



CODERS.BAY

BARRIEREFREIHEIT

CRASHKURS

Das BIK-Projekt – barrierefrei informieren und kommunizieren – bietet in einer offiziellen Präsentation folgende Argumente an:

- Vorsorge für künftige Technologien
- Maßstab für Nutzerfreundlichkeit
- Browserunabhängigkeit
- Geräteunabhängigkeit
- Nutzung eines komfortablen CMS
- Valider Code für HTML und CSS
- Neue Zielgruppen
- Imagegewinn durch soziale Kompetenz

Der Mehraufwand für die Optimierung einer Webanwendung auf Barrierefreiheit beläuft sich Schätzungen zufolge auf unter 15% der Gesamtkosten eines Webprojekts. Zum Vergleich: die Optimierung einer Webapplikation für unterschiedliche Browser, also eine Browserunabhängigkeit, beträgt geschätzt 10% der Projektgesamtkosten

Bedienung:

- Im Hinblick auf die Bedienung eines CMS ist eine barrierefreie Bedienung des Backends als der Verwaltungsebene des Systems unabdingbar
- Technisch ist der Inhalt vom eigentlichen Layout zu trennen und somit sind Verantwortungsbereiche klar definiert
- Der Redakteur kann Inhalte pflegen und neu erstellen, ohne dabei die Gestaltung anpassen zu müssen. Das fällt nicht zwangsläufig in seinem Aufgabenbereich
- Das System bietet eine klare Benutzerverwaltung, die möglichst auf feingranularer Ebene eine Zugriffssteuerung auf die verschiedenen Bereiche und Möglichkeiten des CMS zulässt
- Darüber hinaus ist über das Rollen- und Rechtemanagement ein mehrstufiges Genehmigungsverfahren realisierbar und die Freischaltung von modifizierten und neuen Inhalten ist steuerbar

Erstellung

- Das CMS unterstützt die barrierefreie Erstellung einer Webanwendung
- Es nimmt Anweisungen entgegen und setzt sie in validen und den gültigen Spezifikationen der W3C und der BITV entsprechenden Code um
- Das System ist auch ohne JavaScript lauffähig

Automatisierung

- Das System stellt Oberflächen für die Administratoren zur Pflege des CMS bereit, auch wird der Redakteur mit entsprechenden Masken zur Pflege des Inhalts unterstützt
- Der Redakteur pflegt seine Inhalte in einem Editor, der einer Textverarbeitung ähnlich ist. Das System setzt den darin gepflegten Inhalt automatisch in barrierefreies HTML Markup um
- Das System unterstützt den Redakteur beim Bearbeiten von Inhalten und weist ihn auf mögliche Barrieren hin, beispielsweise eine fehlende Bildbeschreibung
- Der Editor, mit dem der Redakteur die Inhalte des Systems pflegt, stellt Assistenten zur Erstellung von Standardinhalten (Tabellen, Listen, etc. bereit und unterstützt damit die Erstellung barrierefreier Webseiten
- Die Auszeichnung fremdsprachiger Wörter und Textpassagen wird vom CMS teilweise automatisch übernommen. Auf Basis manuell hinterlegter Wortlisten kann das System fremdsprachige Passagen markieren

Grundsätze einer Applikation

Klarheit

Die Benutzerschnittstelle einer Anwendung muss eindeutig sein und eine klare Sprache, Fluss und Hierarchie für die visuellen Elemente aufweisen. Optimal an dieser Stelle ist es, wenn auf ein Handbuch zur Erläuterung der Funktionsweise einer Anwendung verzichtet werden kann. Es sollte einem Benutzer nicht möglich sein, unter normalen Umständen einen Fehler zu provozieren

Kompaktheit

Häufig wird ein Anwender durch permanente Hilfe, Beschreibungen, Warnungen und Hinweise durch eine Applikation navigiert. Häufig führt das zu einer Überflutung von Informationen und Beschreibungen, sodass der Nutzer die Orientierung verliert. Klarheit und Kompaktheit miteinander zu verbinden und beide Ansprüchen gerecht zu werden, ist häufig eine große Herausforderung

Vertrautheit

Baue bei dem Design einer Anwendung hauptsächlich auf vertraute Steuerelemente und Programmabläufe. Innovation ist gut und sollte in einer Anwendung durchaus ihren Platz finden, im Kern solltest du es jedoch nicht übertreiben. Ziel ist es, dem Nutzer das Gefühl zu geben eine Anwendung als vertraut einzustufen, obwohl er sie gerade das erste Mal bedient. Überflute deshalb nicht mit zu vielen neuen Steuerelementen

und Programmabläufen. Genauso solltest du Veraltetes und Überzogenes meiden. Sowohl zu viel Innovation als auch zu viel Tradition können kontraproduktiv sein, die Mischung macht's.

Reaktionsfreudigkeit

Erstrebenswert ist es, dass eine Anwendung nach einem Klick oder einer Tasteneingabe sofort eine Rückmeldung gibt und den Benutzer informiert, was genau im nächsten Schritt passiert. Eine schnelle unmittelbare Reaktion nach einer Interaktion ist hier die Anforderung. Sicher kann die jeweilige Interaktion mit der Anweisung von Fall zu Fall etwas länger dauern, lass darüber deinen Anwender aber nicht im Ungewissen, sondern informiere ihn über eventuelle Ladezeiten und zeige ihm an, was genau die Anwendung gerade macht

Konsistenz

Hierbei geht es vor allem darum, dass gleiche Aktionen in einer Anwendung auch immer gleich ausgelöst werden. Hierzu zählen größtenteils die Anordnung und die Gestaltung der einzelnen Masken einer Software. Beispielsweise sollten Schaltflächen zum Auslösen einer Interaktion entweder immer rechts oder konsistent links angezeigt werden. Eine Maske sollte sich ausschließlich im Inhalt von einer anderen Maske unterscheiden und nicht in der wesentlichen Struktur und im Aufbau

Ästhetik

Oftmals fehlt es vielen Entwicklern an Kreativität und einer Ader bzw. Affinität zu optisch ansprechenden Anwendungslayouts. Häufig wird der Aspekt verdrängt, dass gerade die Optik und ein ansprechendes Design einer Software enorm wichtig für den Anwender sind. Hier entscheidet sich, ob ein Nutzer gerne mit deiner Anwendung arbeitet und wie ansprechend sie auf ihn wirkt. Untersuchungen und persönliche Erfahrungen zeigen immer wieder, dass eine optisch ansprechende Anwendung den Nutzer über kleinere Fehler und Ungereimtheiten einer Software hinwegsehen lassen

Effizienz

Gute Applikationen sind produktiv und unterstützen den Anwender maßgeblich bei seiner täglichen Arbeit oder bei der Bewältigung anfallender Aufgaben. Hier dreht sich das Hauptaugenmerk immer um die Aufgabe, in Bezug auf Effizienz sind das Design oder das Layout einer Anwendung weniger wichtig. Beides wird aber ernsthaft bearbeitet werden müssen, um eine Anwendung auch tatsächlich effizient zu gestalten

Toleranz

Benutzer machen Fehler. Im Idealfall rechnet eine Applikation aber genau damit und geht problemlos mit Fehlereingaben und Unregelmäßigkeiten um. Die Anwendung sollte den Anwender dabei unterstützen, aus einer fehlerhaften Aktion wieder in einen geregelten Programmfluss zu kommen. Hier empfiehlt sich eine Undo-Funktion, die eine Benutzereingabe oder -aktion problemlos rückgängig machen kann. Sicherheitsabfragen sind hier der falsche Weg, weil sie dem Benutzer bereits vor dem Ausführen einer Aktion einen Fehler unterstellen. Zudem sind solche Abfragen alles andere als benutzerfreundlich und bei einer Undo-Funktion auch vollkommen überflüssig

Barrierearm statt barrierefrei

Verstärkt setzt sich in der letzten Zeit der Begriff des barrierearmen Webdesigns durch. Das beruht auf der Tatsache, dass eine 100%-ige Barrierefreiheit einer Webanwendung nicht zu erreichen ist. Verschiedene Benutzerinteressen, Hilfsmittel und technische Voraussetzungen machen die Barrierefreiheit einer Anwendung unmöglich

Säulen der Barrierefreiheit

Eine barrierefreie Webanwendung basiert grundsätzlich auf zwei Säulen: der Accessibility und der Usability

Accessibility

Unter Accessibility ist die Zugänglichkeit bzw. Erreichbarkeit in Bezug auf eine Anwendung zu verstehen. Sie beschreibt die Fähigkeit, Informationen für jeden Benutzer zugänglich zu machen, unabhängig von technischen und körperlichen Voraussetzungen und Einschränkungen. Um die Zugänglichkeit einer Anwendung deutlich zu verbessern, können diverse Maßnahmen getroffen werden. Ziel ist es, nahezu keinen Benutzer von der Verwendung der Anwendung auszuschließen. Es gilt Barrieren, die sich beinahe immer bei der Entwicklung und dem Design einer Webanwendung einschleichen, zu erkennen und mit geübten Handgriffen zu beseitigen.

Usability (Benutzerfreundlichkeit)

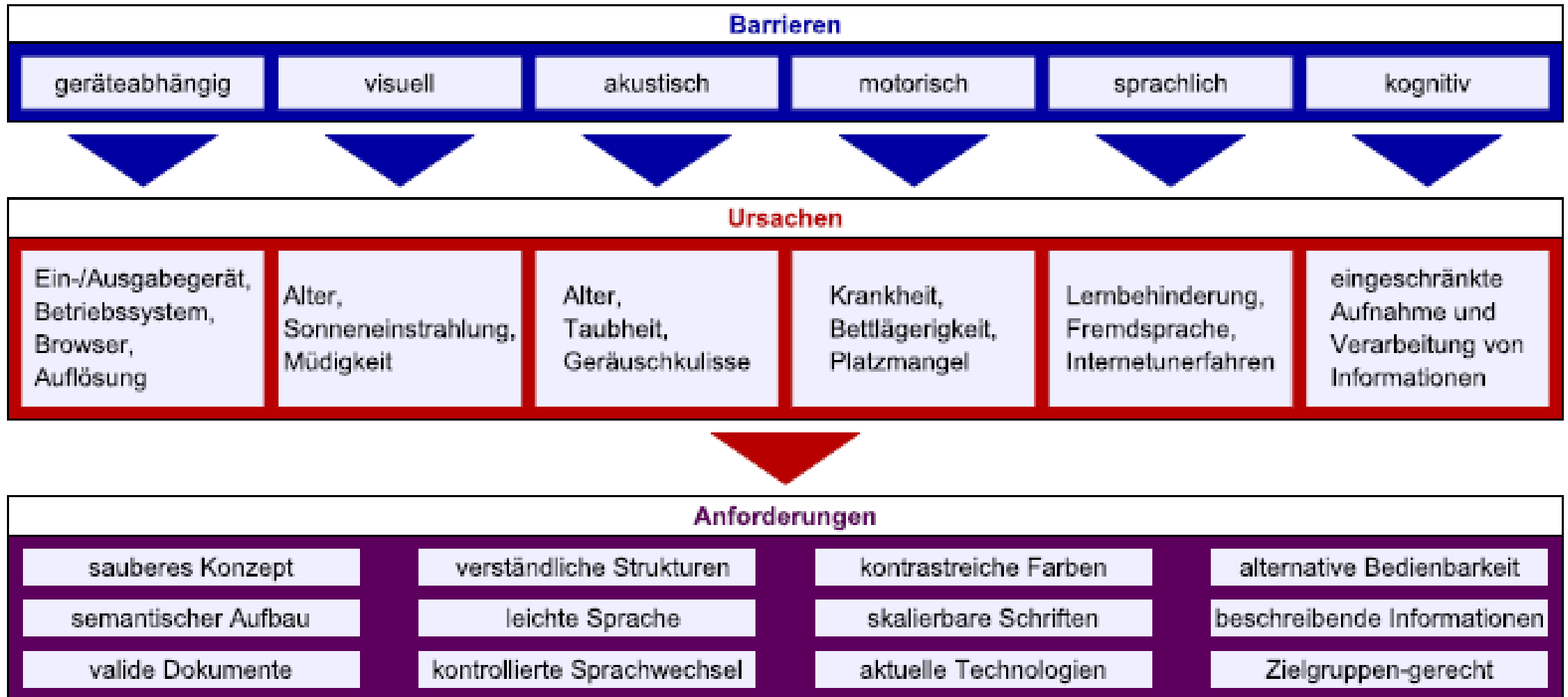
Sie strebt nach der idealen Strukturierung von Informationen, um eine effiziente Nutzung von Informationen und Technologie überhaupt erst zu ermöglichen. Eine Anwendung sollte sich möglichst an feste und etablierte Standards halten und der Erwartung eines Benutzers entsprechen. Das Motto sollte sein, mit möglichst wenigen Schritten gezielt, standardisiert und selbsterklärend eine Information zu finden oder eine Aktion durchzuführen. Es sollte klar sein, dass eine Optimierung nach Usability-Kriterien nicht unbedingt zur Verbesserung der Accessibility führt.

Wer braucht's?

Barrierefreiheit geht uns alle an

Hier eine kleine Auswahl an Zielgruppen:

- Nutzer mit leerem Funkmaus-Akku
- Gruppe der 50+
- Menschen mit einer Sehnenscheidenentzündung im Handgelenk
- Blinde Nutzer
- Nutzer mit einer Rot-Grün-Sehschwäche
- Nutzer mit langsamer Internetverbindung
- Anwender mit Sehbehinderung oder Blindheit



<http://www.die-barrierefreie-website.de/barrierefrei/barrierefreiheitsfaktoren.png>

Behindertengleichstellungsgesetz (BGG)

2002 in Deutschland in Kraft getreten

Erste große Meilenstein zur Gleichstellung behinderter Menschen

- Benachteiligungsverbot behinderter Menschen für öffentliche Einrichtungen
- Gewährleistung von Barrierefreiheit in den Bereichen Verkehr und Bau
- Recht auf Verwendung von Gebärdensprache und anderen Kommunikationsmöglichkeiten abseits des Standards
- Bestimmung für die Gestaltung von öffentlichen Formularen und Vordrucken
- Bestimmungen für barrierefreie Informationstechnik

Barrierefreie Informationstechnik-Verordnung (BITV)

Ergänzung zum Behindertengleichstellungsgesetz (BGG), die ebenfalls im Jahr 2002 in Kraft getreten ist

Sie definiert genaue Kriterien, nach denen man eine Anwendung beurteilen und nach Gesichtspunkten der Barrierefreiheit in Hinblick auf das BGG bewerten kann.

BITV hat keinen Einfluss auf private oder kommerzielle Webangebote. Gerade kommerzielle Anbieter von Webinhalten sind jedoch dazu aufgerufen, ihre Angebote weitestgehend barrierefrei zu gestalten. Dabei können sie sich entweder an der nationalen BITV oder an den internationalen Web Content Accessibility Guidelines (WCAG) des W3C orientieren.

Die BITV entstammt dem Kontext der Web Content Accessibility Guidelines (WCAG) des W3C. Das sind Richtlinien, die Barrierefreiheit in Webapplikationen auf internationaler Ebene definieren.

Die BITV definiert im Wesentlichen folgende Punkte:

- Bereitstellung von Alternativen zu Audio- und visuellen Inhalten
- Struktur/Semantik nicht nur durch Farben darstellen
- Markup und Style Sheets korrekt einsetzen und verwenden
- W3C-Techniken und -Richtlinien einhalten
- Klare Navigationsmechanismen
- Einfach gehaltene Inhalts- und Satzstrukturen
- Geräteunabhängiges Design wählen
- Möglichst auf dynamische Layouts setzen

Web Content Accessibility Guidelines (WCAG)

Die WCAG sind Richtlinien und Empfehlungen für barrierefreie Webanwendungen, die von der W3C ausgearbeitet und zur Verfügung gestellt wurden.

Die WCAG bilden in vielen Ländern die Grundlage für nationale Gesetze und Verordnungen und werden darüber hinaus auch von Unternehmen als Vorgabe für eine Optimierung der Barrierefreiheit übernommen. Da das W3C sich zudem auch mit der Standardisierung von HTML Markup befasst, ist es sinnvoll, auch die Definitionen von Standards in Bezug auf die Barrierefreiheit im Web in die Hände des W3C zu legen.

Der wesentliche Unterschied zwischen der BITV und der WCAG liegt in der Detaillierung der Vorgaben. Die BITV spezifiziert nicht im Detail, wie HTML Markup in barrierefreien Webanwendungen zu verwenden ist, sondern beschreibt vielmehr, welches Ziel erreicht werden muss. Technisch detaillierter sind da die Definitionen der WCAG.

Die Kriterien sind in drei Prioritätsstufen untergliedert. Die Kriterien, welche mit Priorität 1 ausgezeichnet sind, sind in jedem Fall zu erfüllen, sonst gilt das Angebot nicht als barrierefrei bzw. barrierearm. Die Kriterien der Priorität 2 sind als Soll anzusehen. Sie sind zwar nicht zwingend vorgeschrieben, gehören aber in einer Anwendung, die ernsthaft Wert auf Barrierefreiheit legt, umgesetzt. Zu guter Letzt sind noch die Priorität-3-Kriterien, die als Kann einzustufen sind. Erfüllt man auch diese Kriterien mit seinem Angebot, kann man das durchaus als vorbildlich betrachten.

Von der WCAG gibt es sogenannte Konformitätsbedingungen. Die Stufen A, AA und AAA sind zu handhaben wie: muss, soll, kann.

Stufe A ist die minimale Konformitätsstufe die jede Webseite erfüllen sollte

Hier der Link zu den Richtlinien. In den klammern steht jeweils die Stufe, in der die Richtlinie angesiedelt ist:

<https://www.w3.org/Translations/WCAG20-de/WCAG20-de-20091029/#guidelines>

Europäische Union

Die europäische Union hat 1999 die so genannte eEurope-Initiative ins Leben gerufen, die verschiedene Ziele zur Informationsgesellschaft definiert hat. Unter anderem hat sie dafür Sorge getragen, dass die Richtlinien der WAI bis 2002 in allen öffentlichen Einrichtungen umgesetzt wurden. Sie ruft die europäischen Mitgliedsstaaten dazu auf, sich für die Umsetzung der Leitlinien einzusetzen.

Einen festen Kriterienkatalog, nach dem ein Angebot als barrierefrei beurteilt werden kann, gibt es bisher noch nicht. Die Länder sind angehalten, nach den bisher bekannten Regelungen und Normen zu agieren und ihre Angebote auf Barrierefreiheit zu optimieren

Österreich

Auch in Österreich ist in der Bundesverfassung festgeschrieben, dass nach dem Gleichheitsgrundsatz ein ausdrückliches Diskriminierungsverbot für behinderte Menschen gilt. Der Bund, die Länder und die Kommunen bekennen sich zur Gleichstellung behinderter und nicht behinderter Menschen und verpflichten sich zur Gestaltung barrierefreier digitaler Informationen.

In § 6 des österreichischen Behindertengleichstellungsgesetzes wird definiert, dass eine Einrichtung barrierefrei ist, wenn sie für jedermann in der allgemein üblichen Weise benutzt und ohne fremde Hilfe erreicht werden kann. Solange sich die Europäische Union nicht auf verbindliche Normen geeinigt hat, gelten die Richtlinien der WCAG 1.0 des W3C

Webstandard des W3C

Das W3C (World Wide Web Consortium) verbindet man eigentlich sofort mit der Standardisierung von HTML Markup. Das trifft zunächst voll und ganz zu, ist aber nicht die einzige Aufgabe dieses Gremiums.

Das W3C wurde Ende des Jahres 1994 gegründet, als der Ruf nach einem einheitlichen Standard für das Schreiben von HTML immer lauter wurde. Ganzheitliche komplexe Anwendungen waren damals zwar noch nicht wirklich ein Thema, dafür aber die großen Unterschiede in der Darstellung der Webseiten auf verschiedenen Browsern. Viele Hersteller schwammen an der Welle des Interneterfolgs mit, und so entstanden nahezu zeitgleich eine Handvoll Browser, die sich weltweit großer Beliebtheit erfreuten. Das große Problem an dieser Entwicklung war jedoch, dass jeder Browserhersteller sein eigenes Süppchen in Sachen HTML-Markup-Interpretation kochte, denn einen Standard hierfür gab es bis dahin nicht. Das hatte zur Folge, dass Webseiten in verschiedenen Browsern unterschiedlich dargestellt wurden.

Nun kommt das W3C rund um Tim Berners-Lee, der als Erfinder des World Wide Web gilt, ins Spiel – ein unabhängiges und von allen Parteien respektierte Gremium, das 1994 auf den Plan getreten ist. Bis heute ist das W3C das Maß aller Dinge in Sachen HTML Markup Definition und Standardisierung. Das Konsortium definiert seither die Darstellung und den Verwendungszweck von HTML und hat maßgeblich dazu beigetragen, dass Webseiten der heutigen Zeit in allen gängigen Internetbrowsern gleich dargestellt werden.

HINWEIS: Das W3C beschäftigt sich neben der Standardisierung von HTML auch mit der Definition von Regeln für XML, XSLT, CSS, XPATH u.v.m. Auch im Bereich CSS waren bis vor einigen Jahren die Darstellungsunterschiede in den Browsern gravierend. Mittlerweile halten sich die Hersteller auch in diesem Bereich größtenteils an die Konventionen des W3C, was dem Webentwickler von heute die Arbeit deutlich erleichtert.

Es ist sinnvoll, sich als Webentwickler oder -designer immer wieder vor Augen zur führen, mit welchen Hilfsmitteln eine Webanwendung verwendet werden kann und welche technischen Möglichkeiten dem Anwender zur Unterstützung bereitstehen.

Am häufigsten finden folgende Hilfsmittel Verwendung:

- Vergrößerungssoftware (Bildschirmlupe)
- Screen Reader und Braille-Zeile
- Sprachsteuerung
- Spezialtastaturen (besonders große/kleine Tasten, Sondertasten)
- Trackball-Mäuse
- Touchscreens
- Elektronische Steuerungen (über Auge- /Kopfbewegungen, Infrarot)
- Blasrohr zur Steuerung über den Atem (saugen, blasen)

Gute Planung erspart viel Arbeit

Meist gilt in Projekten der Grundsatz: Je besser die Planung, desto effizienter der Weg zu einem guten Ergebnis. Gerade in Bezug auf Barrierefreiheit ist es wichtig, sich bereits im Vorfeld der Umsetzung einer Webanwendung Gedanken über mögliche Hürden und Barrieren zu machen.

Oftmals werden Webanwendungen entwickelt, ohne im Vorfeld die Barrierefreiheit in die Planung oder Umsetzung mit einfließen zu lassen. Die hierfür vorgetragenen Gründe sind meist ein knappes Zeit- oder Finanzbudget. Nach Abschluss eines Projekts und der Freigabe der Anwendung an den Endbenutzer kommen meist nachgelagerte Barrierefreiheitsmaßnahmen auf die Entwickler zu.

Dateibeziehungen

Das `<link>`-Element dient der Abbildung von Dokumenten- und Dateirelationen. Das Element wird im Kopfbereich einer Seite deklariert und kann nicht nur für Layoutdateien verwendet werden, sondern verbindet auch inhaltlich zugehörige Dateien miteinander. Mit dem `rel`-Attribut werden die jeweiligen Relationen einer Seite zu anderen Dokumenten abgebildet. Für viele optische Browser stehe ein Add-on zur Verfügung, um Dokumentenrelationen sichtbar und verwendbar zu machen. Aber selbst ohne ein solches Plug-in ist der Aufwand, der durch die Implementierung der Relationen entsteht, keinesfalls umsonst. Gerade der Screen Reader oder auch eine Suchmaschine arbeiten aktiv mit derartigen Seitenverknüpfungen. Blinden Anwendern wird die Navigation durch eine Webanwendung so deutlich vereinfacht und ein weiterer Schritt in Richtung Barrierefreiheit ist getan.

accesskey und tabindex

Es gilt, Alternativen zu schaffen, um mit der Tastatur ebenso komfortabel wie mit der Maus navigieren zu können. Hierzu gibt HTML5 schon die richtigen Werkzeuge mit auf den Weg

tabindex

Der einfachste Weg, einen Hyperlink mit der Tastatur auszuwählen, ist die Tabulatortaste. Drückt man sie, wird der nächste Hyperlink der Webseite ab der aktuellen Position des Cursors angesprungen. Hierbei gilt, dass der nächste Hyperlink derjenige ist, der im Quellcode, als im HTML Markup, als Nächstes folgt, und nicht der nächste optische folgende Hyperlink. Das kann oftmals zur Verwirrung führen, weil die Reihenfolge der Tabulatorsprünge optisch nicht ganz logisch ist.

Diese Problem lässt sich mit einer manuell definierten Tabulatorreihenfolge lösen. Hierfür stellt HTML5 das tabindex-Attribut zur Verfügung. Diesem Attribut kann eine Zahl als Wert zugewiesen werden, die die Reihenfolge der fokussierten Elemente beim Drücken der Tabulatortaste bestimmt

```
<a href="index.html" tabindex="1">Startseite</a>  
<a href="impressum.html" tabindex="3">Impressum</a>  
<a href="produkte.html" tabindex="2">Produkte</a>
```

Ohne das tabindex-Attribut würden die Links jeweils in der optischen Reihenfolge fokussiert. Das tabindex-Attribut wird von allen marktgängigen Browsern unterstützt

Der Einsatz einer Tabulator Reihenfolge ist aus Sicht der Barrierefreiheit unbedingt zu empfehlen, wenn die Reihenfolge von Sprungpunkten optisch von der im Quellcode definierten Reihenfolge abweicht. Entscheidend ist, dass optisch anklickbare Elemente in einer sinnvollen und aufeinanderfolgenden Abfolge angesprungen werden

Hinweis: Seit HTML5 kann das tabindex-Attribut auf alle Elemente angewendet werden.

Achtung: Elemente mit einem JavaScript Event Handler, die lediglich mit der Maus anwählbar sind, sind keinesfalls barrierefrei. Sie sollten vermieden werden. Diese Elemente würden auch nicht durch die Tabulatortaste erfasst.

accesskey und tabindex

accesskey

Einen Schritt weiter als das tabindex-Attribut geht man mit dem accesskey. Hiermit lässt sich einem HTML-Element eine Zugriffstaste zuweisen, mit der es dann später über die Tastatur angesteuert werden kann.

```
<a href="index.html" accesskey="s">Startseite</a>  
<a href="impressum.html" accesskey="i">Impressum</a>  
<a href="produkte.html" accesskey="p">Produkte</a>
```

Die meisten Browser erlauben eine Ansteuerung einer Zugriffstaste über die Tastenkombination Alt + Shift + Zugriffstaste. Wenn eine Taste mehrfach vergeben wird, so wird beim Drücken der entsprechenden Tastenkombination der Hyperlink nicht mehr sofort ausgeführt, sondern nur noch fokussiert und im Fall einer Screen-Reader-Anwendung vorgelesen. Drückt man die Zugriffstaste erneut, springt der Cursor weiter zum nächsten Element, das den gleichen Wert aufweist.

In Hinblick auf die Barrierefreiheit einer Anwendung ist es zu empfehlen, Zugriffstasten dosiert und sinnvoll einzusetzen. Die Navigation einer Webanwendung sowie alle aktionsauslösenden Hyperlinks und Schaltflächen sind mit einer Zugriffstaste auszustatten.

Hinweis: Seit HTML5 kann das tabindex-Attribut auf alle Elemente angewendet werden.

Anwendungsstruktur

Die technische Planung und die Konzeption einer Anwendungsstruktur.

Die Suche und auch die Navigation haben meist eine wichtige Bedeutung, weshalb sie im Kopfbereich der Anwendung platziert werden. Der Grund hierfür ist, dass Nutzer eines Screen Readers beim Öffnen einer Seite zuerst im Suchfeld und beim Weiterlesen in der Navigation der Anwendung landet. Wichtig hierbei ist eigentlich nicht die optische Platzierung dieser Module, sondern die Stelle, an der die Suche und die Navigation im Markup verankert sind, denn nur daran orientiert sich der Screen Reader.

Außerdem werden sie vom Anwender eines visuellen Browsers am ehesten dort erwartet.

Layout und Struktur

Tabless Layout

Element	Beschreibung
<header>	Definiert den Kopfbereich einer Seite oder einer Sektion
<nav>	Bereich für eine Navigation
<aside>	Definiert einen Bereich neben dem eigentlichen Seiteninhalt
<section>	Definiert eine Sektion in einer Webseite
<article>	Definiert einen Artikel
<footer>	Definiert den Fußbereich einer Seite oder einer Sektion

```
<!DOCTYPE html>
<html>
  <head>
    <title>Barrierefreiheit geht alle an</title>
  </head>
  <body>
    <header id="header">
      <div id="logo">
        Logo
      </div>

      <div id="search">
        Suche
      </div>

      <nav id="navigation">
        Navigation
      </nav>
    </header>
```

```
    <aside id="left">
      Links
    </aside>

    <aside id="right">
      Rechts
    </aside>

    <section id="main">
      <nav id="breadcrumb">
        Navigationspfad
      </nav>
      <article id="content">
        Inhalt
      </article>
    </section>
    <footer id="footer">
      Fußzeile
    </footer>
  </body>
</html>
```

Farben und Kontraste

Barrierefreies Webdesign wird durch viele Faktoren beeinflusst. Neben technischen Herausforderungen und strukturellen Kriterien spielt auch eine bedachte Farbwahl eine wichtige Rolle. Die Wahl der in einer Webanwendung zum Einsatz kommenden Farben und Kontraste stellt einen wichtigen Schritt im Designprozess dar. Der erste Schritt auf dem Weg zum Webdesign ist die Auswahl einer oder mehrerer Farben, die sich wie ein roter Faden durch die komplette Anwendung zeihen sollen. Leider wird beim Entwurf eines Designs oftmals vergessen, dass es Farbkombinationen gibt, die für Menschen mit einer Sehschwäche für Farben schnell zum Problem werden können. Des Weiteren sollte das Thema Kontraste im Hinblick auf Barrierefreiheit genau betrachtet werden. Oftmals stehen Hintergrund und Text einer Anwendung nicht in einem ausreichenden Kontrastverhältnis zueinander und können von Anwendern mit Sehschwäche nicht erkannt bzw. voneinander getrennt werden.

Helligkeit und Farbwerte zwischen Hintergrund und Vordergrund muss so gewählt werden, dass ein ausreichender Kontrast entsteht und die Inhalte gut lesbar sind.

Der Color Contrast Analyser (<https://developer.paciellogroup.com/resources/contrastanalyser/>) der Paciello Group ist ein kostenfreies Tool und steht frei zum Download zur Verfügung. Mit diesem Werkzeug lässt sich beispielsweise das Kontrastverhältnis zwischen Vorder- und Hintergrund auf Barrierefreiheit hinsichtlich verschiedener Farbsehschwächen überprüfen. Ein weiteres Feature ist das Simulationsfenster, mit dem Sie eine Webanwendung wie durch eine Brille anschauen und verschiedene Sehschwächen simulieren können.

Die Navigation bietet stets einen Überblick über die zur Verfügung stehenden Inhalte und Funktionen einer Anwendung. Neben der eigentlichen Aufgabe als Sprungbrett zu den einzelnen Bereichen der Software bietet sie zudem auch die Funktion eines Inhaltsverzeichnisses. Dieser Gedanke sollte nicht aus den Augen verloren werden. Eine Navigation zu splitten und auf einer Seite aufzuteilen ist aus Usability Sicht keine gute Idee. Es sollte eine zentrale Navigation geben als zentralen Anlaufpunkt zum Manövrieren durch die Anwendung. Menschen mit Sehbehinderung haben einen Fixpunkt, zu dem sie zurückkehren können, wenn sie zu anderen Inhalten springen möchten.

Aus Sicht der Barrierefreiheit unterscheidet man zwischen zwei Navigationsarten, der Seiten- und der Inhaltsnavigation.

Die Seitennavigation dient als Sprungbrett zu den einzelnen Inhalten einer Webanwendung. Sie sollte an einem zentralen Fixpunkt in der Anwendung verankert und dort auch immer aufzufinden sein. Das dient sowohl der Barrierefreiheit als auch der Benutzerfreundlichkeit. Die Seitennavigation wird vom Benutzer meist in der Kopfzeile direkt über dem eigentlichen Inhalt oder am linken Bildschirmrand erwartet.

Die Inhaltsnavigation unterstützt Nutzer eines Screen Readers, verschiedene Bereiche der aktuell geöffneten Webseite anzusteuern und Inhalte zu überspringen. Hierbei handelt es sich um ein weiteres Navigationsmenü, das in der Regel für den Anwender eines optischen Browsers außerhalb des sichtbaren Bereichs liegt. Für stark sehbehinderte Nutzer ist es von Vorteil, dass das Navigationsmenü bei Bedarf auch sichtbar werden sollte und prinzipiell von jedem Anwender genutzt werden kann. Hier kann mit JavaScript gearbeitet werden.

Die Inhaltsnavigation wird wie die Seitennavigation auch so weit oben wie möglich im HTML-Kontext implementiert. In diesem Menü kommen vor allem Sprungmarken zum Einsatz.

Navigationspfad

Der aktuelle Standort des Benutzers in einer Anwendung wird durch Hervorheben des aktiven Menüpunkts oder durch eine Überschrift verdeutlicht und angezeigt. Neben diesen visuellen Hinweisen über den aktuellen Standort sollte eine Webanwendung zudem einen Navigationspfad, auch bekannt unter der Bezeichnung Breadcrumb Navigation, auf der Webseite platziert werden.

Einsatz findet ein Navigationspfad in umfangreichen Webprojekten mit mehreren Navigationsebenen, die nicht alle von der Startseite zu erreichen sind. Die Breadcrumb Navigation unterstützt den Anwender dabei, ergänzend zur Seitennavigation, schnell und komfortabel zurück zu einer übergeordneten Ebene zu springen.

Platziert wird der Navigationspfad weit oben auf der Webseite. Optimal ist ein Platz direkt über dem eigentlichen Inhalt der Webseite. Getrennt werden die einzelnen Hierarchiepunkte häufig mit dem Zeichen für „größer als“ (>). Der letzte Punkt im Navigationspfad ist immer die aktuelle Seite. Dieser Eintrag ist als einziger nicht mit einem Hyperlink hinterlegt. Alle anderen Einträge verlinken auf die jeweilige Webseite der Ebene

Hinweis: Die Bezeichnung Breadcrumb Navigation (Brotkrumenpfad) entstammt dem Märchen „Hänsel und Gretel“, in dem die Protagonisten im Wald Brotkrumen verstreuten, um später wieder den Weg nach Hause zu finden.

Weitere Navigationshilfen

Ist nicht zwingend notwendig. Es handelt sich eher um eine Empfehlung bzw.. Um das Aufzeigen der Möglichkeiten, die man zur Verfügung hat, um die Navigation für einen Benutzer so bequem und einfach wie möglich zu gestalten.

Sitemap

Eine Sitemap stellt eine Übersichtsseite dar, die mit einem Inhaltsverzeichnis vergleichbar ist. Sie enthält Verweise auf jede einzelne Seite der Webanwendung, die mit einfachen Hyperlinks realisiert ist. Auch für Suchmaschinen ist solch eine Seite von großer Bedeutung, da sie Aufschluss über den kompletten Umfang des Angebots gibt.

Empfehlenswert ist es, die Hierarchieebenen der einzelnen Webseiten auch visuell sichtbar zu machen und mit verschachtelten Listen zu arbeiten.

Index/Glossar

Ein Index bzw. Ein Glossar ist in sehr großen Webanwendungen mit hohem Informationsgehalt sinnvoll. Hierbei handelt es sich um ein alphabetisch geordnetes Stichwortverzeichnis, das Verweise auf die entsprechenden Webseiten enthält, in denen das Stichwort thematisiert wird. Bei einer Webseite mit vielen verschiedenen Themengebieten bietet ein solches Stichwortverzeichnis eine gute Orientierungsfunktion

Kontextbasierte Hilfe

Eine Webanwendung, mit der viel interagiert werden kann, sollte kontextbasierte Hilfeseiten anbieten. Der Benutzer kann sich über die Hilfe informieren, welche Hilfsmittel von der Anwendung angeboten werden.

Generell ist es empfehlenswert, eine allgemeine Orientierungshilfe und Hinweise auf Unterstützung von besonderen Hilfsmitteln im Hinblick auf Barrierefreiheit anzubieten. Auch eine Möglichkeit zum Senden eines Feedbacks zum Thema Barrierefreiheit ist von Vorteil. Bei sehr langen Hilfeseiten bietet es sich an, die einzelnen Kapitel mit einer Sprungmarke zu versehen oder den Inhalt auf mehreren Seiten zu verteilen

Frequently Asked Questions

Häufig gestellte Fragen, das so genannte FAQ, ist eine Sammlung von Fragen, die immer wieder in Bezug auf eine Webanwendung auftauchen. Um sich und in erster Linie dem Benutzer das Schreiben einer Mail zu sparen, bietet das FAQ eine Möglichkeit, häufig gestellte Fragen mit passenden Antworten bereits auf einer Webseite zu veröffentlichen.

Ebenfalls fällt die lange Ladezeit einer Webanwendung unter Barrierefreiheit. Zahlreiche potenzielle Anwender surfen mit einer sehr langsamen Internetverbindung durchs Netz.

Caching

Das Prinzip des Cachings, also das Zwischenspeichern von wiederkehrenden Inhalten in einem Puffer, ist nahezu genauso alt wie die ersten Webanwendungen. Es wird unterschieden zwischen dem clientseitigen und dem serverseitigen Caching.

Serverseitiges Caching

Es handelt sich beim Konzept des serverseitigen Cachings um einen Puffer auf Seiten des Servers.

Clientseitiges Caching

Hierbei können wir je nach Aufbau der IT-Infrastruktur noch einmal zwischen einem Proxy- und einem Client-Cache unterscheiden.

In vielen Unternehmen kommt ein Proxy Server zur Zugriffsteuerung der einzelnen angebundenen Clients auf das Internet zum Einsatz. Er ist zwischen den einzelnen Clients und dem Internet platziert.

Der Webbrowser, über den der Anwender eine Applikation aufruft und öffnet, besitzt ebenfalls einen Zwischenspeicher für Webinhalte. Der Cache des Browsers puffert den Inhalt von Websites und beschleunigt so beim erneuten Aufruf der gleichen Webseite die Ladezeit enorm.

Gerade der Proxy- als auch der Browser-Cache kann schnell zum Problem werden. Die Praxis zeigt immer wieder veraltete, zwischengespeicherte Stände von Webseiten in einem der beiden Puffer. Die Lösungen sind spezielle Metatags, mit denen man das Caching auf der Clientseite ein wenig steuern kann.

Ladezeiten und Cache

Clientseitiges Caching

```
<!DOCTYPE html>
<html>
  <head>
    <title>Barrierefreiheit geht alle an</title>

    <meta http-equiv="cache-control" content="no-cache" />
    <meta http-equiv="pragma" content="no-cache" />

  </head>
  <body>
  </body>
</html>
```

Es gibt Browser, die nur eine der beiden Anweisungen interpretieren können. Eine Möglichkeit, die Lebensdauer von Inhalten im Puffer zu begrenzen, gibt es leider nicht.

Caching externer Inhalte

Gesondert zu betrachten sind Inhalte und Ressourcen, die in eine Webseite inkludiert werden, wie zum Beispiel JavaScript- und CSS-Dateien sowie Bilder und andere Medien. Geschickt ist es, CSS und JS in externe Dateien auszulagern und dann in eine Webseite einzubinden. Somit erzielt man erstens eine Trennung zwischen dem eigentlichen HTML Markup und anderen Erweiterungen, und zweitens werden diese extern eingebundenen Dateien separat vom Browser oder Proxy gepuffert.

Jedes Zeichen, das in einer Datei enthalten ist, verbraucht Speicherplatz. Dazu gehören auch Leerzeichen, Zeilenumbrüche und Tabs. Zum Komprimieren von CSS und JS Files gibt es kostenlose Helfer. Es hat sich außerdem etabliert, komprimierte Dateien mit einem Kürzel zu versehen., (stylesheet.css => stylesheet.min.css)

Welche Auswirkungen es auf den PageSpeed hat und warum eine optimierte PageSpeed wichtig ist erfahrt ihr hier:
<https://seo-ranking-tools.de/seo-guide/onpage/pagespeed/>

Pop-ups

Ein kontrovers diskutiertes Thema ist die Nutzung von Pop-up-Fenstern in einer Webanwendung.

<https://www.barrierefreies-webdesign.de/knowhow/pop-up-fenster/sinn-und-unsinn.html>

Barrierefrei texten

Unter den Begriff der Barrierefreiheit fallen weit mehr als nur die technische Optimierung von Webanwendungen und die Gewährleistung auf Zugänglichkeit mittels verschiedener Hilfsmittel. Barrieren können auch gerade bei Texten entstehen, die unter den Gesichtspunkten der Barrierefreiheit nicht optimiert wurden.

Tipp: Einfach geschriebene Texte können neben Menschen mit Behinderung auch leichter von Suchmaschinen interpretiert und klassifiziert werden. Das führt nicht selten zu einem höheren Ranking bei einem Suchergebnis in einer Suchmaschine

KVA – Kurz, verständlich, ansprechend

Schlicht und einfach geschriebene Texte, die alles Wissenswerte enthalten, sind dienlicher als kompliziert formulierte Fassungen einfacher Sachverhalte.

Texter sollten versuchen folgende Dinge zu vermeiden:

- Fremdwörter
- Redundanzen
- Überlange zusammengesetzte Wörter
- Überlange Schachtelsätze
- Abkürzungen
- Füllwörter
- Komplizierte Metaphern und abstrakte Begriffe

Das KISS-Prinzip (Keep it short and simple) definiert noch einmal kurz und prägnant, dass Text im Web in jedem Fall kürzer ausfallen sollen als die auf Papier geschriebene Version. Je kürzer, desto besser.

Die vier Merkmale der Verständlichkeit

Der Inhalt wird nach den Kriterien Einfachheit, Gliederung, Prägnanz und Anregung bewertet.

Details zu den vier Merkmalen:

<https://www.barrierefreies-webdesign.de/knowhow/verstaendlicher-text/vier-merkmale-der-verstaendlichkeit.html>

Formatierung und Darstellung

Einen überlangen Fließtext möchte niemand lesen, schon gar nicht am Bildschirm oder auf einem mobilen Gerät. Das menschliche Auge braucht Sprungmarken, Fixpunkte, an denen es sich orientieren kann. Dazu verwendet man: Absätze, Überschriften und Bilder oder Markierungen mit Unterstreichung, Kursiv- und Fettschrift

Schriften

LESERLICHE Schriften nutzen und nicht zu viele mischen!!

Weniger ist mehr!

Der BITV-Test ist ein Prüfverfahren für die umfassende und zuverlässige Prüfung der Barrierefreiheit von informationsorientierten Webangeboten.

<https://bitvtest.de/start.html>

Tools für den Selbsttest

Lynx

Ist ein textbasierter Webbrowser, der eine Webanwendung in reiner Textform darstellt. Ein unverzichtbares Tool, wenn man sein Angebot für den Screen Reader optimieren möchte. Lynx stellt seine Anwendung so dar, wie sie auch ein Screen Reader darstellen und interpretieren würde.

Lynx Viewer: <http://delorie.com/web/lynxview.html>

Browser: <https://lynx.browser.org/>

Kompliziert zu installieren!

JAWS

Job Access With Speech (JAWS) ist der wohl bekannteste Screen Reader und kommt bei einem Großteil der sehbehinderten Webbenutzer zum Einsatz. Das Programm ist in der Lage, eine Webanwendung in reiner Textform anzuzeigen und darüber hinaus die Inhalte vorzulesen.

Eine Alternative zur JAWS-Installation ist ein kostenloses Firefox-Plugin mit dem Namen Fangs. Hierbei handelt es sich um einen Screen-Reader-Emulator (Stand März 2019: It no longer works on current versions of Firefox.)

Colorblind Webpage Filter

<https://www.toptal.com/designers/colorfilter>

Der Colorblind Webpage Filter ist ein Onlinetool, mit dem man Webanwendungen so anzeigen kann, wie sie von einem Farbenblinden wahrgenommen werden. Das Tool simuliert verschiedene Farbfehlsichtigkeiten und bietet darüber hinaus die Möglichkeit, das Angebot in Graustufen darzustellen

Accessibility Evaluation Toolbar

Hierbei handelt es sich um eine Toolbar. Sie stellt zahlreiche Hilfsmittel bereit, um eine Webanwendung auf Barrierefreiheit zu analysieren.

Die Erweiterung gibt es für Firefox und Chrome

<https://wave.webaim.org/extension/>

W3C Validator

<https://validator.w3.org/>

<https://jigsaw.w3.org/css-validator/>

Die W3C stellt einen Validator zur Prüfung von CSS-Code und Markup zur Verfügung.



CODERS.BAY

QUELLE:

TIMM BREMUS - BARRIEREFREIHEIT

ISBN: 978-3-86802-095-3

DIV. INTERNETSEITEN SIEHE EINZELNE FOLIEN