screenshots; confirmation after successful processing
Understand service	Direct service-page entry, chosen language	Test-person answers; audience and help are understandable
Download	First visit with a defined consent state	File is accessible and fulfils the promise
Keyboard journey	Put mouse aside and reload page	Record sequence; task can be completed without a mouse

3. Test the service promise through a concrete task

Ask someone uninvolved to explain, after reading the introduction, who the offer is for, which problem it addresses and what happens next. Record their answers verbatim. Do not correct them during the task. Someone already familiar with your terminology finds different problems from a suitable first-time visitor.

In the fictional example, Wartungsblick initially displays only broad efficiency claims. The task requires understanding the difference between software, consulting and ongoing support. A revised promise passes only when those distinctions are apparent from the content. Your own satisfaction with new wording cannot replace repeating the test.

4. Check navigation and orientation on smaller screens

Start on a detail page rather than only the homepage. Find primary navigation, the current topic, the contact route and the way back to an overview. Repeat at a narrow viewport and with enlarged text. Record clipped wording, obscured controls and links whose destinations remain unclear before activation.

In the fictional case, mobile navigation requires two unclear symbols before the demo page becomes reachable. That is an observable journey, not proof of lost revenue. Set a precise acceptance condition: a test person finds the next step through understandable labels and can close the menu without losing their place.

5. Run a repeatable keyboard and form check

Put the mouse aside and start with the keyboard after reloading. Move through navigation, submenus, the form and dialogs. Record the sequence and whether your current position remains visible. Open and close relevant controls and check that work can continue afterwards. The W3C reference covers keyboard operation, focus and understandable form guidance among other topics; use it for deeper assessment, not an automatically earned approval badge.

Then submit an empty form, an invalid email address and finally a valid test entry. Check whether errors explain the affected input and preserve existing information. For Wartungsblick, an unexplained disabled button is a specific finding: record the starting state and inputs, then repeat the same conditions after the repair.

Test steps	Expected outcome	Evidence when failing
Tab to navigation and form	Position is apparent; relevant controls are reachable	Record order and missing control
Open and close a menu or dialog	Operation succeeds; work can sensibly continue	Record keystrokes and ending position
Submit an empty form	Missing required inputs are clearly identified	Screenshot and affected fields
Submit an invalid email	Understandable correction guidance without lost inputs	Record inputs and message
Submit a valid test entry	Success follows processing; recipient receives entry	Reconcile time, test identifier and received record

6. Check images, contrast and media deliberately

Review the page without its images and note which substantive information disappears. Important diagrams need an understandable explanation; decorative shapes do not need artificial keyword narratives. Use an appropriate contrast tool to inspect the colours actually rendered rather than only the original design. Consult the linked W3C reference for formal criteria.

For videos, test playback, available captions, understandable controls and a useful textual alternative for central information. In the fictional Wartungsblick case, one video contains the only implementation explanation. The finding is therefore not “more text would be nice” but “the process cannot be understood without the video”. Acceptance requires an alternative that covers the substantive information.

7. Test the complete enquiry rather than the click

A conversion task does not end when someone presses a button. Follow a labelled test entry to its actual destination: confirmation view, email, CRM record or agreed callback. Record duplicates, missing ownership and contradictory confirmations. A positive analytics event alone does not prove that an actionable contact reached the receiving team.

In the fictional example, Wartungsblick also checks whether a download request is incorrectly categorised as a demo request. The visitor’s expectation remains central. If a conversation is promised, the next step and responsible team must be clear. A technical failure needs an understandable response and a sensible retry or contact route.

8. Assess technical access separately

Check public destination pages while logged out and record redirects, errors and accidental sign-in barriers. Ask the technical owner to inspect response status, relevant indexing instructions and the rules applicable to each crawler. Google describes robots.txt as a way to manage crawl access; having such a file does not guarantee indexing. Assess actual access conditions together.

Also compare central statements in delivered content with what visitors see. Hidden wording or text embedded only in images makes independent inspection harder. For search and answer systems, technical availability, understandable content and actual selection are different questions. Record “technically inspectable” when supported rather than turning that observation into “will be cited by every language model”.

9. Investigate loading problems under comparable conditions

Repeat important tasks with an empty cache and documented conditions. Observe shifting content, blocked first actions or a form changing position while loading. One measurement from a fast office connection does not explain every visitor’s experience. Combine technical measurements with the actual task walkthrough.

In the fictional Wartungsblick case, an image loaded late shifts the demo button. The team records device, viewport and loading sequence, then repeats the check after the change. It makes no specific revenue claim. The first defensible improvement is that the control no longer moves unexpectedly during the same task and can be activated reliably.

10. Prioritise by task blockage and strength of evidence

Distinguish a blocked core task from friction and an unproven optimisation idea. Record the finding’s scope: does it affect a central journey, a specific viewport or an unusual edge case? Estimate effort after understanding the cause sufficiently. A visually small change can be technically demanding, and taste alone cannot establish a large impact.

The fictional team repairs a missing demo submission before rewriting the page title. Both may matter, but a demonstrated task blockage takes priority. Every action names an owner and uses the same test for acceptance. Unproven impact hypotheses enter a later comparison with their uncertainty preserved.

Finding class	Fictional finding	Decision and acceptance
Blockage	A valid demo request reaches nobody	Repair first; receive an identifiable test entry
Access problem	Form cannot be fully reached using the keyboard	Repair specifically; repeat the same keyboard path
Understanding gap	Visitor confuses download and demo	Clarify promise; repeat independent reading test
Hypothesis	Another CTA phrase might attract more interest	Plan a comparison; do not describe it as a proven defect

11. Run an audit workshop with a clear handoff

Prepare a working session with marketing, the technical owner and the person handling enquiries. One possible agenda allocates ten minutes to task selection, twenty to a shared walkthrough, twenty to sorting findings, fifteen to prioritisation and fifteen to acceptance planning. These timings are an adaptable facilitation suggestion.

Bring documented evidence rather than debating every design preference live. A finding enters implementation only when the expected and observed behaviour are distinguishable. If nobody can access the actual destination of an enquiry, assign a verification task instead of inventing a successful outcome. Finish the meeting with a small set of reproducible repairs.

12. Release with retesting and known limitations

After a change, repeat the original test and at least the immediately adjacent step. A repaired button may lead to the wrong destination; a working delivery process may now create duplicate records. Record date, version and retest outcome. Confirm only what was actually checked.

The first audit round is complete when core journeys are documented, significant blockages are repaired or explicitly accepted as unresolved, and actions have clear ownership. Give remaining questions a review date. Technical release is neither a complete accessibility determination nor evidence of better rankings or leads. It establishes a more reliable foundation for subsequently observing impact.

Editable finding and acceptance record

These completed starting points are explicitly fictional. Add your URL, starting conditions, actual evidence and owner for each row. “Not tested” is valid and must not be reinterpreted as “passed”.

[Open the editable worksheet in Excel / Google Sheets](#)

Test task	Fictional starting point and steps	Expected result / your evidence	Owner / retest
Understand service	Open service page directly; ask person to explain audience and help	Software and consulting are distinguishable / complete	Add your entry
Mobile navigation	Narrow viewport; open menu, find demo, close menu	Destination is understandably reachable / complete	Add your entry
Keyboard	Reload; navigate to submission without a mouse	Position apparent, no blockage / complete	Add your entry
Input errors	Submit empty form and invalid email	Useful error guidance, inputs retained / complete	Add your entry
Received enquiry	Send a clearly labelled valid test entry	Confirmation and actual receipt / complete	Add your entry
Image information	Review diagram without image; read alternative explanation	Essential meaning remains understandable / complete	Add your entry
Public access	Open direct destination URL while logged out	No accidental sign-in barrier / complete	Add your entry
Loading behaviour	Empty cache; observe first CTA during loading	No unexpected position change / complete	Add your entry
Download	Request guide and open the actual file	File accessible and fulfils promise / complete	Add your entry
Regression	Repeat original and adjacent steps after repair	Expected result without downstream failure / complete	Add your entry

Application questions

Can a marketing team conclusively assess accessibility on its own?

This workbook supports specific manual preliminary checks, but does not replace a complete specialist assessment. Record tested scope and open areas. Use the W3C reference and appropriate expertise for formal requirements and complex findings.

What distinguishes a defect from an optimisation idea?

A defect contradicts a clear expected outcome through observable behaviour, such as an enquiry not arriving. A different headline might work better, but that remains an impact hypothesis. They require different evidence and should stay distinguishable in the action plan.

Should I review all pages or one conversion journey first?

Start with important tasks and representative journeys. This reveals central blockages and helps establish a usable test protocol. Expand to the full inventory and unusual page types afterwards. A quick overview is not equivalent to a complete review of every page.

Is technical availability enough for visibility in language models?

No. Availability, access permissions, understandable content and actual system selection are different conditions. Record what you can specifically demonstrate. Passing a technical check cannot promise mentions, rankings or correct representation by a language model.

Sources and context

The linked primary sources support the technical foundations. Examples in this resource are fictional and are not client references. Example figures are assumptions, not promised outcomes. Check current platform capabilities before implementation.

[W3C WAI - WCAG 2.2 Quick Reference](https://www.w3.org/) (www.w3.org)

[Google Search Central - Introduction to robots.txt](https://developers.google.com/search/docs/essentials/intro) (developers.google.com)