# Hochschule der Medien Digitale Inklusion - Eine Win-Win-Lösung für Unternehmen und Kunden

## Online-Workshop: Easy Web Check – Pirmin Gersbacher Redigierte Live-Mitschrift 13.03.2024 Vormittag

Der vorliegende Text ist die redigierte Version der Live-Mitschrift.

© 2024 Delfinterpret GbR

(Gottfried Zimmermann) Wir kommen zum nächsten Programmpunkt: Jetzt geht es darum, weshalb ihr heute wahrscheinlich da seid. Einfaches Testen von Webseiten für Laien. Pirmin Gersbacher, mein Mitarbeiter vom Kompetenzzentrum an der Hochschule der Medien, wird den Prozess Easy Web Check vorstellen.

Die Idee ist, dass man eine Möglichkeit hat, auch als Laie/Laiin eine Webseite auf Barrierefreiheit zu prüfen. Es ist nicht gut, wenn man bei so einem Projekt bis zum Ende, und am Ende lässt man eine Webseite prüfen, wenn schon alles komplett ist. Dann bekommt man das Ergebnis, dass es noch viele Fehler gibt und man müsste eigentlich von vorne beginnen.

Die Idee ist, dass man auch als Laienperson vorab Fehler finden und kommunizieren kann. Man findet vielleicht die schlimmsten Fehler, wenn auch nicht alle. Aber die häufigsten, die wahrscheinlich auch am wichtigsten sind. Auch beim Einkauf, wenn man selbst nicht Webseiten programmiert oder designt, kann man trotzdem die Tools einsetzen, die uns Pirmin gleich vorstellt, um erstmal einen Eindruck zu bekommen, welche Probleme es eigentlich gibt. Pirmin wird auch dazu sagen, inwieweit alle Anforderungen abgedeckt sind. BITV-Test ist unser Maßstab. Damit können wir nicht alles, aber Wichtiges abdecken, ohne ein Experte oder eine Expertin zu sein.

Bevor wir an Pirmin weitergeben, kommt jetzt noch Alexander Pfingstl von der Bundesüberwachungsstelle, er sagt noch etwas dazu. Bitte, Alexander! Willst du rüberkommen? Setz dich einfach hier her auf meinen Stuhl. Dann kann man dich sehen.

(Alexander Pfingstl) Ein herzliches guten Morgen auch von mir. Schön, dass so viele Interessierte da sind. Das Thema Barrierefreiheitstests - auch gerade für Laien - ist ein großes und wichtiges Thema. Wir sind die Bundesüberwachungsstelle. Wir dürfen und müssen seit 2020 öffentliche Seiten des Bundes überprüfen. Die gleichen Stellen gibt es in jedem Bundesland. Wir bzw. unsere Länderkollegen bieten auch immer Prüfungsberatung für die Ergebnisse an. Ein großer Punkt ist: Wie kann ich selbst etwas prüfen?

Was Gottfried gesagt hat, ist sehr richtig. Viele beauftragen etwas, sie geben sich große Mühe und stellen hinterher fest: Das ist schwierig. Dann wundern sie sich, wie das passieren konnte.

Wir haben uns überlegt, dass man sich zwei Arten von Fehlern überlegen kann, inhaltliche und technische Fehler. Gerade die inhaltlichen Fehler lassen sich über den Web Check testen. Sie sind aber auch selbst gut nachvollziehbar. Es sind oftmals Geschichten wie: Habe ich Überschriften signalisiert, habe ich die Bilder beschriftet? Tatsächlich sehr einfache Sachen, wo ich als Autor oder Autorin feststellen kann, ob ich es richtig gemacht habe, um einfach anzufangen. Das ist wirklich wichtig, denn Webseiten sind dynamisch, auch wenn man von einer Firma testen lässt oder ob wir sie testen. Das ist immer nur eine Momentaufnahme von einem Tag oder einer Uhrzeit.

Die Webseiten sind so dynamisch. Es kommt jeden Tag etwas dazu. Jeden Tag verschwindet auch Content. Ich muss dauerhaft gewährleisten, dass ich es nicht schlechter mache als es vielleicht vorher schon war. Und es ist gut zu lernen, mit welchen einfachen Möglichkeiten man arbeiten kann, um ein Gefühl für die Barrierefreiheit zu bekommen. Man holt zwischenzeitlich Feedback von Experten ein. Und zwar, ob man sich auf den Ebenen, die man nicht richtig testen kann, in die richtige Richtung bewegt. Über die Jahre sehen wir dann eine Verbesserung, zumindest bei den öffentlichen Stellen. Hier ist das gefordert.

Wir haben dieses Jahr den zweiten EU-Bericht, den wir im Rahmen dieser Initiative machen müssen. Wir müssen über den Fortschritt der Barrierefreiheit berichten. Natürlich trägt gerade das selbst Testen auch dazu bei, dass man selbst auch einen Fortschritt erreichen kann und nicht immer warten muss, bis irgendjemand Zeit hat, die eigene Webseite zu testen. Soweit erstmal von mir und bis später.

(Gottfried Zimmermann) Danke, Alexander. Übrigens: Der Easy Web Check ist ein kostenloser Test und wir bieten auch Schulungen dazu an. Die muss man bezahlen. Aber diese Checklist und den Test kann man kostenlos verwenden. Ich gebe weiter an Pirmin. Du kannst jetzt deinen Bildschirm teilen und uns den Easy Web Check vorstellen.

(Pirmin Gersbacher) Hallo von mir. In diesem Abschnitt geht es darum: Wie kann man einfach Webseiten testen, auch als Laien. Wir haben uns ein Konzept überlegt: Das ist Easy Web Check. Den will ich heute zumindest teilweise vorstellen.

Erstmal zu mir. Mein Name ist Pirmin Gersbacher. Ich bin Anfang 30, habe braune Haare, einen Bart, eine eckige Brille. Meine normale Brille ist gestern kaputtgegangen. Ich habe meinen Master an der HdM in Stuttgart in Medieninformatik gemacht. Seit etwas über zwei Jahre bin ich wissenschaftlicher Mitarbeiter im Kompetenzzentrum in Stuttgart. Dort bin ich auch Tester für Webseiten, also BITV-Prüfer und ich prüfe auch mobile Apps.

Heute möchte ich gerne Easy Web Check vorstellen. Was sind die Lernziele? Wir wollen uns einen Teil von Easy Web Check angucken. Warum einen Teil? Normal geht der Workshop 4 Stunden. Heute kriegen wir das leider nicht unter. Aber es geht uns einfach darum, ein paar wichtige Prüfschritte anzuschauen. Welche Analysetools gibt es? Was können wir da empfehlen? Welche Hilfsmittel gibt es, die man vielleicht noch gar nicht kennt, die einem dabei helfen, so etwas selbst zu testen?

Es ist uns wichtig, die Checkliste zu Easy Web Check zu zeigen. Sie ist ein gutes Hilfsmittel, das ausgearbeitet wurde, um auch selbst Webseiten zu testen.

Der Easy Web Check selbst ist, wie Gottfried schon meinte, eine vereinfachte Prüfmethode. Wir legen den Schwerpunkt auf die low hanging fruits, hier sind keine technischen Kenntnisse im Webdesign oder Programmierkenntnisse erforderlich. Man muss sich auch nicht mit HTML auskennen, oder wissen, wie man den HTML-Code einer Webseite anguckt, sondern einfach, wie man sich das anschauen und dann bewerten kann.

Der komplette Easy Web Check deckt einen Teil, aber nicht alle Kriterien nach BITV 2.0 ab. Konkret sind es 52 von 94. Hier gibt es auch eine Tabelle. Ich würde sie gerade mal zeigen. Es ist eine Excel-Tabelle. Die hat mein Kollege Andreas Burkard entwickelt. Er hat auch Easy Web Check entwickelt.

Hier links, in der ersten Spalte der Tabelle, sieht man die ganzen Tabellen aus den EN 31549, aus der Tabelle A1 - alles, was für Webseiten wichtig ist und gesetzlich abgedeckt werden sollte.

Er hat Easy Web Check genommen und geschaut, was manuell geprüft wird. Das hat er hier gelb eingetragen. Das muss man sich anschauen und mit dem Wissen, das man hat, mit Easy Web Check entscheiden: trifft es zu oder nicht?

Hier gibt es noch Punkte, wo drinsteht, welches Tool man verwenden kann. Teilweise kann man die meisten einzelnen Prüfschritte automatisch prüfen mit den einzelnen Tools. Das schauen wir uns auch nochmal an. Die roten Punkte sind nicht abgedeckt.

Wenn man einmal schaut, direkt am Anfang, da ist ein ziemlich großer roter Block, die Sechserkriterien. Da geht es um Zwei-Wege-Sprachkommunikation. Also: Wenn man ein Zoom, Big Blue Button oder Microsoft Teams entwickelt, worauf muss man dabei achten? Der Gedanke war hier: Es kommt nicht so oft vor. In der Regel hat eine Webseite das nicht. Wenn doch, muss man im Kopf haben: Es wird nicht hier geprüft. Das heißt, man muss noch einmal selbst gucken.

Genauso weiter unten. Bei den Elfer-Prüfschritten, 11.8. Da geht es hauptsächlich um Autorentools. Wenn man selbst ein WordPress oder Moodle entwickelt, muss man gewisse Sachen beachten, damit die Leute, die Inhalte einpflegen, diese barrierefrei machen können. Das ist auch nicht Teil von Easy Web Check. Es ist der Standard für Webseiten, dass man so etwas entwickelt. Die Folien geben wir wahrscheinlich auch noch raus, oder?

(Gottfried Zimmermann) Können wir, ja. Ich schicke gleich noch den Link zur Tabelle. Der war veraltet. Es kommt gleich nach. Aber die Checkliste habt ihr schon.

(Pirmin Gersbacher) Sehr gut. 52 von 94 verpflichtenden Kriterien aus der BITV sind damit abgedeckt. Auch nochmal der Hinweis: Manche Kriterien werden nur unvollständig geprüft. Man kann nicht alle Sonderfälle auf einfache Art und Weise da mit aufnehmen. Aber es ist das Meiste in der Excel vermerkt.

Der Gedanke bei Easy Web Check ist, dass es eine halbautomatische Prüfung ist. Wir setzen zum einen Teil automatische Tools – Browser-Erweiterungen – ein, die wir auf Barrierefreiheit testen. Es sind verschiedene andere Tools. Ich werde gleich zeigen, was man verwenden kann, um einfacher zu prüfen. Zusätzlich ist immer eine manuelle Sichtprüfung nötig.

Heute haben wir uns entschieden, den Fokus auf die automatischen Tools, den ersten Teil, zu setzen: Was kann man prüfen und wie kann man das dann interpretieren, was diese Tools genau anzeigen.

Wichtig wäre hier noch der Link zur Checkliste. Diese ist auch online verfügbar. Easy Web Check wurde von Andreas Burkard und Gottfried Zimmermann entwickelt, eine vereinfachte Prüfmethode, mit der man in kurzer Zeit und mit kostenlosen Werkzeugen Barrieren finden kann. Es sind keine tiefen technischen Kenntnisse erforderlich und es ist kein Ersatz für ausführliche Prüfmethoden. Es geht darum, während des Prozesses, am Anfang etwas selbst testen zu können, auch wenn man kein Entwickler ist oder hineinzuschauen, dass es ein bisschen schneller geht. Ich springe mal ans Ende. Das Dokument ist auch unter CC freigegeben. Sie können es verwenden und selbst benutzen, wie Sie wollen.

Aufgebaut ist es grundsätzlich so: Es gibt immer eine Überschrift. Das ist das Tool, das man verbinden kann. Hier in dem Fall der Siteimprove Accessibility Checker und verschiedene Checkpunkte. Diese sind immer positiv formuliert. Checkpunkt 1 ist beispielsweise: „Bilder haben jeweils einen Alternativtext. Dieser darf auch leer sein, er darf nur nicht komplett fehlen.“

Der Sinn der Checkliste ist: Wir haben hier diesen Punkt, gucken uns die Webseite an, wir verwenden das Tool und unter Verwendung des Tools können wir dann sagen: Der Checkpunkt ist erfüllt oder nicht. Wenn er erfüllt ist, können wir ein Häkchen dranmachen und zum nächsten Punkt gehen.

Wenn man hier ein bisschen runtergeht, hat man auch andere Tools. Zum Beispiel verschiedene Bookmarklets. Da werde ich beispielhaft etwas zeigen – verschiedene Browser-Erweiterungen. Und hier ganz unten, da kommt man nicht darum herum, ist dann die Sichtprüfung.

Das sind Prüfschritte oder Checkpunkte, die man sich durchlesen muss. Wie zum Beispiel Punkt 56: Bewegte Inhalte, zum Beispiel animierte GIFs, die nicht nach fünf Sekunden automatisch stoppen, müssen abschaltbar sein. Man nimmt also die Webseite, guckt sich an, ob es bewegte Inhalte sind wie solche GIFs, stoppen sie nach fünf Sekunden? Und wenn nicht, sind sie abschaltbar? Wenn das der Fall ist, kann man wieder ein Häkchen setzen und hat somit den Punkt von Easy Web Check wieder erfüllt.

Bei einem weiteren wichtigen Punkt, wenn ihr die Checkliste macht, hat sich mein Kollege etwas gedacht: Man sollte wirklich von vorne anfangen und so durchgehen. 1, 2, 3 usw., vor allem, weil die Checkpunkte teilweise aufeinander aufbauen. Am Anfang gucken wir also, was kann man vollautomatisch testen? Wo kann man Tools einsetzen? Dann kommt die Sichtprüfung für den Rest, den man nicht testen konnte.

Das erste Tool, das wir haben, ist der Siteimprove Accessibility Checker. Er ist ein automatisches Prüftool. Automatische Prüftools sind in der Regel Browser-Erweiterungen. Man installiert sie, um zusätzliche Funktionalität in den Browser hinein zu bekommen. Wir haben ein kleines Programm, das man starten kann. Die Tools überprüfen automatisch auf Zugänglichkeitsprobleme. Die haben ihre eigenen Checklisten, die sie abarbeiten. Die gucken für uns in das HTML hinein und listen auf, welche Fehler gefunden wurden. Andere Tools heben es direkt in der Webseite hervor. Wir nutzen den Siteimprove Accessibility Checker. Aber es gibt eine Vielzahl von Tools. Auch Accessibility Testing Tools oder von WAVE oder TOOLKIT, da gibt es verschiedene Vorlieben, je nach Abstraktion, wie das angezeigt wird.

Wir empfehlen und verwenden den Siteimprove Accessibility Checker. Hier direkt der Hinweis. Solche automatischen Prüftools, wie der Accessibility Checker finden im Schnitt weniger als ein Drittel der Barrieren. Es ist sehr positiv ausgedrückt. Die kostenlosen Tools finden eigentlich noch weniger. Mein damaliger Kollege hat es ein bisschen untersucht. Im Prinzip kam er darauf, dass diese teuren, kommerziellen Tools, wo man pro Seite bezahlt, im Schnitt ein Drittel der Barrieren finden konnten. Das ist wichtig im Kopf zu behalten, dass man nicht denkt, man lässt das kostenlose Tool über eine Webseite laufen, und wenn man keine Barrieren gefunden hat, ist alles gut. Das ist nicht so. Es ist eine nützliche Ergänzung zum manuellen Prüfen. Deswegen bietet es sich an, dass wir es für den Einstieg benutzen, um schon mal einen Teil abzudecken.

Dann würden wir einmal den Siteimprove Accessibility Checker anschauen. Ich habe im Foliensatz zwei Links. Den gibt es für Chrome und Firefox. Ich gehe mal drauf. Ich habe auf den Link geklickt und bin im Chrome-Webstore.

Ich bin jetzt im Chrome-Webstore. Im Prinzip ist hier die Seite, Siteimprove Accessibility Checker. Rechts ist ein Button "Hinzufügen". Ich klicke drauf, habe ein Extra-Fenster, Webextension - und schon ist es installiert. Starten oder verwenden kann man ihn oben neben der Adressleiste, wo die Webseite drinsteht. Da sind verschiedene Icons. Er müsste dort stehen. Wenn er nicht da ist, gibt es ein Puzzlestück "Extensions", da ist er aufgelistet. Ich kann ihn anpinnen, mit „pin“, der Pinnnadel. Dann steht er hier immer rechts bei der Adressleiste und kann direkt verwendet werden.

Wenn wir ihn zum ersten Mal verwenden, ich klicke einmal drauf. Da kommt erst noch das Anmelde-Fenster. Man hat ein Anmeldeformular, wo man Vorname, Nachname, Land usw. einträgt. Das kann man gerne machen, muss man aber nicht. Unter dem Formular sind zwei Buttons. Da gibt es auch "skip form", dann kann man es überspringen und ihn trotzdem verwenden.

Ich zeige Ihnen das Tool, wenn man eine Seite untersuchen will: Ich bin auf der Seite von der HdM Stuttgart. Die öffnet man. Man klickt oben auf das kleine Icon. Dann öffnet sich ein zweites kleines Fenster, das sich über die Webseite drüberlegt. Da werden Fehler oder Sachen, die gefunden wurden, aufgelistet. Dann kann man sie sich anschauen und auf die Fehler klicken. Dann bekommt man noch mehr Informationen. Es wird aufgelistet, wo die Fehler auf der Webseite auftreten. Das heißt, man kann sogar direkt zu der Stelle hinspringen, dann wird das hervorgehoben. Und es unterstützt einen dabei, solche Fehler schon einmal zu finden. Es ist auch vollautomatisch mit dem Klick.

Wie überprüfen wir das? Ich habe hier jedes Mal eine Folie. Am Beginn steht ein Satz. Hier haben wir: „Bilder haben jeweils einen Alternativtext. Dieser darf auch leer sein. Er darf nur nicht komplett fehlen.“ Dieser Satz könnte bekannt vorkommen. Das ist nämlich genau der erste Checkpunkt der Checkliste. Der Aufbau ist immer so: Wir nehmen den Checkpunkt, stellen den kurz anhand von Fehlerbeispielen vor, zeigen, wie der Fehler erkannt oder angezeigt wird vom jeweiligen Tool - hier vom Accessibility Scanner – und: Wie kann man erkennen, was hier der Fall ist?

Der erste Punkt ist, dass Bilder einen Alternativtext haben. Dieser darf auch leer sein, nur nicht komplett fehlen. Ich zeige es mal an einem Fehlerbeispiel. Ich bin auf der Seite transfermarkt.de. Da sind auch immer ganz schöne Fehler zum Angucken. Ich starte Siteimprove Accessibility Checker. Der guckt jetzt kurz drüber, lädt ein bisschen. Das Fenster hat sich schon geöffnet. Je mehr Inhalt oder Fehler vorhanden sind, desto länger braucht er zum Auflisten.

Wir haben den Fehler. Der Fehler wird in einem Extrafenster aufgelistet: „image missing a text alternative“. Das heißt, das wäre für den ersten Checkpunkt ein Ausschlusskriterium. Es gibt hier also ein Problem mit dem Alternativtext. Wir können dahin gehen. Hier unten wird ein Bild hervorgehoben. Diesem fehlt der Alternativtext. Wenn man sich doch ein bisschen mit HTML auskennt: Alternativtexte können ausgegeben werden. Die sind normalerweise ein Image-Element, da ist das Bild drinnen. Da sind auch verschiedene Sachen. Hier steht zum Beispiel „SRC“. Das ist die Quelle, wo das Bild herkommt. Irgendwo müsste in diesem Bildelement ALT, der Alternativtext, drinstehen. Da könnte stehen: „Eine Gruppe Fußballer jubelt.“ Mir fehlt gerade der Kontext. Aber im Prinzip wäre das der Alternativtext, der vom Screenreader für das Bild ausgegeben wird.

Man könnte ihn leer lassen. Dann hätte der Siteimprove Accessibility Checker keine Fehler gefunden. Das Problem ist: Es darf nicht komplett fehlen. Wenn man das ALT-Attribut komplett weglässt, kann es vorkommen, dass der Screenreader bestenfalls einfach vorliest: „Bild mit fehlendem Alternativtext“. Was ich auch schon gemerkt habe und was viel dramatischer ist, dass der Screenreader trotzdem versucht, irgendwo eine Info herzubekommen und dann zum Beispiel den Dateinamen vorliest. Dann liest er zum Beispiel hier vor: „Erster FC Saarbrücken, Pokal 33057.jpeg usw.“

Das macht die Verwendung der Webseite schwer bis unmöglich. Vor allem wegen der vielen Bilder. Wenn die alle keine Alternativtexte haben, bekommt man ständig solche kryptischen Dateinamen vorgelesen. Das macht die ganze Verwendung kaputt.

Erster Checkpunkt: Wir lassen das Tool laufen. Es kommt dieser Fehler. Das heißt, hier fehlt wirklich komplett das ALT-Attribut. Das führt zu den beschriebenen Problemen. Damit wäre der Checkpunkt jetzt nicht erfüllt.

Beim nächsten Punkt geht es um etwas ähnliches, Bildlinks und grafische Bedienelemente. Alles, was klickbar ist und gleichzeitig ein Bild. Zum Beispiel eine verlinkte Grafik, ein Bedienelement, ein kleines Icon. Hier besagt der Prüfschritt: „Diese haben jeweils entweder eine Beschriftung oder einen Alternativtext.“

Wir gehen jetzt noch mal auf die Seite. Da war nämlich auch der Fehler. Ich lasse wieder den Siteimprove Accessibility Checker laufen. Dann müsste jetzt gleich der Fehler kommen. Dann kommt der Fehler „link missing a text alternative.“ Beim Checkpunkt gefordert, dass irgendetwas, was zum Link gehört oder klickbar ist, eine Textalternative haben sollte.

Schauen wir uns das an: ein typischer Fehler. Wir haben hier oben im Header verschiedene Social Media Icons, zum Beispiel Facebook, X von Ex-Twitter, Instagram usw. Diese Elemente haben keine Textalternative. Wenn man es mit dem Screenreader verwendet, könnte er nichts ausgeben. Wenn man die Icons nicht sieht, wüsste man jetzt nicht, was man damit machen kann und würde vielleicht erfahren, dass da irgendetwas Klickbares ist. Das ist so eine typische Barriere im Sinne von diesem Checkpunkt und wird damit auch abgedeckt. Man erkennt es gut. Wenn eben dieser Fehler „link out of text alternative“ kommt, ist da ein Problem mit dem Checkpunkt.

Was noch automatisch überprüft werden kann, ist, ob alle Formular-Elemente programmatisch korrekt verknüpfte Beschriftungen haben. Hier unten am Bildschirmrand sehen wir ein Beispiel. Das ist ein kleiner Ausschnitt von einer Webseite. Wir haben stehen: „Kommentare“, darunter ein Eingabefeld. Wenn man das anschaut, würde man sehen, das steht darüber. Man würde sich denken: „In das Eingabefeld kommen Kommentare.“ Das Problem ist, wenn der Fehler „form field is not labeled“ kommt, dass keine Verknüpfung da ist zwischen dem Label und dem Eingabefeld. Das führt dazu, dass Screenreader-Nutzern nicht vorgelesen wird, wofür das Eingabefeld da ist.

Das heißt, wenn man beim Eingabefeld ist, muss man vielleicht wieder zurücknavigieren und gucken, wo das Label ist. Vielleicht steht es auch weiter weg oder es ist ein anderer Code dazwischen. Das lässt sich aber einfach vermeiden, wenn man dem Entwickler sagen würde: Das muss verknüpft sein. In der Regel müsste er wissen, wie das geht. Es ist nicht schwierig. Das gehört zum Standard. Es ist praktisch, dass es einfach direkt vom Tool erkannt wird, ohne direkt hineinschauen zu müssen.

Ein weiterer Vorteil, wenn es verknüpft ist: Man kann in der Regel auf „Kommentare“ klicken. Also auf das Label klicken, dann wird der Fokus dort hineingesetzt. Das hat eigentlich nur Vorteile.

Ein weiterer, sehr gängiger Fehler, oder hier wieder positiv formuliert: „Die Kontrastunterschiede von allen Texten im Vergleich zum Hintergrund sind mindestens 4,5:1“, bei normaler Größe. Bei großer Schrift kann es noch ein bisschen geringer sein mit 3:1. Der Helligkeitskontrast zwischen einer Schrift auf einer Webseite und dem Hintergrund, muss groß genug sein, um für jeden gut lesbar zu sein.

Ich gehe wieder auf eine Seite. Ich bin auf der Seite vom Eurospace-Center. Ich lasse das Tool laufen. Viele erkennen schon, wo Probleme sein könnten. Wenn der Fehler kommt. Hier ist es etwas unbenannt: „colour contrast does not meet minimum requirement“. Das heißt, irgendwo auf der Seite ist ein Kontrastproblem und der Checkpunkt damit nicht erfüllt. Ich kann wieder draufgehen. Das Hauptproblem: Die Webseite hat als Hauptfarben ein helles Blau und ein Weiß, da ist der Kontrast nicht hoch genug. Wenn wir darauf klicken, sehen wir direkt, dass hervorgehoben wird. Es ist einfach nur 2,31:1. Auch das Orange hier vom aktuell hervorgehobenen Punkt als Hintergrund ist eigentlich zu gering. So kann man das ganz einfach schnell mit dem Tool testen und erkennen, wo Probleme sind.

Warum ist das ein Problem? Menschen mit niedriger Sehschärfe, Farbfehlsichtigkeit, oder altersbedingtem verminderten Kontrastempfinden können den Text schlecht oder gar nicht lesen. Man merkt es selbst unten bei dem Beispiel. Wir haben weiß auf blau, das wird anstrengend. Ich gehe noch mal kurz auf die Seite. Hier haben wir die Texte blau auf weiß. Ich habe zu Hause noch einen zweiten alten Bildschirm. Da kann man das noch schlechter lesen, weil es einfach ein schlechterer Bildschirm ist. Wenn man hier oben weiß auf schwarz hat, lässt es sich besser lesen.

(Gottfried Zimmermann) Wir hatten gerade einen Crash. Entschuldigung, ich bin mir nicht sicher, wo es aufgehört hat. Ich glaube, beim Kontrast. Vielleicht können wir die Frage noch einmal aufgreifen, ob die fehlenden Alt-Tags immer vom Tool Siteimprove gefunden werden.

(Pirmin Gersbacher) Es werden viele gefunden. Aber nicht alle. Es ist nicht zu 100 % zuverlässig. Es ist heute auch nicht mehr mit drinnen bei Easy Web Check. Aber normalerweise verwenden wir zusätzlich später noch ein Bookmarklet. Das ist ein kleines Extra-Tool, wo man das Ganze noch einmal überprüfen kann. Es ist schon so, dass man nicht komplett darauf gehen kann. Aber für den Checkpunkt ist es okay.

Ich würde einmal mit meinem Notebook-Sound weitermachen und hoffe, der Ton geht einigermaßen. Ich glaube, wir kriegen das bestimmt gleich wieder hin.

(Gottfried Zimmermann) Warte mal kurz, ich bin gleich drinnen. Jetzt geht es weiter mit dem anderen Mikro.

(Pirmin Gersbacher) Ich war jetzt noch bei dem Punkt "Niedriger Kontrast bei Texten". Ich glaube, wir waren soweit fertig. Der nächste Punkt wäre noch die Hauptsprachen. „Die Sprache auf der Seite wurde korrekt per Lang-Attribut im HTML-Element festgelegt.“ Das Tool kann dies erkennen. Es spart uns einen kleinen Schritt. Ich bin wieder auf der Seite.

Gucken wir mal in das Markup hinein. Ich mache einen Rechtsklick auf der Seite unter „Suchen“. Hier rechts sieht man das HTML, wie die Seite aufgebaut ist. Grundsätzlich ist so eine Seite immer gleich aufgebaut. Sie fängt mit einer eckigen Klammer [doctype HTML] an. Dann kommt „HTML.“ Und hier drinnen muss irgendwo so ein „lang=en“ stehen zum Beispiel. Ich wechsle einmal auf die deutsche Seite. Jetzt ist wieder „HTML“ und „lang=de“. Hier wurde es richtig gelöst. Hier geht es wirklich hauptsächlich darum, wenn man so eine Seite mit dem Screenreader besucht, dann teilt sie dem Screenreader, wenn das Lang-Attribut vorhanden ist mit, welche Sprache es ist. Screenreader verwenden verschiedene Wortlisten. Bestimmte Worte gibt es nur in bestimmten Sprachen. Sie werden auch anders ausgesprochen. Deshalb sollte der Screenreader wissen, in welcher Sprache die Webseite ist und entsprechend wählen, welche Wortliste er nimmt, damit er die Wörter korrekt aussprechen kann.

Ich bin hier auf der Infoseite der pipe org extension. Und dann kommt der Fehler „siteimprove accessibility scanner, page language has not been identified.“ Das heißt, hier fehlt eben am Anfang dieses lang=en. Das müsste es in diesem Fall sein. Es würde in Deutschland dazu führen: Man redet deutsch. Man hat seinen deutschen Screenreader und geht auf diese englische Seite. Der würde die Info nicht bekommen und nimmt standardmäßig Deutsch und liest einem den ganzen Text im schlimmsten Denglisch vor. Am Anfang klingt es noch witzig. Aber eigentlich ist die Seite so nicht zu verwenden. Das Tool spart uns, wie ich es gezeigt habe: Rechtsklick, reingucken, ob es da ist. Dann bringt es den Fehler „page language has not been identified.“ Wenn der Fehler nicht oft kommt, kann man ein Häkchen dransetzen und der Checkpunkt ist erfüllt.

Ein weiterer Punkt, der von dem Tool erkannt werden kann, behandelt Fließtextlinks. Hier heißt es: „Fließtextlinks sind ausreichend vom restlichen Fließtext zu unterscheiden.“ Das heißt, sie sind entweder unterstrichen, fettgedruckt oder haben einen ausreichend hohen Kontrast zum restlichen Text von 3:1.

Gucken wir am besten wieder direkt ein Fehlerbeispiel an. Ich bin auf der Seite vom Gymnasium Neckartenzlingen. Ich lasse den Siteimprove Accessibility Checker laufen. Ich gucke, ob der Fehler auftritt. Er braucht wieder einen Moment.

Ich gucke einmal durch. Hier steht "links are not clearly identifiable" - also nicht zu identifizieren. Hier ist der Fehler.

Hier gibt es verschiedene. Erst mal vorneweg. Was ist ein Fließtext-Link? Den sehen wir hier. Wir haben einen ganz normalen Text auf der Webseite. Dort steht: „… dürfen Sie uns gerne helfen, wenn Sie mit Ihren Anregungen eine Mail an das Presse-Team des Gymnasiums senden." Und "Presseteam" ist rot und ist ein Link, wo man draufklicken kann. Dann öffnet sich direkt der Mail Client. Den Link kann man nur erkennen, weil er rot ist. Wenn man drüber geht, ändert er die Farbe. Aber im Prinzip ist er ausschließlich farblich hervorgehoben. Das ist ein Problem. Man kann es auch anschauen. Dann sind wir näher dran. Hier kann Firefox das ein bisschen simulieren. Ich habe wieder einen Rechtsklick auf „Untersuchen“ gemacht. Hier oben am Rand sind Auswahlpunkte und Pfeile. Da steht „Barrierefreiheit“. Hier kann man simulieren, wie die Seite ohne Rot wäre, bei einer Rot-Grün-Schwäche – oder auch ohne Grün.

Ich simuliere jetzt mal "kein Rot“. Es ist dann nicht mal eine Rot-Grün-Schwäche, sondern wirklich das komplette Fehlen vom Rotsehen. Auf einmal kann man den Link nicht mehr sehen. Er war ausschließlich farblich. Wenn man so etwas macht, kann es dazu führen, dass Menschen mit einer eingeschränkten Farbwahrnehmung nicht erkennen können, dass es ein Link ist. Das kann man ganz einfach lösen, zum Beispiel, in dem man es unterstreicht.

Unterstreichen ist hier auch die beste Lösung. Man kann es fett drucken. Dann ist aber nicht klar, dass es ein Link ist. Mit dem Kontrastunterschied ist es schwieriger. Der Kontrast muss groß genug sein zum anderen Text und zum Hintergrund und der normale Text auch zum Hintergrund. Das Einfachste ist wahrscheinlich zu unterstreichen oder ein Icon davorzusetzen. Darauf also achten. Das ist auch wieder der Fehler, der kommt, dass Links nicht nur mit Farbe angezeigt werden.

Was das Tool noch erkennt, ist kein so wichtiger Punkt. Es sind keine leeren Überschriften vorhanden. Wenn man es laufen lässt, kommt der Fehler "empty headings". Man hat im Markup eine Überschrift definiert, hat aber die Überschrift leer gelassen. Das kommt gerne vor, wenn man Content Management Tools wie Moodle oder WordPress verwendet. Da wurde etwas vorbereitet und vergessen, eine Überschrift hineinzupacken. Das ist geringfügig. Man wird nicht durchfallen bei einer Prüfung. Aber das Tool erkennt es und es ist wichtig. Stellen, die dann vorgelesen werden, können verwirrend sein mit dem Screenreader, wenn sie sonst gar nicht sichtbar sind.

Und das gleiche auch für Listen. Hier ist der Checkpunkt „Es sind keine leeren Listen vorhanden.“ Da hat man vielleicht etwas vergessen, einzupflegen oder es ist noch ein Überbleibsel. Das kann das Tool auch erkennen. Es bringt einen Fehler: „container element is empty.“ Da kann man den Entwickler oder Designer darauf hinweisen, dass beim Einpflegen etwas übriggeblieben ist.

Ein ganz praktischer Punkt ist noch, dass es Abschneiden und Überlagerungen erkennen kann bei Browsern. Der Checkpunkt „Texte werden bei einem Browser-Zoom auf 200 % nicht abgeschnitten und überlagern nicht.“ Das heißt, am besten auch wieder anhand einer Webseite anschauen. Diese benutzen wir ganz gerne. Das ist eine indische Online-Partner-Börse. Hier hat man verschiedene Fehler. Ich gehe hier hinein und zoome auf 200 % oder ich zoome erst mal raus. Die Standardansicht ist ziemlich klein. Wenn ich hinein zoome mit Steuerung und Rädchen oder Strg++, drücke, beginnt der Text sich zu überlappen. Das heißt, wenn irgendjemand darauf angewiesen ist, das vergrößert darzustellen, damit er die Texte besser lesen kann, kann er sie jetzt umso weniger lesen, weil hier auf einmal drei verschiedene Schriften ineinander gerutscht sind und sich überlagern. Das ist ganz cool bei dem Tool, dass es das auch erkennen kann. Man wird es trotzdem noch einmal händisch anschauen müssen. Es nimmt aber schon mal Arbeit ab. Das könnte man nämlich sonst übersehen. Nutzer mit eingeschränkter Sehfähigkeit sind oft auf die Vergrößerung angewiesen. Man muss darauf achten, dass so etwas nicht vorkommt.

Einen letzten Punkt habe ich noch: „Sichtbare und zugängliche Beschriftungen stimmen überein.“ Der Checkpunkt sagt hier: „Die sichtbare Beschriftung von links und oder Buttons und deren Alternativtexte stimmen überein.“

Wenn man Alternativtexte bereitstellt, sieht man es unten am kleinen Ausschnitt vom Markup ein Aria Label. Das wird für Screenreader dargestellt, damit es vorgelesen wird. Das beschreibt dann aber auch wieder andere Inhalte. Es kann vorkommen, wie hier, da haben wir einen kleinen Ausschnitt vom Menü. Da steht auf der Webseite „Siteimprove crawler“. Das ist ein Link, womit man das Menü öffnet zu seinem eigenen Nutzermenü.

Das war wahrscheinlich gut gemeint. Man hat „Aria Label“ draufgesetzt mit "Nutzermenü". Der Screenreader würde "Nutzermenü" vorlesen. Es steht aber „Siteimprove crawler“ drauf. Das führt zu einem Problem, wenn Menschen auf Sprachausgabe angewiesen sind, weil etwas anderes hinterlegt ist als das, was man dort sieht. Es macht die Kommunikation vielleicht schwieriger. Es sollte nichts anderes dastehen. Die Anforderung ist, dass alles, was man sieht, auch im Alternativtext vorkommen sollte. Das ist auch wieder praktisch bei dem Tool. Das wird sonst gerne übersehen. Es kann den Fehler bringen und schreibt dann „visible label and accessible name do not match.“ Dann weiß man, dass es ein Problem gibt, das von diesem Checkpunkt behoben werden muss.

Das war der erste Block zum Siteimprove Accessibility Checker, was man prüfen kann. Jetzt ist erst mal 15 Minuten Pause, glaube ich.

(Gottfried Zimmermann) Wir haben jetzt kurz vor 11:00 Uhr und machen bis 11:10 Uhr Pause, 14 Minuten. Und dann sehen wir euch wieder. Die Aufnahme werden wir auch kurz stoppen. Streckt Euch ein bisschen, macht euch frisch und dann geht es weiter mit Easy Web Check.

(Pirmin Gersbacher) Ich glaube, man kann mich hören. Perfekt! Dann ist es jetzt hoffentlich ein bisschen besser. Probieren wir es mit dem Headset. Wir waren an diesem Punkt. Wir haben angeschaut, was wir mit dem Siteimprove Checker machen können. Wir wollen noch weitere Sachen anschauen, zum Beispiel Bookmarklets.

Was sind Bookmarklets? Die werden auch Favlets genannt. Das sind Lesezeichen. Statt einer URL ist ein Javascript Code hinterlegt. Ich kann so etwas zeigen. Da ist so ein Bookmarklet. Da ist reiner Javascript Code hinterlegt. Der wird ausgeführt, wenn man draufklickt. Da sollte man auf die Quelle vertrauen, wo es herkommt. Was wir ausgewählt haben, ist offiziell. Man braucht keine Angst zu haben. Die eingesetzten Bookmarklets findet man auf der Checkliste, die ich am Anfang gezeigt habe.

Ganz unten unter der Überschrift „Werkzeugliste“ sieht man die verschiedenen Links nochmal zum Site Accessibility Checker und zu verschiedenen Browser-Erweiterungen. Ein Beispiel: „Inhalte gegliedert Bookmarklet“. Ich gehe auf den Link. Wir haben hier die Werkzeugliste auf BITVtest.de. Da kommt man irgendwann zu „Bookmarklets zur Direktinstallation“. Und da ist eben dieses „Inhalte gegliedert.“

Man hat verschiedene Möglichkeiten, man hat den Link, man sieht sehr klein am Bildschirmrand Javascript stehen. Also das, was dahinter liegt. Man kann den Link nehmen, Rechtsklick und „Lesezeichen für Link hinzufügen“. Das heißt dann wird es direkt gespeichert. Man kann auch per Drag and Drop hinzufügen. Man klickt drauf, hält die Maus fest und geht hier hoch in die Lesezeichenleiste. Dann wird es auch angelegt. Im Prinzip ist das alles, was wir machen müssen. Wir haben jetzt dieses Bookmarklet „Inhalte gegliedert“.

Wenn wir es ausführen wollen - ich mache es auf dieser Seite, klicke hier drauf, dann wird der JavaScript-Code ausgeführt. Wir sehen, da wurden jetzt noch mal zusätzlich solche Elemente hineingeschrieben. Im Prinzip ist jetzt einfach direkt auf der Webseite zu sehen, was eine Überschrift ist, welches Level die Überschrift hat. Hier zum Beispiel H2. Es geht ja von H1 bis H6. Wir können sehen, dass Texte mit Paragraf ausgezeichnet sind. Das spart uns, dass wir hier mit Rechtsklick „untersuchen“ anklicken, selbst hineinschauen und im ganzen Markup suchen, wo die Überschrift und sonst etwas ist.

Das erleichtert uns das Bookmarklet, in dem es auf die Webseite schreibt, wie die Inhalte gegliedert sind, also welche HTML-Strukturelemente bei den Überschriften und Paragrafen angezeigt werden. Hierzu haben wir wieder eine Reihe von Checkpunkten in unserer Checkliste.

Der erste Punkt ist: „Es ist nur eine H1-Überschrift vorhanden, die den Inhalt der Seite zusammenfasst.“ Das heißt, hier geht es wirklich darum, das ist relativ einfach zu prüfen. Gehen wir mal auf die Seite von der TU Dresden. Ich lasse das Bookmarklet von hier oben, klicke drauf und auf einmal sehe ich, hier ist nicht nur eine H1, wie es im Checkpunkt gefordert wird, sondern hier sind überall H1-Überschriften.

In diesem Fall wäre dieser Checkpunkt nicht erfüllt. Es ist eine Hilfe, sich auf der Seite zu orientieren. Die Lösung wäre: Man macht nur eine h1. Die ganzen H1, die wir hier gesehen haben, setzt man ein Level runter, auf H2. Dann ist es untergeordnet. Die eine H1 beschreibt den Inhalt der aktuellen Seite. Hier hat man direkt die Information.

Dann der nächste Checkpunkt: Da geht es darum, warum Überschriftenstrukturen so wichtig sind. Die Überschriftenstruktur ist sinnvoll. Wenn eine Überschrift einer anderen untergeordnet ist, sollte sie eine höhere Überschriftenebene haben. Ein Fehlerbeispiel sehen wir unten auf der Folie. Wir haben im Text verschiedene Medien mit Barrieren. Da ist das Bookmarklet H3 angezeigt. Es ist also eine H3-Überschrift.

Darunter steht ein Video ohne Untertitel. Das ist mit H1. Das macht keinen Sinn. Wenn man sich noch nicht so mit HTML auskennt, kann man es sich vorstellen wie ein Word-Dokument. Dann würde es keinen Sinn machen, dass Punkt 2.1.1 kommt und dann direkt danach kommt Punkt 2 und dann wieder etwas Untergeordnetes. Es geht wirklich darum, die Struktur der Seite aufzubauen.

Wir haben es hier noch mal. Blinde Nutzer sind auf Überschriften angewiesen, um sich die Struktur der Seite vorzustellen und auch Inhalte überspringen zu können. Genauso ist es, wenn wir eine Webseite verwenden, wie ich gerade. Ich habe gerade versucht, zum Bookmarklet zu kommen und bin drüber gescrollt. Ich habe hier Überschriften nacheinander vorgelesen. Ich würde nie hingehen und mir die komplette Seite Wort für Wort durchlesen, bis ich zu dem Punkt komme. Genauso können das auch Screenreader-Nutzer verwenden, indem sie von Überschrift zu Überschrift springen, um den Inhalt zu finden, der sie interessiert. Und dann kann man danach weiterlesen.

Daher muss die Überschriftenstruktur, also die Hierarchie, sinnvoll sein. Das ist hier gefordert. Also auf eine H1 folgt H2, dann H3 usw. Um es klarzustellen, dass der Punkt untergeordnet ist. Es ist kein neues Kapitel, „Ein Video ohne Untertitel“, sondern es gehört zum Punkt "verschiedene Medien mit Barrieren". Und darum sollte da zum Beispiel eine H4 sein.

Der nächste Checkpunkt, auch wieder mit dem Bookmarklet, ist: „Es werden nicht grundlos Überschriften übersprungen." Das ist keine so dramatische Barriere. Im Prinzip findet man sich auch trotzdem zurecht. Man kann es aber gut erkennen.

Hier sehen wir das Beispiel auf dem Screenshot. Das ist an sich eine Seite. Wir haben hier H3, eine Überschrift, Abschnitt 1. Darauf folgt eine H5-Überschrift, „Kalender“. Da ist die Frage, warum man die H4 ausgelassen hat. Dem entsprechend kann man schnell mit dem Tool prüfen und es anpassen. Es ist nichts Dramatisches, aber auf jeden Fall etwas, was wir mit aufgenommen haben, denn es lässt sich leicht verhindern.

Der nächste Punkt ist, dass Überschriften, die visuell nach Überschriften aussehen, vom Sinn her welche sind, dass man diese entsprechend mit H1 bis H6 auszeichnen sollte. Also, dass man nicht folgendes macht: wieder dieses Fehlerbeispiel, wir haben das Bookmarklet hier unten laufen lassen. Hier steht „Das Textfeld“. Und darunter steht ein Text, der beschreibt, was ein Textfeld ist. Hier wird uns mit dem Bookmarklet nichts angezeigt werden. Das heißt, man hat es mit CSS ein bisschen angepasst, darüber gestellt. Aber für Screenreader-Nutzer würde es wie normaler Text behandelt werden. Vom Prinzip kann man auch wieder erkennen, wir haben Sachen, die sind Überschriften und sehen visuell auch so aus. Mit dem Bookmarklet können wir erkennen, ob es wirklich eine ist. Wenn nicht, kann man es entsprechend anmerken, dann ist der Checkpunkt nicht erfüllt.

Genauso ist es umgekehrt. Texte, die keine Überschriften sind, sollten nicht als solche ausgezeichnet werden. Das macht die ganze Navigation mit dem Screenreader schwierig bis kaputt, wenn man von Überschrift zu Überschrift springt.

Es wird eigentlich nur verwendet wie hier unten in diesem Beispiel: Wir haben einen Text. "Viel Spaß beim Testen". Der sollte einfach nur groß sein und die Person, die es erstellt hat, dachte: „Man macht eine H1 draus, dann steht es schön groß da.“ Das macht keinen Sinn. Der Text überschreibt nichts, er ist am Ende der Seite. Man sollte nicht nur aus visuellen Gründen eine Überschrift verwenden, sondern eher den Text größer machen – mit CSS entsprechend. Überschriften soll man nicht missbräuchlich verwenden, nur um Text größer zu machen.

Eine eher geringfügige Barriere ist folgende: Es geht um Tags für fettgedruckte und kursive Texte. Der Checkpunkt: „Es werden überall die Tags strong und EM, statt dem veralteten „B“ und „I“ verwendet, um Texte fett oder kursiv zu machen“. Der Hintergrund ist folgender: Ich bin auf der Seite vom Fluchtraum Bremen und mache das Bookmarklet "Inhalte gegliedert". Hier wird von dem Bookmarklet dieses "strong" reingeschrieben. Das ist ein Positivbeispiel. Hier ist der fette Text und er wäre, hätten wir das "B" verwendet, nicht semantisch. Er wäre trotzdem fettgedruckt. Aber für den Screenreader wäre der Text eben nicht fett gedruckt.

Wir sehen es hier. Es ist schon bewusst. Man will die Info geben, es ist eine wichtige Nachricht, die an der Stelle steht. Darum hat man es auch fett gedruckt. Deswegen sollte man "strong" verwenden, weil es semantisch ist und nicht nur das "B", das "fett" darstellt. Aber wie gesagt, es ist eher geringfügig. Man kann es aber gut erkennen und deswegen haben wir es zur Überprüfung aufgenommen.

Noch eine geringfügige Barriere ist folgende. „Textabsätze werden mit P-Elementen untergebracht.“ Hier ist ein Fehlerbeispiel. Hier ist Text und hier steht einfach kein „P“ nebendran. Es ist Text, es steht ein Paragraf, wir haben es hier auf der Seite schöner gemacht. Jeder Text steht mit eigenem "P" und ist entsprechend strukturiert. Man kann es auch schnell erkennen mit dem Bookmarklet. Es sollte entsprechend hier beachtet werden.

Genauso, wenn man das Bookmarklet verwendet, sollten nicht irgendwelche leeren P-Tags dastehen. Wir haben hier dieses P, und da steht einfach nichts dazwischen und noch mal eines. Vielleicht wurde auch nur etwas vergessen, aber manchmal wird es verwendet, um zu erzwingen, dass Leerzeichen da sind, Spalten hinzufaken. Das kann man hier erkennen. Das sollte auch vermieden werden. Im schlimmsten Fall kann es sein, dass Screenreader an der Stelle "leer" vorlesen. Das macht es unnötig kompliziert und erschwert die ganze Nutzung, wenn ständig "leer" vorgelesen wird, nur weil man versucht hat, mit solchen leeren Elementen eine Struktur auf die Seite zu bringen.

Wir kommen zum letzten Punkt. Wir hatten jetzt eben den Siteimprove Accessibility Checker als Analysetool. Wir hatten ein beispielhaftes Bookmarklet. Und das dritte, was man noch verwenden kann ist auch Teil von Easy Web Check: Browser-Erweiterungen.

Im Prinzip hatten wir schon so etwas Ähnliches mit dem Siteimprove Accessibility Checker. Den haben wir im Browser installiert. Hier ist aber Software, die neue Funktionen zum Browser hinzufügt. Das sind solche speziellen Erweiterungen, um bei der Überprüfung auf Barrierefreiheit zu unterstützen. Das heißt, das prüft nicht selbst, sondern man kann sie ergänzend einsetzen oder für einen bestimmten Aspekt. Eine Liste findet man auch in der Checkliste. Unter den Prüfwerkzeugen kann man die verschiedenen Browser-Erweiterungen sehen.

Wir wollen uns taba11y anschauen. Das kann man auf Chrome nachinstallieren. Tab = Tabulator und a11y = Accessibility. Die gibt es ausschließlich für Chrome und sie ermöglicht es, sich die Tab-Reihenfolge der Webseite anzeigen zu lassen.

Vielleicht erkläre ich noch mal, warum das Ganze nötig ist. Bei vielen Einschränkungen und Behinderungen ist es nicht möglich, die Webseite mit einer Maus zu bedienen. Sie wird alternativ mit der Tastatur bedient. Ich kann es auch einmal zeigen auf der Seite der HdM. Wenn ich die Seite jetzt nicht mit der Maus verwenden könnte, kann ich alternativ die Tabulator-Taste drücken. Bei der Tastatur links ist die Steuerungstaste, die Umschalt- und darüber ist die Tabulator-Taste. Das sind zwei entgegengesetzte Pfeile. Wenn ich darauf drücke, sieht man hier, dass ein Rahmen erscheint. Der zeigt mir an, was gerade fokussiert ist. Wenn ich die Webseite verwende und ich möchte etwas über die Zulassung erfahren, dann tabbe ich hin mit der Tabulator-Taste, drücke dann Enter und jetzt bin ich auf der Seite. Ich kann hier auch wieder durchnavigieren mit der Tabulator-Taste.

Der Fokus soll nicht nur auf der Maus liegen, sondern man soll mit Tastatur die Webseite bedienen können. Würde ich so eine Webseite untersuchen, müsste ich wahrscheinlich überall durch-tabben, die Tabulatortaste drücken und schauen, ob es funktioniert. Das taba11y soll einem dabei helfen, das Ganze etwas zu verkürzen. Ich gehe noch mal auf die Seite von vorhin und lade sie erneut. Das gibt es leider nur für Chrome. Ich kann es noch mal kurz mit taba11y zeigen. Das installiert man genau gleich wie vorher den Checker. Wenn es drinnen ist, erscheint es oben rechts in der Leiste mit der URL, wo man die Adresse eingibt. Da ist so ein blauer Punkt mit einer weißen Verbindung. Da heißt es dann auch „taba11y“. Das lassen wir laufen mit „run taba11y.“

Dann wird das, was ich gerade gemacht habe, mit der Tabulator-Taste durchgehen, simuliert. Es braucht ein bisschen, je nach Größe der Seite. Das geht es jetzt Stück für Stück durch. Es sind viele Elemente. Es erspart einem das Durchtabben. Wir sehen gleich, dass jedes Element, das über die Tabulator-Taste erreichbar wäre, einen roten Punkt bekommt. Es kommt ein roter Punkt mit der Zahl der Reihenfolge, in der es angesprochen wird. Es kann uns helfen, verschiedene Barrieren zu finden.

Ganz konkret sagt der Checkpunkt: „Alle Bedienelemente sind mit der Tastatur fokussierbar.“ Das bedeutet, es gibt eine Markierung und eine Nummer in taba11y. Man kann das mitlaufen lassen. Schauen wir uns das an. Ich weiß schon, wo ein Fehler auf der Seite ist. Ich zeige es noch einmal ohne. Wenn man hier runterscrollt, hat man eine Art Menü. Man kann nach verschiedenen Profilen filtern, zum Beispiel nach Religion, Stadt, Bundesstaat, College. Es ist immer noch diese Partnerbörse. Wenn man dann irgendwo draufklickt, erscheint jedes Mal eine Liste mit Links, die man auswählen kann. Die Sachen sind mit der Maus bedienbar. Da kann man entsprechend herausfiltern.

Wenn ich taba11y laufen lasse und zur Stelle hinscrolle, sehe ich die Punkte, die klickbar waren, haben alle keine Nummer. Wenn ich die Webseite nur mit Tastatur verwenden kann, habe ich keine Möglichkeit, den Filter mit Religion oder Stadt oder Bundesstaat oder sonst etwas zu wählen. Ich kann nicht zu den Links gelangen. Vielleicht ginge es über Umwege oder ich könnte es irgendwo suchen. Aber das ist nicht Sinn der Sache. Die Tastaturbenutzer wurden hier vergessen. Das kann man erkennen, indem man sieht, dass keine Zahl draufsteht.

Der nächste Punkt: Es sollte nicht nur eine Nummer haben, sondern die Reihenfolge sollte sinnvoll sein. Das kann man gut mit dem taba11y erkennen, weil es durchnummeriert ist. Man macht am besten einen Screenshot von diesem Beispiel. Hier ist zum Beispiel ein Fehler. Wir haben eine Webseite, wir haben taba11y durchlaufen lassen. Da steht links oben „9“, rechts davon „10“, das ist noch in Ordnung. Aus dem Bild raus: „11“, „12“, „13“. Dann geht es nach unten. Nach der 13 steht wieder die 25. Wenn man mit der Tastatur durchgeht, geht es von links nach rechts zunächst, Dann springt es weiter nach unten, dann nach oben, dann wieder nach oben. Das ist nicht Sinn der Sache. Das lässt sich auch leicht überprüfen. Man merkt selbst, ob das sinnvoll ist. Es kann auch sein, dass man hochspringt und schauen muss, wo gerade der Fokus ist. Vom Prinzip kann man danach gehen: von oben nach unten und wir lesen von links nach rechts. So sieht man normalerweise eine Seite. Die Hauptanforderung ist: Es muss sinnvoll sein. Nicht wie hier 25 neben 13.

Ein weiteres Tool, das ich zeige, ist der WCAG Contrast Checker. Das ist auch wieder eine Browser Extension. Es ist auf den Folien verlinkt, aber auch auf der Checkliste. Es ist noch eine zusätzliche Absicherung. Wir hatten vorhin schon den Siteimprove Accessibility Checker. Der schaut schon mal nach Kontrasten. Das hier ist noch ein zusätzliches Tool, das man laufen lassen kann. Dann werden verschiedene Kontrastfehler komplett automatisch erkannt.

Ich lasse es beispielhaft laufen auf der Seite von der HdM Stuttgart. Wenn man es installiert hat, ist oben eine kleine Farbpalette angezeigt. Die heißt eben auch wieder „WCAG Color-contrast-Checker“. Es erscheint links am Bildschirmrand ein Fenster. Dort sieht man mögliche Probleme. Man muss beachten: Es werden auch Sachen angezeigt, die absichtlich versteckt sind.

Wir haben hier zum Beispiel weiß auf weiß, ein Kontrast von 1. Das ist kein Fehler, sondern man wollte, dass man das nicht sieht. Ein Vorteil von dem Tool ist, es kann Siteimprove nicht so gut, Es werden auch Zustände beachtet. Hier ist das Rot auf dem Dunkelgrau. Das hat einen zu niedrigen Kontrast. Wir haben das Kreuz, 1.98. Das Verhältnis ist zu gering. Dann sehen wir, dass die Lupe hervorgehoben wird. Aber die ist eigentlich weiß.

Wenn man darüber hovert, wird sie rot. Es ist wichtig, die verschiedenen Zustände zu beachten. Mit dem Tool kann man helfen, das zu überprüfen. Das Rot auf dem Grau-Kontrast ist zu niedrig. Das muss man dann entsprechend bei diesem Checkpunkt wieder anmerken.

Ich glaube, wir haben noch 20 Minuten für Fragen. Damit wäre ich auch am Ende mit diesem Teil von Easy Web Check-Checkliste. Sie findet man auch online. Da sind auch die ganzen restlichen Punkte drauf. Danke für die Aufmerksamkeit.

(Gottfried Zimmermann) Vielen Dank, Pirmin. Das war ein Teil von Easy Web Check. Du hast die Details sehr schön gezeigt. Ich hoffe, dass ihr, die jetzt zugehört habt, ob ihr nun Web-Experten seid oder Laien, etwas mitgenommen habt und es selbst in der Zukunft, zumindest teilweise, anwenden könnt.

Es gibt eine komplette Checkliste. Ich habe den Link in den Chat gestellt. Wir bieten auch Schulungen an vom Kompetenzzentrum für Digitale Barrierefreiheit an der Hochschule der Medien.

Es gab eine Frage im Chat. Ihr könnt Fragen hineinstellen. Bei umfangreichen Fragen können wir euch auch per Audio freischalten.

Die erste Frage im Chat: Gibt es für den Easy Web Check eine eigene Webseite, wo man die aktuellen Dinge herunterladen kann?

Die haben wir im Moment noch nicht. Es gibt ein Webinar darüber, wo wir das, wie heute auch, auf einer Webseite festhalten, was die aktuelle Version ist. Aber wir sind dran. Wir werden auf jeden Fall eine separate Webseite für Easy Web Check erstellen.

Dann dürft ihr gerne eure Fragen stellen. Ich fange an, Pirmin: Wenn man sich eine einzelne Webseite anschaut und alle Tests durchmacht mit Easy Web Check, mit wie viel Zeit muss man rechnen?

(Pirmin Gersbacher) Schwierig. (lacht) Ganz konkret haben wir das noch nicht angewendet. Wir haben es nicht offiziell angeboten, sondern eher durchgesprochen. Der Workshop selbst geht vier Stunden. Die braucht man auch wenn man die verschiedenen Beispiele zeigt. Ich will nichts Falsches sagen. Aber je nach Komplexität kann es mal zwei Stunden gehen, um schnell drüberzugucken, oder eben länger.

(Gottfried Zimmermann) Das heißt, wenn man es mit dem BITV-Test vergleicht. Der ist aufwendiger. Wie lange würde man da ungefähr im Vergleich brauchen?

(Pirmin Gersbacher) Das kann Tage gehen.

(Gottfried Zimmermann) Man hat tatsächlich eine schnellere Methode, allerdings natürlich unvollständig. Das ist schon klar. Darum empfehlen wir es, das zu machen, weil es nicht nur einfacher ist, sondern schneller, um einen ersten Einblick zu bekommen. Wir haben Kunden, die kommen und sagen: „Wir hätten gerne ein Prüfsiegel für unsere Webseite.“ Dann machen wir einen ersten Check oder schauen einfach mal drüber. Wir sehen, da ist noch so viel im Argen. Dann sagen wir: „Bevor wir mit einem BITV-Test anfangen, machen wir einen einfachen Test und schulen die Entwickler\*innen und Designer\*innen und zeigen, wie man es richtig macht.“ Und wenn man meint, man hat alles richtig gemacht, können wir daran denken, einen umfangreichen BITV-Test zu machen.

Gerne eure Fragen im Chat.

Welche Produkte genau sind von der Anpassung der EAA-Norm betroffen? Webauftritte, Desktop und Mobil oder auch alle Apps, E-Books, PDF-Dateien. Wenn auch andere Produkte betroffen sind, habt ihr Tool-Empfehlungen oder Info-Plattformen in die Richtung Apps, E-Books und PDFs?

Die EAA hat noch keine Norm, mit der er verbunden ist. Aber es wird eine Zukunftsvision der EN 301549 sein. Wir haben eine Form davon schon vorliegen. Das wird aber nicht die sein, die letztendlich harmonisiert wird. Wir wissen, es betrifft nicht nur Webauftritte, sondern alle Software, Apps, Dokumente, die verbunden sind, zum Beispiel mit E-Commerce, also mit Onlineshops. Also mit allem, was verbunden ist, wenn ein Kaufvertrag zustande kommt zwischen einem Unternehmen, einer Organisation und einem Endkonsumenten. Also nur B2C.

Alle Formen von mobilen Medien sind betroffen, die beteiligt sind. Zum Beispiel auch, um einen Termin online mit dem Friseur auszumachen, wenn der Friseur entsprechend groß ist. Es gibt eine Ausnahme für Kleinstunternehmen.

Das heißt die Frage ist durchaus berechtigt, wenn Produkte betroffen sind. Das sind sie. Nächste Frage: Habt ihr Tool-Empfehlungen in Richtung Apps, eBooks, PDFs? Zu den Apps kann Pirmin etwas sagen.

(Pirmin Gersbacher) Ich habe geschaut, was es an kostenlosen Tools gibt. Wirklich kostenlos gibt es nur für Android etwas. Das ist der Accessibility Scanner. Er ist von Google. Und so kann man da auch ganz gut darauf vertrauen. Er macht ein paar Prüfungen. Das geht ein bisschen in die Richtung wie das Site-improve-Tool, das wir gesehen haben. Man öffnet die App, lässt es laufen. Es guckt nach Textalternativen, Kontrastproblemen, Größe von Buttons. Aber es ist weniger im Umfang als die Web Tools, weil es ein Blackbox-Testing ist. Man kann nicht das HTML einer Webseite untersuchen und hat keinen Zugriff auf den Code.

(Gottfried Zimmermann) Bei eBooks kommt es darauf an, ob sie dem Standard EPUB3 entsprechen, oder als PDF gemacht wurden. Für EPUB3 gibt es auch ein Tool, um das zu prüfen. Das müsste ich nachher recherchieren und stelle es später in den Chat. Für PDF gibt es PAC. Dazu hören wir heute Nachmittag mehr. Da verweise ich auf später, bleibt einfach dran.

Alle diese automatischen Tools können nicht alle Anforderungen überprüfen, sondern nur, was automatisch entdeckbar ist. Das sind in der Regel, so ganz grob, ein Drittel der Fehler. Man sollte also immer von Hand nachprüfen. Man kann einen Eindruck bekommen, zum Beispiel beim Einkauf. Ob das, was man geliefert bekommt, einigermaßen barrierefrei sein kann.

Die nächste Frage: Habt ihr Empfehlungen zu guten, barrierefreien Webseiten, Empfehlungen für Designer? Da gibt es zwei Likes dazu.

Das ist jetzt interessant. Es klingt ein bisschen wie, was ich von Kunden oft höre: "Könnt ihr mir eine Schulung für digitale Barrierefreiheit in 60 Minuten geben?"

Natürlich gibt es diese Empfehlungen. Aber was man dann anwenden muss, um Webseiten barrierefrei zu gestalten, ist immer individuell. Man muss eigentlich die ganze Bandbreite kennen. Die Frage ist schon spezifischer. Als Beispiele für gute, barrierefreie Webseiten: Da haben wir keine Liste vorliegen. Aber man kann zu den Organisationen gehen, die sich dafür einsetzen, dass es mehr Barrierefreiheit gibt. Den BITV Bund zum Beispiel, können wir nennen. Oder auch die Fachstelle für barrierefreies Design des Bundes. Oder die Pfennigparade hier. Oder auch bei uns das Kompetenzzentrum für digitale Barrierefreiheit der Hochschule der Medien oder IAAP D-A-CH. Also schaut da einfach nach und nehmt die Organisationen, die sich dafür einsetzen.

Empfehlungen für Designer: Es gibt Schulungen in der Richtung, aber das kann man nicht einfach nur in einer Stunde kurz machen.

(Pirmin Gersbacher) Ich habe noch kurz etwas gepostet. Es sind jetzt nicht so Empfehlungen für Designer. Wenn man auf den Link klickt, sind es verschiedene Accordion-Elemente, Karussell, sonst irgendwas. Und das ist von W3C. Da hat man zumindest eine Inspiration, wie man so etwas barrierefrei machen kann. Sie haben darauf geachtet.

(Gottfried Zimmermann) Super, dass jetzt Fragen kommen. Es sind alles Gute Fragen. Gibt es eine Liste für Websites, die vom Barrierefreiheitsstärkungsgesetz betroffen sind? – Nein, denn das muss man immer selbst herausfinden, ob man betroffen ist. Es sind auch nicht nur Websites betroffen. Es gibt verschiedene Sektoren im Wirtschaftsbereich, die betroffen sind. E-Commerce ist ganz wichtig. Aber auch der Fahrkartenerwerb oder die Automaten selbst, die dann oftmals eine Software draufhaben. EBooks, da muss man eben reinschauen, gegebenenfalls Juristen fragen. Immer dann, wenn eine Kundenbeziehung B2C da ist. Wo etwas gekauft wird, also erworben wird, ist das Vorschrift.

Die nächste Frage: Wie viele Fehler sind insgesamt noch zulässig? (Lacht) Das ist eine tolle Frage. Wir kommen darauf, was ist eigentlich Barrierefreiheit? Wir haben öfter gehört, dass man von „barrierearm“, nicht „barrierefrei“ reden sollte, weil so ganz perfekt kann man nicht sein. Wenn wir von Barrierefreiheit reden sollten, dann im gesetzlichen Kontext oder in dem Kontext, ob man eine Norm erfüllt hat oder nicht. Dann muss man sagen, man fällt durch. Wenn nur ein einzelner Fehler auftritt, ist es nicht mehr barrierefrei, also ein Fehler im Sinne von: Man hat gegen eine Norm verstoßen oder eine Anforderung nicht eingehalten.

Der Alexander kann vielleicht noch mehr dazu sagen. Was ich dazu sagen darf, es sind alles Menschen, die einen bewerten. Also bei den Bundesüberwachungsstellen oder bei den Marktüberwachungen für kommerzielle Anbieter. Die sehen natürlich schon, wenn man sich bemüht. Wenn man einen Fehler macht, ist es wesentlich besser, als viele Fehler zu machen. Vor dem Gesetz ist man aber bei jedem Fehler gleichermaßen durchgefallen. Alexander, möchtest du etwas dazu sagen? Dann gebe ich dir das Mikro.

(Alexander Pfingstl) Du hast völlig recht. Es ist immer wichtig, darauf zu achten, keine Blockaden auf den Seiten zu haben. Das sagen wir immer, wenn man im Bereich E-Commerce anfängt. Man möchte, dass die Leute die Seiten nutzen und dort einkaufen können. Das erste Ziel sollte es also sein, dass jede Nutzergruppe irgendwie durch diese Seite durchkommt. Und dann ist es so, wie du es gesagt hast: Es wird sicherlich darum gehen – man kann sich als Nutzer irgendwann ja auch beschweren, bei der Schlichtungsstelle des Bundes oder bei den Betreibern der Seite selbst; also dann, wenn man meint, dass es Probleme auf der Seite gibt. Dann wird es sicherlich bei der Beurteilung des Gesamtfalls eine Rolle spielen, zumindest nach meinem Verständnis. Wie bemüht sind die einzelnen Marktteilnehmer, die Seite zu verbessern? Gibt es Fortschritte und ein Konzept dafür? Wenn das da ist, gehe ich davon aus, dass es seitens der ganzen Parteien anerkannt und entsprechend behandelt wird von den ganzen Regelungen und auch in Bezug auf die möglichen drohenden Strafen, die im Gesetz stehen.

(Gottfried Zimmermann) Danke. Interessant in dem Zusammenhang ist auch die Weiterschreibung der WCAG, der Web Content Accessibility Guidelines. Momentan haben wir die Version 2.2. Da gibt es noch das sogenannte Konformitätsprinzip - alles oder nichts. Man muss alles erfüllt haben. Bei WCAG 3.0, das ist schon im Entwurf, wird es so sein, dass man von 0-100 Punkte bekommt und dann eben auch mit wenigen Fehlern trotzdem noch konform ist. Das entspricht eigentlich mehr der Realität. Aber bis die Gesetzgebung dahin kommt, dass WCAG 3.0 gesetzlich forciert wird, da gehen wohl noch 5-10 Jahre ins Land. Das ist meine Einschätzung.

Wie groß sind Kleinstunternehmen, die ausgenommen sind? – Das ist relativ einfach zu beantworten. Man muss beides erfüllt haben. Man muss nämlich weniger als zehn Mitarbeitende haben und unter 2 Millionen € Umsatz im Jahr. Also wenn beides erfüllt ist, weniger als zehn Angestellte und unter 2 Millionen Umsatz, ist man ausgenommen.

Was haben wir noch an Fragen? Das ist speziell für B2B-Webseiten. Diese sind tatsächlich nicht betroffen vom Gesetz. Aber trotzdem ist es schön, wenn man sie barrierefrei macht. Die Kunden sind Unternehmen. Die haben Mitarbeitende, die darauf angewiesen sind, dass es barrierefrei ist. Die sind gesetzlich verpflichtet, für ihre Mitarbeiter\*innen mit Einschränkungen entsprechende Arbeitsbedingungen zu gewährleisten, gemäß SGB IX zum Beispiel. Deswegen ist es immer gut, sich daran zu halten, barrierefreie Software zu erstellen.

(Alexander Pfingstl) Öffentliche Stellen.

(Gottfried Zimmermann) Öffentliche Stellen sind durch BITV und das Behindertengleichstellungsgesetz natürlich dazu verpflichtet, dass alles barrierefrei ist. Auch die interne Software, die verwendet wird.

Ein Kommentar: Ich finde es Wahnsinn, dass man so spät in Deutschland diese Regel, das Gesetz walten lässt. Bis Juni 2025 ist es ja nicht mehr lange. Können Anbieter, die den Fixtermin nicht einhalten, mit Strafen rechnen? Oder wie kann man sich das vorstellen? Es ist keine kleine Aufgabe.

Ich gebe dir recht. Da müssen wir die Politik fragen, was ich gesehen habe, ist, dass man sehr viel Zeit verplempert hat – damit, dass man sich über die Zuständigkeit gestritten hat: Bund, Länder, Wirtschaft oder Sozialministerium. Eigentlich ist es seit 2019 bekannt, dass da etwas kommen wird, weil der European Accessibility Act schon im Jahr 2019 herauskam. Es ist also nicht überraschend. Aber man hat tatsächlich schon viel Zeit verbraten. Das ist schade. Ich bin aber trotzdem zuversichtlich, dass die Privatwirtschaft das insgesamt besser hinkriegen wird als die öffentlichen Stellen, die seit 2019 in Teilen, seit 2020 komplett barrierefrei sein müssen. Und wir sehen immer noch große Defizite.

Kommentar: Wenn man mit Betroffenen testen möchte, ist es natürlich immer zu empfehlen, mit Betroffenen selbst zu testen. Aber die Verfügbarkeit dieser Personen ist auch schon begrenzt. Weil eben viele testen lassen wollen. Es ist aber eine gute Idee. Da kann man sich an Organisationen wenden, die mit Menschen mit Einschränkungen arbeiten, die sie angestellt haben. Zum Beispiel hier die Pfennigparade. Diese Personen werden natürlich dafür bezahlt.

Das Barrierefreiheitsgesetz greift ja ab Juni 2025. Gestern wurde ja gesagt, dass die Aktualisierung der EN 301549 erst zu Anfang 2026 kommen wird. Wie wird mit dem zeitlichen Gap umgegangen? Referenziert man in der Zwischenzeit auf die aktuelle EN 301549? – Eigentlich zwei Antworten. 1. Ja, wir nehmen einfach im Moment das, was es schon gibt. Es ist nicht nur die EN 301549, sondern ein anderer Standard, den es in der Vorversion schon gibt. Der ist EN 17161. Da geht es um Design für alle. Das ist auch wichtig für das Barrierefreiheitsstärkungsgesetz. Und dann gibt es noch mal die EN 17210. Da geht es um Barrierefreiheit für Gebäude und bauliche Dinge. Aber auch so etwas wie Service Desks, also vor Ort, baulich.

Und dann gibt es noch mal 3 weitere Standards, die aber noch nicht entwickelt sind. Da gibt es noch nicht einmal eine Vorversion. Da sind zum Beispiel auch Helpdesk oder Notrufnummern dabei.

Erste Antwort: Man kann die Version nehmen, die es gibt, wenn es sie schon gibt.

Zweite Antwort: Von den Standards gibt es immer schon Vorversionen, die in den Kreisen der Standardisierung zirkulieren. Die kann man anschauen. Zum Beispiel soll es, zumindest laut Zeitplan, ab Herbst 2024, eine Vorversion für EN 301549 geben, nämlich die Version 411.

Es ist immer gut, sich über barrierefreies Design zu informieren, die Mitarbeiter und Mitarbeiterinnen schulen zu lassen. Wenn die Standards da sind, kann man diese entsprechend lesen, verstehen und umsetzen. Es wird nichts komplett Neues erfunden.

Kommentar: Daran arbeiten wir als Agentur für Barrierefreiheit. Wir konzentrieren uns auf das Prüfen aus Nutzerperspektive. Das ist sicher gut und wichtig.

Eine Empfehlung von meiner Seite: Bevor man die Betroffenen als Tester mit reinnimmt, sollte man versuchen, die technischen Anforderungen zu erfüllen. Denn es gibt einfach eine ganze Menge zu betrachten. Und wenn man das praktisch mit „trial and error“ mit Betroffenen macht, dann ist es sehr mühsam und vielleicht auch ein bisschen Zeitverschwendung. Also zuerst mal wirklich versuchen, das anzuwenden, was es als technische Standards gibt und dann natürlich mit Betroffenen testen. Das ist immer gut und wichtig.

Die EN-Normen, ja. Ich schreibe das mal noch hinein. DIN 17161 ist die deutsche Übersetzung der EN 17161, da geht es um Design für alle. Die DIN 17210. Da geht es um Barrierefreiheit für Gebäude, zum Beispiel auch Servicedesks oder Haltestellen. Ich kann auch die Links noch reinstellen. Diese Standards muss man käuflich erwerben. Ich gehe aber davon aus, dass man sie auch kostenlos herunterladen kann, wenn man sich registriert, weil sie für das Gesetz verpflichtend sind. Für die EN 301549 gibt es diese Möglichkeit schon. Man kann sich bei der Bundesüberwachungsstelle registrieren und klarmachen, warum man den Standard anwenden muss. Und dann kann man ihn kostenlos herunterladen.

Ich glaube, wir haben eine Punktlandung geschafft. 12:00 Uhr, jetzt machen wir eine Mittagspause. Nutzt diese, um diese Meinungsumfrage mitzumachen, das geht bei AccessibleEU ein. Das hilft bei der Finanzierung der Schriftdolmetschung und der Gebärdensprachdolmetschung. Es ist gut, wenn ihr positive Meinungen abgebt. Aber gerne auch Kritik, damit können wir uns verbessern. Wir sehen uns wieder um 13:00 Uhr. Bis dahin wünsche ich euch eine schöne und erholsame Mittagspause.