

Praxis und Stolpersteine bei A1 1y-Audits

A11y TUGA95

Jürgen Bartl

- Software Usability Ende 1990er, 2003 TYPO3, 2004 Accessibility
- 2006 BIENE, eAward Austria, Alternativer Medienpreis
- Begleitung A1 1y-Projekte/Audits und A1 1y-Umsetzungen:



≡ Bundeskanzleramt



Ministerium für Migration, Justiz und Verbraucherschutz

≡ Bundesministerium Soziales, Gesundheit, Pflege und Konsumentenschutz



≡ Bundesministerium Klimaschutz, Umwelt, Energie, Mobilität, Innovation und Technologie



≡ Entsorgungsbeirat

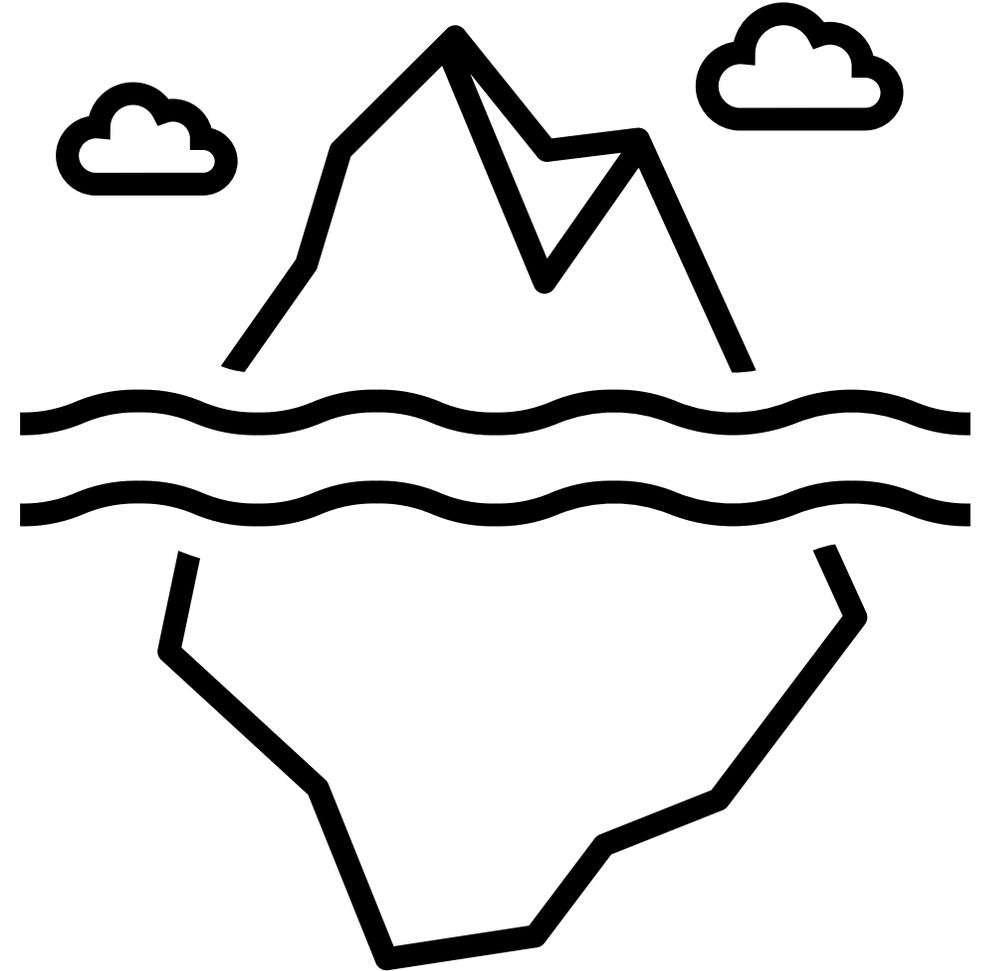


≡ Bundesministerium Bildung, Wissenschaft und Forschung



Startschwierigkeiten

- Audit beginnt, wenn das Projekt schon fertig ist
- Konzeptentwicklung ohne „A11y in Mind“
- Nicht barrierefreies Konzept schon vervielfältigt:
 - blog.* / shop.* / karriere.* / magazin.* / presse.* / landing.* / lifestyle.* / b2b.* / login.* / ...
- -> nachträgliche UI-Konzeptänderung = hoher Aufwand
- Kosteneinplanung für den A11y-Prozess mangelhaft
- Hilflöse Versuche mit Overlays



Overlays

<https://overlayfactsheet.com/en/>

Alle Argumente in einer URL:

- Mit Overlays keine WCAG-Konformität erreichbar
- Großteils redundante Funktionalität, die Benutzer_innen selbst im Browser steuern können (Kontrast, Typografie, Farben)
- Manche Dinge können automatisch repariert werden, die meisten nicht
- DSGVO ggf. problematisch
- Die große Mehrheit der Benutzer_innen lehnt Overlays ab

Große Projekte – große Probleme

Typische große Brocken:

- Verkaufsstrecken und Checkout-Prozesse: Komplex/aufwändig zu sanieren.
Tasks: Übersicht, Robustheit, Vorhersehbares Verhalten, Browser-Backbutton
- Kundenportale (Rich Internet Applications)
- Session Clocks / Timeouts
- Formulare: Validierung, Fehlerbehandlung, Kennzeichnung, Unterstützung (WCAG SC 3.3 Input Assistance) / Uploads
- Abhängigkeiten (Backend, automatisch erzeugte PDFs)
- Furchtbar schlechter Code: Semantik, `<div class=„headline-400“>`, ARIA-Missbrauch
- UI-Komponenten: gut funktionierende nicht einfach zu finden / müssen meist modifiziert werden

Beispiel 1: Comboboxen

The image shows two examples of comboboxes in a web form. The first combobox is labeled 'Handelsname (Auswahl aus Liste oder manuelle Eingabe) [erforderlich]:'. The search input contains 'cov' and the dropdown list shows several COVID-19 vaccine options: 'COVID-19 Vaccine BioNTech / Pfizer (Comirnaty)', 'COVID-19 Vaccine Moderna (Spikevax)', 'COVID-19 Vaccine AstraZeneca (Vaxzevria)', and 'COVID-19 Vaccine Janssen (Johnson)'. Below the combobox is a green button labeled '+ Weitere Impfung oder Arzt'. The second combobox is labeled 'Reiseziele (erforderlich)'. The search input contains 'Österreich' and the dropdown list shows a list of countries: 'Afghanistan', 'Aland Islands', 'Algerien', and 'Amerikanischen Samoa-Inseln'. Above the dropdown list are two buttons: 'alles loeschen x' and 'Österreich x'. To the right of the dropdown list is a link labeled 'Info'.

- Info für Screenreader beim Reintabben und navigieren
- aria-autocomplete, role=„combobox“, aria-owns, aria-placeholder, aria-describedby, aria-controls ...
- Kommunikation über ARIA live regions: Mitteilung, wenn Ergebnisse vorhanden sind und wie navigiert werden muss, was im Feld steht

Beispiel 2: Auto-Suggest

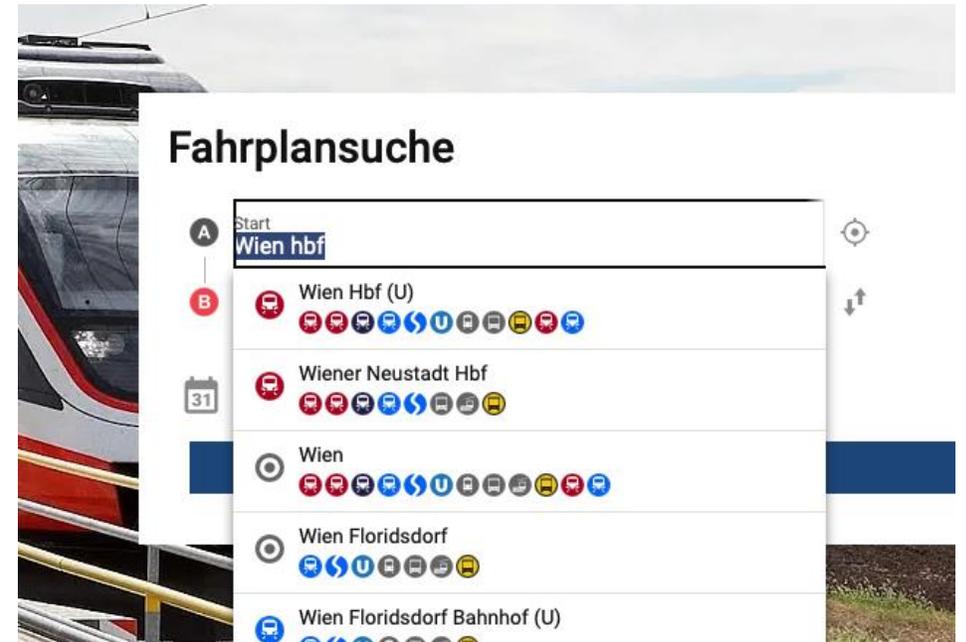
- Navigation innerhalb Widgets sicherstellen
- Kommunikation, wie navigiert werden muss
- Mitteilung über Erfolg: „x Ergebnisse verfügbar“
- Kommunikation, was ausgewählt ist
- Muss mit Screenreader getestet werden, tastaturtauglich alleine reicht nicht

<div id="announce" class="visually-hidden" aria-live="assertive"> 4 Vorschläge gefunden.

Zum Navigieren verwenden Sie die Auf- und Abwärtspfeile </div>

aria-live="assertive": für zeitkritische/kritische Benachrichtigungen, die unbedingt die sofortige Aufmerksamkeit des Benutzers erfordern.

aria-live="polite": Wird gesprochen, wenn Benutzer Pause macht



Q fahr|

SUCHEN

FAHRRAD
FAHRRADVERSICHERUNG



Radwandern » Tipps für einen gelungenen Ausflug | Generali Blog

Radwandern leicht gemacht: Tipps für Ihre *Fahrradtour* 20.09.2023 | Artikel Die Satteltaschen gepackt, die Karte studiert, die Reifen aufgepumpt – auf...



DAVID Institut

für die Gesundheit aus Blut und Harn. Spiroergometrie: Belastungstest am *Fahrrad* mit Herz

Beispiel 3: Session Clock

2 unterschiedliche Felder Visible / Screen Reader:

```
<div role="status" aria-live="polite" id="session-timer-sr" class="timer-hidden">Automatische Abmeldung in 10 Minuten</div>
```

```
<div role="timer" aria-hidden="true">Automatische Abmeldung: <span id="session-timer">09:17</span> Min.</div>
```

Warnung 1 Minute vor Ende:

```
<div class="modal" tabindex="-1" role="alertdialog" id="session-dialog" aria-modal="true" aria-labelledby="session-title" aria-describedby="session-dialog-text">
```

```
[...] <h5 id="session-title" class="modal-title">Die Sitzungszeit läuft in 1 Minute ab!</h5>
```

-> Focus im Modaldialog -> Button OK -> Focus Ausgangsposition!



Automatische Abmeldung: 09:24 Min.

Online-Meldung von Nebenwirkungen human

Angaben zur meldenden Person (Schritt 1/6)

Online-Meldung von Nebenwirkungen human

Automatische Abmeldung in 10 Minuten
Automatische Abmeldung: 09:45 Min.

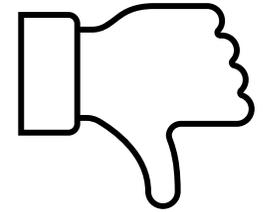
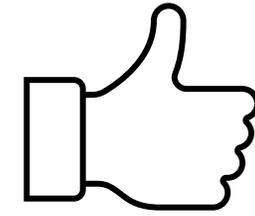
Angaben zur meldenden Person
(Schritt 1/6)



Was macht einen guten Testprozess aus?

- Überblick schaffen / Vorbericht mit Eingrenzung Hauptprobleme
- Komplexität herausnehmen, Projekt splitten: „Low hanging Fruits“
- Teilbereiche abreißen & neu bauen statt sanieren oft einfacher
- Testen, testen, testen: Fokus Mobilgeräte (bis 60% Mobilzugriffe)
- Eigene Screenreadertests (Know How aneignen), Mobile Screen-Reader verhalten sich tw. anders (Schrittmodus)
- Screenreader-Benutzer_innen-Test einfließen lassen: SR-Test gut, aber bewerten / keine anderen Barrieren schaffen
- Gutes, nachvollziehbares Prüfverfahren, z.B. BIK: <https://bitvtest.de/pruefverfahren/wcag-22-web>
- FFG Report Tool: <https://wad-report-tool.ffg.at>

Gutes Audit? Schlechtes Audit?

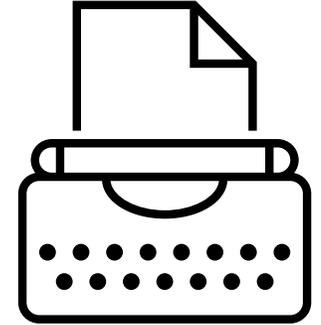


- Es werden nicht nur Prüfpunkte erfüllt/nicht erfüllt aufgelistet
- Detaillierte Analyse für jedes Problem + erwartetes UI-Verhalten
- Success Criteria betroffen, Links auf W3C/Prüfverfahren, Textliche Beschreibung, warum das ein Problem ist.
- Lösungsansätze: Es gibt viele gute Best Practice Lösungen (High-End Beispiele öffentlicher Sektor/Bund) – Auditor_in sollte gute Kenntnisse haben, was sich bewährt hat und warum
- Kein Best Practice verfügbar: Auditor_in recherchiert/testet Vorschläge und trennt Spreu vom Weizen (UI-Komponenten)
- Unterstützung bei konzeptionellen Barrieren (Design, Umstrukturierung, Ändern von Grafiken, Inhaltsvermittlung, Wording etc.) und Office-Formaten
- Integration ins Team -> Coaching

Unterstützung Entwickler_innen

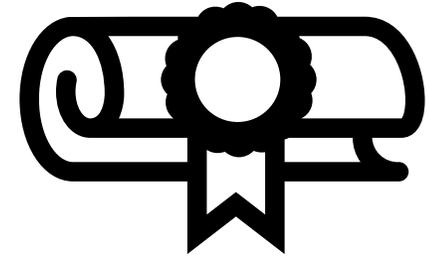
- Für einzelne Interface-Elemente gute Beispiele geben: Slider, Tab Panels, Formularfehlerhandling, Accordions, Burger Menus, Comboboxen etc.
- Quell-Literatur und Links angeben: Nicht nur Prüfpunkte sondern recherchierte Lösungsansätze (Experten, Öffentliche Projekte, Webaim, Blogbeiträge o.ä.)
- Hilfestellung bei FE-Coding: Debugging mit Screenreader, Prototypen-Tests während Entwicklung
- Screenreader-User_innentest fließt im Idealfall mit ein / Zeitpunkt individuell

Redaktion und Multimedia



- Redaktion muss geschult werden: Wie wird getextet?
- Cheat Sheets verfassen: Alttexte, Konventionen/Regelwerk für wiederkehrende Prozesse: Linktexte, Dateiformat- und Größenangaben, Kennzeichnungen externer Links oder Kontextwechsel, Telefonnummern, Mailadressen etc.
- PDF! Kompletten Workflow überarbeiten: Indesign-/Word-Vorlagen, Schulung und Anleitungen für Export, Letzte Instanz A11y-Checks Ergebnis/Bugfixing
- Videos: Nicht trivial / Transkript + UT + Audiobeschreibung

Launch und danach



- Barrierefreiheitserklärung erstellen, ggf. Abstimmung Zertifizierungsprozess WACA-Zertifikat
- Realistische Prognosen für noch vorh. Barrieren (PDF, Medien)
- A11y-QS: Alles, was sich ändert -> vor Einsatz testen, bewerten: Neue UI-Komponenten, Zahlungsanbieter, Karten-Interfaces, SEO-Maßnahmen etc.
- Regelmäßige Re-Checks: Nichts bleibt, wie es ist

DANKE!

Ein paar Links

- Zugängliche Combobox mit Multi-Select
<https://mynameleon.github.io/aria-autocomplete/>
(Geringfügige Modifikation nötig / invalider CSS-Code – ansonsten Mobile Voice Over Probleme)
- Accessible auto-suggest mit Codepen-Beispiel:
<https://uxmastery.com/anatomy-of-an-accessible-auto-suggest/>
<https://codepen.io/ademcificioglu/pen/xdOyXv>
- Prüfverfahren BIK WCAG 2.2 (Web)
<https://bitvtest.de/pruefverfahren/wcag-22-web>
- FFG Report Tool
<https://wad-report-tool.ffg.at> / Original WCAG-EM Report Tool <https://www.w3.org/WAI/eval/report-tool/>
- WCAG-EM Bewertungsmethodik:
<https://www.w3.org/TR/WCAG-EM/>