

Testen

Projektleitung

Geschäftsführung

IT

Recht

So schaffst du Rechtssicherheit und Nutzerfreundlichkeit ohne doppelte Arbeit

Weil Barrierefreiheit kein Einmal-Projekt ist, sondern ein kontinuierlicher Qualitätssicherungsprozess. Ohne einen systematischen Testprozess wird die Zugänglichkeit deiner Website mit jedem neuen Feature, jedem Content-Update und jedem Relaunch unweigerlich wieder schlechter.

Zufällige Checks sind ineffizient und lückenhaft. Ein strukturierter Prozess stellt sicher, dass du Barrieren findest, bevor es deine Nutzer:innen tun, und schafft die notwendige Dokumentation, um deine Bemühungen im Rahmen des EAA und BFSG jederzeit nachweisen zu können.

Du wandelst damit blinden Aktionismus in eine messbare und verlässliche Qualitätssicherung um.

Wann musst du das machen?

- Vor jedem größeren Launch oder Relaunch einer Website oder App.
- Als fester Bestandteil jedes Entwicklungs-Sprints für neue Features.
- In regelmäßigen Abständen (z. B. vierteljährlich), um den Status quo der wichtigsten Seiten zu überprüfen.
- Wenn du einen Rahmen für dein internes oder externes Test-Team schaffen musst.

Welches Gesetz verlangt das?

- EAA / BFSG: Die Gesetze verlangen nicht nur die initiale Herstellung der Barrierefreiheit, sondern auch deren Aufrechterhaltung. Ein dokumentierter Testprozess ist der beste Nachweis, dass du dieser Verpflichtung nachkommst.
- EN 301 549 / WCAG 2.1: Die Normen definieren die Kriterien, die du testen musst. Diese SOP definiert das „Wie“ des Testens.

Der Prozess im Detail

1 Den Testumfang festlegen – Repräsentative Seiten auswählen

Du kannst nicht immer alles testen. Priorisiere strategisch, um mit deinem Aufwand die größte Wirkung zu erzielen. Wähle einen repräsentativen Querschnitt deiner Website aus.

- Seitentemplates: Wähle eine Seite für jeden Haupt-Seitentyp (z. B. Startseite, Produkt-Detailseite, Kategorie-Übersicht, Blogartikel, Kontaktseite).
- Kernprozesse: Teste die vollständige Nutzerreise für alle kritischen Funktionen wie den Checkout-Prozess, die Registrierung, den Login oder die Newsletter-Anmeldung.
- Seiten mit hohem Traffic: Analysiere deine Web-Statistiken und nimm die Top-5-Einstiegsseiten in den regelmäßigen Test auf.
- Komplexe Komponenten: Seiten, die interaktive Elemente wie Slider, Filter, Karten oder komplexe Formulare enthalten, müssen immer Teil des Tests sein.

2 Kombiniere Test-Methoden (Die Test-Pyramide)

Verlasse dich nie auf eine einzige Methode. Die Kombination macht den Prozess robust.

- Stufe 1: Automatisierte Tests (Basis-Check): Nutze Tools wie axe DevTools oder Lighthouse , um schnell niedrigschwellige Fehler zu finden (fehlende Alt-Texte, Kontrastfehler, fehlende Labels). Dies ist ideal für schnelle Checks durch Entwickler:innen.
- Stufe 2: Manuelle Tests (Tiefenprüfung): Dies ist der wichtigste Teil, da hier die Logik und Bedienbarkeit geprüft wird, die Tools nicht erfassen können.
- Tastatur-Test: Navigiere die gesamte Seite nur mit der Tab-Taste. Ist alles erreichbar und bedienbar? Gibt es einen sichtbaren Fokus?
- Screenreader-Test: Nutze NVDA oder VoiceOver, um die Seite wie ein blinder Mensch zu erleben. Werden alle Inhalte logisch vorgelesen?
- Stufe 3: Nutzer-Tests (Realitäts-Check): Der Goldstandard. Beziehe in regelmäßigen Abständen Menschen mit unterschiedlichen Behinderungen in deine Tests ein. Ihr Feedback ist unbezahlbar, um Barrieren zu finden, die du selbst übersehen hast.

3 Erstelle ein standardisiertes Testprotokoll

Dokumentation ist kein Selbstzweck. Sie ist dein Beweismittel und dein Werkzeug zur Fehlerbehebung. Erstelle eine Vorlage (z. B. in Excel, Notion oder Jira).

Spalte

Beispiel / Inhalt

Seite / URL

<https://beispiel.de/kontakt>

WCAG-Kriterium

2.4.7 Focus Visible

Problembeschreibung

Der „Absenden“-Button hat keinen sichtbaren Fokus-Rahmen bei Tastaturnavigation.

Priorität

Kritisch

Screenshot / Video

Link zum visuellen Beweis

Verantwortlich

[Name des Entwicklers]

Status

Offen / In Arbeit / Erledigt

4 Verankere den Test-Rhythmus im Team

Definiere, wann und wie oft getestet wird, um Barrierefreiheit zu einem festen Ritual zu machen.

Zeitpunkt

Was wird getestet?

Wer testet?

Während der Entwicklung

Neue Komponenten & Features

Entwickler:innen (automatisiert + manuell)

Vor jedem Release

Alle neuen oder geänderten Seiten/Funktionen

QA-Team (manuell nach Protokoll)

Vierteljährlich

Der gesamte repräsentative Seiten-Pool

QA-Team / EAA-Verantwortliche:r

Jährlich

Ausgewählte Kernprozesse

Externe Nutzer:innen mit Behinderungen

•

5 Werte die Ergebnisse aus und leite Maßnahmen ab

Ein Testbericht, der in der Schublade landet, ist nutzlos.

- Review der Ergebnisse: Gehe das Testprotokoll im Team durch.
- Priorisierung der Bugs: Nutze eine Matrix aus Kritikalität und Aufwand, um zu entscheiden, was sofort behoben werden muss (siehe SOP 6).
- Tickets erstellen: Lege für jeden zu behebbenden Fehler ein klares Ticket im Projektmanagement-Tool an.
- Fortschritt dokumentieren: Halte im Testprotokoll fest, wann ein Fehler behoben wurde. Dies schafft eine lückenlose Dokumentationskette.

Ergebnis

Am Ende dieser SOP hast du:

- Einen klaren, strukturierten und wiederholbaren Prozess, um Barrierefreiheit zu sichern.
- Die Verantwortung für das Testen auf verschiedene Schultern im Team verteilt.
- Eine lückenlose Dokumentation, die als Nachweis für Behörden und für die interne Qualitätssicherung dient.
- Sichergestellt, dass Barrierefreiheit keine einmalige Anstrengung bleibt, sondern ein fester Teil deiner Unternehmenskultur wird.

Und hier noch ein hoffentlich nützlicher E-Mail-Baustein, mit dem du darüber informierst, dass nun DSGVO und Barrierefreiheit synchronisiert sind:

Betreff: Einführung unseres neuen, verbindlichen Testprozesses für Barrierefreiheit

Hallo Team,

um die Qualität und rechtliche Konformität unserer digitalen Produkte (EAA/BFSG) nachhaltig zu sichern, führen wir ab sofort einen systematischen Testprozess für Barrierefreiheit ein.

Dieser Prozess ist für alle neuen Features und für regelmäßige Audits verpflichtend. Er kombiniert automatisierte, manuelle und nutzerbasierte Tests.

Die Details zum Prozess, die zu prüfenden Seiten und die Protokoll-Vorlage findet ihr hier: [Link zum Dokument in Confluence/Notion]

Die wichtigsten Verantwortlichkeiten sind:

- Entwicklung: Führt automatisierte Tests durch.
- QA: Führt manuelle Tastatur- und Screenreader-Tests für jedes Release durch.

Dieser Prozess ist entscheidend, um unsere gesetzlichen Pflichten zu erfüllen und eine exzellente User Experience für alle zu gewährleisten.

Viele Grüße [Dein Name]

Fragen & Antworten

Wie oft sollten wir Nutzer-Tests mit echten Menschen mit Behinderungen durchführen, und wie finden wir geeignete Testpersonen?

Nutzer-Tests mit echten Menschen sind der Goldstandard, aber auch der aufwendigste Teil.

- Häufigkeit: Idealerweise einmal jährlich für deine wichtigsten Kernprozesse oder bei größeren Redesigns. Für sehr dynamische Produkte können auch kürzere Intervalle (z.B. alle 6 Monate) sinnvoll sein. Wichtiger als die Frequenz ist die Qualität des Feedbacks und die Bereitschaft, daraus zu lernen.
- Testpersonen finden:
 - Organisationen für Menschen mit Behinderungen: Viele gemeinnützige Organisationen bieten Kontakt zu Testpersonen oder vermitteln diese direkt.
 - Spezialisierte Agenturen: Es gibt Agenturen, die sich auf Barrierefreiheitstests mit Menschen mit Behinderungen spezialisiert haben.
 - Universitäten/Forschungseinrichtungen: Manchmal gibt es hier Kooperationsmöglichkeiten im Bereich Mensch-Computer-Interaktion.
 - Communitys und Foren: Mit der nötigen Sensibilität kann man auch in Online-Communitys nach Freiwilligen suchen, aber hier ist Vorsicht geboten.

Wichtig ist immer, du, die Testpersonen angemessen zu vergüten für ihre Zeit und ihr wertvolles Feedback.

Die SOP spricht von einem %22repräsentativen Querschnitt%22 der Seiten. Wie detailliert sollte diese Auswahl sein, um wirklich aussagekräftige Ergebnisse zu liefern?

Ein repräsentativer Querschnitt ist entscheidend, um mit begrenzten Ressourcen die maximale Abdeckung zu erzielen. Er sollte nicht nur die Seitentypen und Traffic-Spitzen abdecken, sondern auch:

- Alle einzigartigen Module/Komponenten: Jedes Modul, das eine einzigartige Interaktion oder Darstellung bietet und auf verschiedenen Seiten verwendet wird (z.B. ein spezieller Slider, ein Filter, ein komplexes Menü), sollte mindestens einmal in einem Testumfeld vorkommen.
- Verschiedene Browser/Geräte: Teste nicht nur auf einem Gerät, sondern auch auf verschiedenen Browsern (Chrome, Firefox, Safari) und Bildschirmgrößen (Desktop, Tablet, Mobile), da die Barrierefreiheit dort variieren kann.
- Sprachen (falls mehrsprachig): Wenn deine Website mehrsprachig ist, teste wichtige Seiten in jeder angebotenen Sprache, um sicherzustellen, dass Sprachkennzeichnungen und Übersetzungen korrekt sind.
- Login-Bereiche/Personalisierung: Wenn deine Website personalisierte Bereiche oder Login-Funktionen bietet, teste auch diese vollständig, da hier oft spezifische Barrieren entstehen.

Je diverser der Testumfang, desto umfassender die Erkenntnisse.

Die SOP empfiehlt automatisierte Tools wie axe DevTools oder Lighthouse als %22Basis-Check%22. Was sind die typischen Fehler, die diese Tools nicht finden können und die manuelle Tests unverzichtbar machen?

Automatisierte Tools sind großartig für einen schnellen Überblick und das Auffinden von offensichtlichen technischen Fehlern, aber sie sind „blind“ für den Kontext und das Nutzererlebnis, du. Typische Fehler, die sie nicht finden können , sind:

- Fokus-Reihenfolge: Ob der Tastaturfokus logisch durch die Seite springt.
- Tastaturfallen: Ob ein Nutzer in einem Dialog oder einer Komponente gefangen ist.
- Sinnhaftigkeit von Alternativtexten: Sie können erkennen, ob ein alt -Attribut fehlt, aber nicht, ob der Text („Bild123.jpg“) aussagekräftig ist.
- Logischer Lesefluss für Screenreader: Ob Inhalte in einer sinnvollen Reihenfolge vorgelesen werden (z.B. ob eine Überschrift vor dem zugehörigen Text kommt).
- Verständlichkeit von Texten: Ob komplexe Sätze oder Fachjargon die Verständlichkeit beeinträchtigen.
- Interaktive Komponenten: Ob Dropdowns, Slider oder komplexe Filter mit der Tastatur vollständig bedienbar sind.
- Semantische Beziehungen: Ob Elemente, die visuell zusammengehören, auch für Screenreader als Einheit wahrgenommen werden.

Genau deshalb sind manuelle Tests und insbesondere Screenreader-Tests so unverzichtbar.

Unser Team ist klein und hat keine dedizierten QA-Ressourcen. Wer sollte die manuellen Tastatur- und Screenreader-Tests durchführen?

In kleineren Teams muss die Verantwortung oft geteilt werden. Hier sind Optionen:

- Entwickler:innen selbst: Jeder Entwickler sollte für die Barrierefreiheit der von ihm entwickelten Features selbst mitverantwortlich sein und grundlegende Tastatur- und Screenreader-Tests durchführen. Das fördert „Accessibility by Design“.
- UX-Designer:innen: Da sie die Nutzerführung gestalten, sollten sie ebenfalls grundlegende Tests durchführen können, um die User Experience für alle zu validieren.
- Projektleitung/EAA-Verantwortliche:r: Du selbst solltest die wichtigsten Kernprozesse regelmäßig stichprobenartig prüfen, um ein Gefühl für den Status Quo zu bekommen und die Einhaltung des Prozesses zu überwachen.
- Externe Dienstleister: Wenn das interne Know-how oder die Zeit fehlt, kann es sinnvoll sein, externe Dienstleister für regelmäßige manuelle Tests zu beauftragen.

Wichtig ist, dass diese Tests nicht übersprungen werden, auch wenn die Ressourcen knapp sind. Sie sind der kritischste Teil deines Testprozesses.

Wir nutzen ein Projektmanagement-Tool (z.B. Jira, Asana, Trello). Wie integriere ich das standardisierte Testprotokoll und die Fehlerbehebung am besten in unseren bestehenden Workflow?

Die Integration in dein bestehendes PM-Tool ist der Schlüssel zur Effizienz.

- Ticket-Vorlage: Erstelle eine spezifische Ticket-Vorlage für Barrierefreiheits-Bugs. Diese Vorlage sollte Felder haben für:
 - WCAG-Kriterium (z.B. „2.4.7 Focus Visible“)
 - Problembeschreibung
 - Betroffene URL / Komponente
 - Screenshot / Video des Problems
 - Priorität (z.B. „Kritisch“, „Hoch“, „Mittel“)
 - Verantwortlicher
 - Status
- Labels/Tags: Verwende spezifische Labels oder Tags wie „Accessibility Bug“, „A11y-Test“, um diese Tickets leicht identifizieren und filtern zu können.
- Dashboards/Berichte: Erstelle Dashboards oder Berichte, die den Status der Barrierefreiheits-Bugs visualisieren, sodass du den Fortschritt und die offenen Baustellen immer im Blick hast.
- Workflows: Integriere Barrierefreiheitstests und -behebungen fest in eure Sprints oder Release-Workflows, sodass sie nicht vergessen werden.