

1 Barrierefreiheit verstehen

Behinderungen und Barrieren



Blindheit

Nutzen weder Bildschirm noch Maus. **Screenreader** und **Braillezeile** wandeln Text in taktile oder akustische Informationen.



Motorische Behinderung

Können ihre Arme nur eingeschränkt bewegen und bedienen den PC über **Spezial-Maus** und **Bildschirmtastatur**



Sehbehinderung

Sehen trotz Sehhilfe weniger als 30% und benötigen eine **Bildschirmlupe** und gute **Kontraste** um noch etwas zu erkennen.



Kognitive Behinderung

Haben Probleme beim Lernen und Verstehen und sind deshalb auf **einfache Inhalte** angewiesen.



Gehörlosigkeit

Haben nie Sprache gehört und benötigen dadurch **einfache Texte** oder **Gebärdensprachvideos**.



Technische Probleme

Probleme durch unpassende oder inkompatible Infrastruktur. Das Einhalten von **Standards** und **Flexibilität** hilft.

Durch **Behinderungen** entstehen im Internet **Barrieren**. Diese können teilweise durch **assistive Technologien** aufgehoben werden. Besser ist jedoch ein **barrierefreies Design**.

Standards

Weltweite Standards des W3C

Web Accessibility Initiative WAI

Richtlinien für Barrierefreiheit des W3C.

WCAG 2.0

HTML

WAI-ARIA

Erw. für AJAX

ATAG

Programme zum Erstellen von Webseiten

UAAG

Browser und assistive Technologien

Schweizer Standards

eCH-0059

Gemeinwesen

P028

Bundesrichtlinien

Für Webseiten von **öffentlichen Institutionen** ist Barrierefreiheit gesetzlich vorgeschrieben.

3 Überprüfen und zertifizieren

Kleine Barrieren können in der Praxis zu grossen Problemen führen. Deshalb sollte die **umgesetzte Barrierefreiheit** in jedem Fall **überprüft** werden.

Zu Beginn des Projekts



Anforderungen an die Barrierefreiheit festlegen

- Welche Teile von welchen Richtlinien (Standards) sollen befolgt werden
- Was muss erfüllt und geprüft werden (messbare Kriterien)



Testen von visuellen Prototypen, Struktur und Navigation

- Sobald erste Prototypen bestehen, sollten diese gegenüber den Richtlinien überprüft werden
- In Design, Struktur und Navigation liegen die Hauptursachen für Barrieren

Bei jeder grösseren Änderung



Wichtige Punkte durch schnelle Tests prüfen

- Tastaturbedienbarkeit
- Textgrösse im Browser auf «Extra Gross» stellen
- Inhalt ohne CSS, Bilder und JavaScript betrachten
- HTML-Validator (W3C)
- WCAG-Checker (WAVE, TotalValidator)

Gegen Ende des Projekts



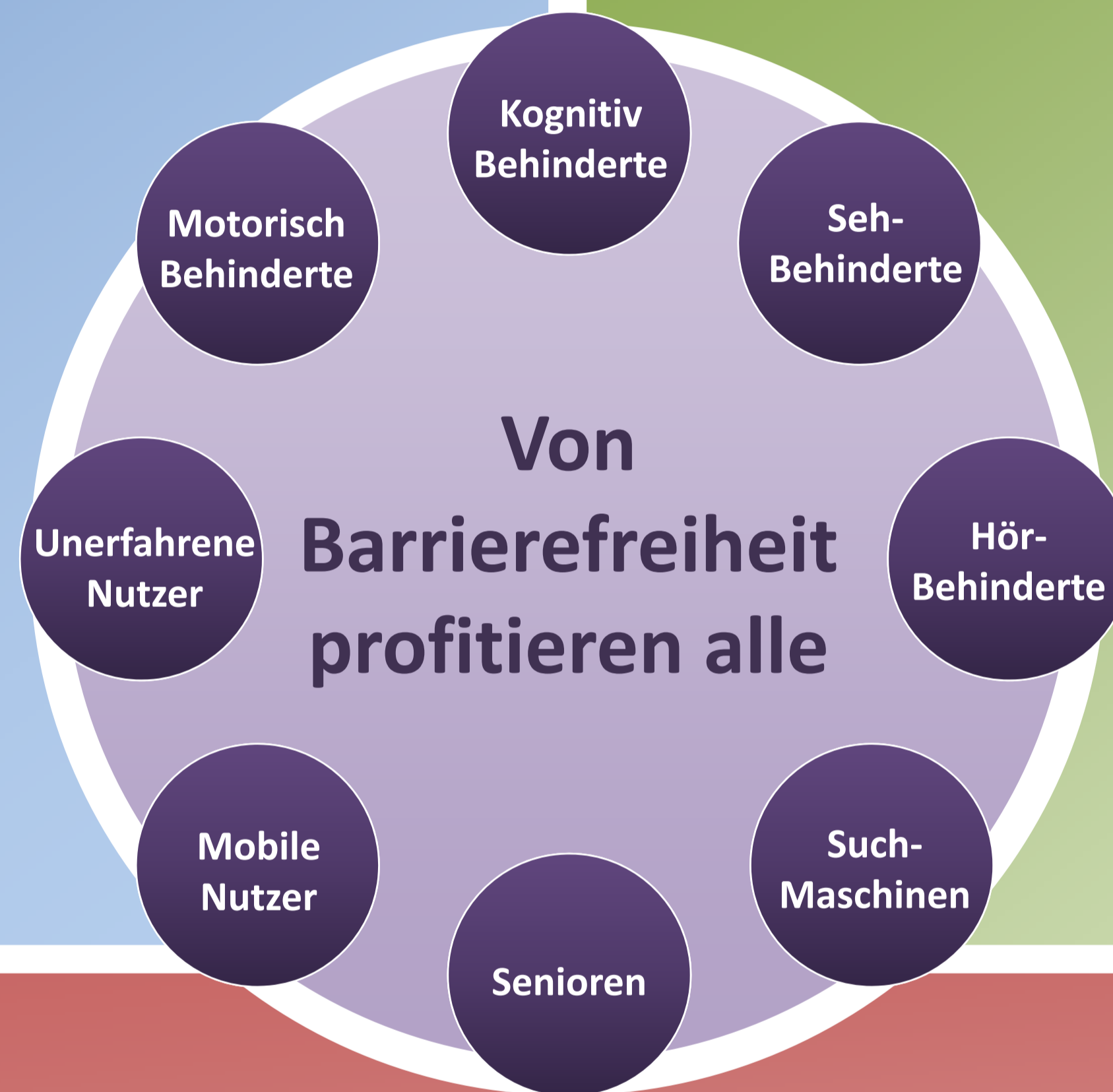
Komplette Webseite auf Barrierefreiheit prüfen

- WCAG Konformitätstest (z.Bsp. mittels Checklisten von access-for-all.ch)



Zertifizierung

- Externe Prüfung der Webseite durch Experten und Behinderte
- Zertifizierung für WCAG 2.0 A, AA oder AAA



2 Umsetzung in der Praxis

Leitgedanken

Selbstbestimmungs-Prinzip

Das Design sollte **möglichst wenig fix** vorgeben. So kann der Nutzer Seitenformat, Farbe oder Schriftgrösse auf seine Bedürfnisse anpassen und damit manche Barrieren vermeiden.

Universelles Design

Es sollte nur **eine Lösung für alle** Nutzer geben. Sonderlösungen für alte, blinde, oder mobile Nutzer werden oft nicht benutzt, sind nicht vollständig oder aktuell und aufwändig in der Wartung.

Zwei-Sinne Prinzip

Stellen Sie sicher, dass **alle Informationen**, mit Hilfsmitteln, **über zwei Sinne** wahrgenommen werden können. Die Wahrscheinlichkeit, dass bei einem Nutzer zwei Sinne behindert sind, ist deutlich kleiner.

Allgemeine Designprinzipien

Wahrnehmbarkeit

- Textalternativen für grafische Inhalte anbieten
- Untertitel für Audio- und Videodateien
- Flexible Darstellung (Format, Farben, Schriftgrössen)

Bedienbarkeit

- Mit der Tastatur bedienbar
- Genügend grosse Timeouts
- Design darf keine Anfälle verursachen
- Navigationshilfen und Ortsangaben anbieten

Verständlichkeit

- Einfache und verständliche Texte
- Konsistenter Aufbau und gute Selbsterklärbarkeit
- Eingabehilfen und aktive Fehlervermeidung

Robustheit

- Maximale Kompatibilität mit Browsern und Hilfsmitteln

Barrierefreies HTML

Allgemeines

- Dokumentsprache** richtig setzen (xml:lang="de")
- Wichtige Infos in **Meta-Tags** und **Seitentitel**

Layout und Navigation

- Trennung von Inhalt und Design** durch CSS
- Layout durch <div> Elemente, **keine Tabellen**
- Echte Überschriften** (<h1>,<h2>) verwenden
- Versteckte Überschrift** für Navigationsblock einfügen
- Navigation sollte **auch ohne Javascript** funktionieren

Grafiken und Tabellen

- Inhalt von Grafiken im **alt-Attribut** beschreiben.
- Tabellen mit Header** (<th>) und Summary ausstatten

Formulare

- Eingabefelder mit einem **<label>-Element** beschriften
- Labels nahe** an Eingabefeldern platzieren

Barrierefreie Rich Applications

AJAX

- Regionen mit dem **role**-Attribut beschreiben
- Aktualisierbare Inhalte mit **aria-live** markieren
- Tastaturnavigation durch setzen des **tabindex**-Attributs
- Mit HTML und JavaScript nachgebaute Widgets (Slider, Tabs,Tree,...) mit dem **role**-Attribut beschreiben

Adobe Flash

- Beschriftung grafischer Elemente** mit Informationsgehalt
- Definieren einer sinnvollen **Tab-Reihenfolge**
- Audiodeskription** für Multimediainhalte anbieten

Microsoft Silverlight

- Tastaturkürzel, Alt-Text und Bezeichnung von Elementen über **AutomationProperties** setzen
- Auf veränderte **Farbschema-Einstellungen** reagieren
- Zoom-Funktionalität anbieten

Barrierefreie Dokumente



Adobe PDF

- Setzen von **Tags** zur Navigation. Mit Acrobat über das Menu **Advanced->Accessibility**
- Logische Lesereihenfolge** sicherstellen (Besonders bei mehrspaltigen Dokumenten)
- Alternativ-Texte** für Grafiken anbieten
- Dokumentsprache** richtig setzen
- Tabellenüberschriften** für alle Datentabellen



Microsoft Word

- Korrekte Verwendung von **Überschriften**
- Alternativ-Texte** für Grafiken setzen
- Falls benötigt, **«echte» Spalten** verwenden