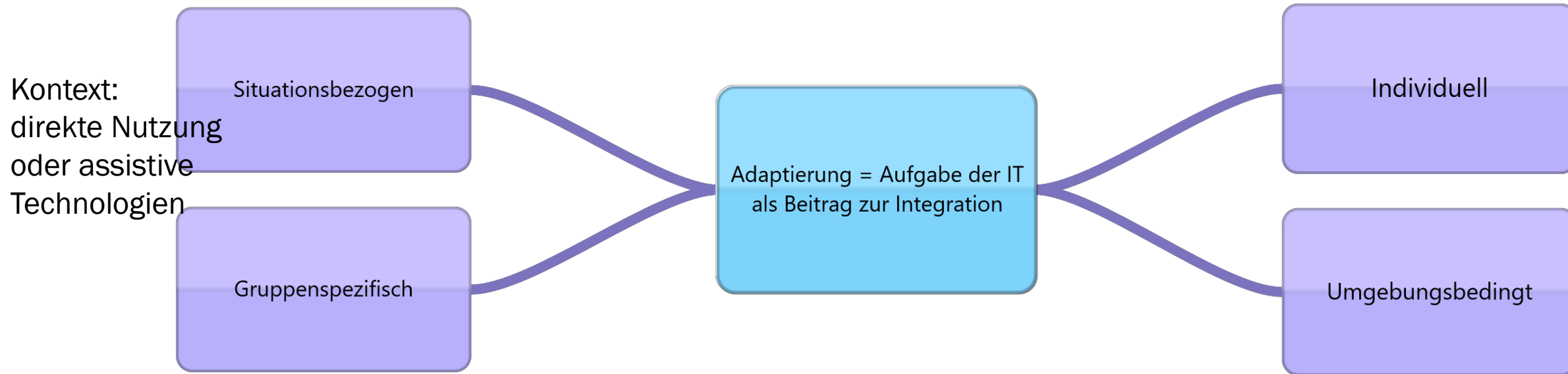


# BARRIEREFREIE DOKUMENTE

# 1. USER

## Was ist eine Behinderung? Arten

Körperliche od. Geistige Beeinträchtigung, die eine oder mehrere wichtige Lebensaktivitäten erheblich einschränkt. ZB.: mangelnde Koordination, Sprachbehinderung, Dyslexie, Sehbehinderung



## Was ist Barrierefreiheit bzw. Accessibility

Umfang, in dem Produkte/Systeme/Dienstleistungen/Umgebungen/Einrichtungen durch Menschen aus einer Bevölkerungsgruppe mit den weitesten Benutzeranforderungen, Merkmalen und Fähigkeiten genutzt werden können, um identifizierte Ziele in identifizierten Nutzungskontexten zu erreichen.

# Usability als Teil der Accessibility

## USABILITY (Gebrauchstauglichkeit)

- Effektivität
- Effizienz
- Zufriedenstellung

## Accessibility (Barrierefreiheit)

- Wahrnehmbarkeit
- Bedienbarkeit
- Verständlichkeit
- Robustheit
- Zugang zu Infos
- Kommunikation ermöglichen
- Mobilität verbessern

## Universelles Design, Design 4all

- Breite Nutzbarkeit: viele Anforderungen
- Flexibilität: vielfältige Ein-und Ausgabeformen
- Intuitivität: ohne Hilfe nutzbar
- Sensorisch wahrnehmb. Infos: Ausgabekanäle
- Fehlertoleranz: auch unexakte Eingaben ok
- Niedriger körp. Aufwand: keine kompl. Bewegungsabläufe
- Größe u. Platz für Zugang: keine räuml. Einschränkung

# Screen Reader

Konkretes Beispiel: → NVDA

Funktionsweise? (auch dann Bezug zu JavaScript)

Screenreader (Aufgaben, woher Infos für zB Linien in EXCEL)

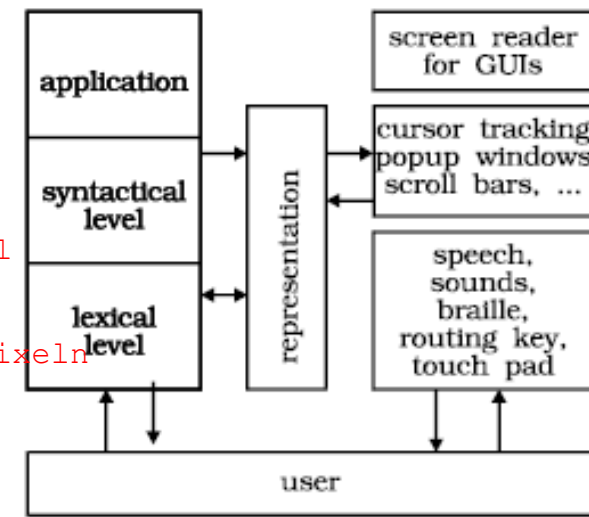
Beispiel:

- Abby Filereader = Texterkennungsoftware,
- Hörbuch lesen durch Sprachsynthese
- Browser: DOM = document-object-model (ARIA Technik)
- Nicht sinnvoll wäre: vorlesen von Pixeln als Text

## Lesen mit & ohne Braille

## Lesen mit Sprachausgabe

## Merkmale



## Konzepte

- VERFOLGEN: Text/Maus Cursor (Änderungen auf Bildschirm ausmachen)
- ERKUNDEN: Zeilenweise; unabh. von Anwendung (Offscreen Model, Pixel abgebildet)
- ROUTING (durch Taster als Mausersatz): Sensoren bestätigen; Synch. Zw. Anwendungsfokus & Erkundung mgl.
- ADAPTIERUNG: Funktionstasten; Konfigurationsdateien; Skriptsprache (nur 2 mgl. Zur Zeit)
- PROBLEM: Nicht-verfolgen!

- Verfolgen Cursors, Menüoptionen etc durch Sprachsynthese (für Überschriften, Absätze, Sätze, Wörter, Buchstaben)
- Geschwindigkeit bis zu 500 WpM
- Erkunden mit weiterer Tastaturebene (Jaws)
- Mausersatz: Kontextmenü per Tastenbefehl; Routing per Pfeiltasten und Aktivierung (4 Ebenen, 1 zusätzl. Zu shift, Umschalt, Strg)
- Adaptierung durch GUI: Einstellungen in Android; scripting per GUI in Windows (Jaws = job accesss with speech)

- Open Source/ frei
- Skriptfähig
- Ohne Installation
- Java Script
- Mobile Geräte: Mobile Speak (Android, Symbian); Talks (Symbian); VoiceOver (iPhone)

## Interaktion mit Routingtasten

- ROUTING = MAUSERSATZ
- Maus bewegen → klicke Braillesensor
- Mausschalter bestätigen → Klicke Braillesensor
- Modus notwendig! Quelle selektieren; Erkunden Ziels; Ziel wird erzeugt
- Halte Maustaste: 1. Doppelklick  
Ziehen & loslassen: 2. Doppelklick

- Screenreader = multimodales System: dh visuelle Darstellung + Braille + Sprachsynthese
- BOTTOM UP: OCR, Bilderkennung (Quelle der Info ist Zeichenerkennung, Iconerkennung → geringe Performanz)
- TOP DOWN: Zwischensprache u. mehrfache unterschiedliche Aufbereitung (Anwendung erzeugt Braille); Webtechniken zur Profilierung (self voicing od. Selfbrailing → sollte Braille und Sprache gleich gut adressieren)
- MIDDLE OUT: filter in der graf. Benutzungsoberfläche (GUI)

Usergruppe	Barrieren	Lösungen
Gehörlos	Hört ggü. Nicht Beim Telefonieren	<ul style="list-style-type: none"> <li>- Videotelefonie per Gebärdensprache</li> <li>- Redundanz: Gebärde &amp; Sprachtranskription (Untertitel, caption vs subtitle)</li> <li>- Avatare visher abgelehnt</li> <li>- Bluetooth für Bridges zw Handy &amp; SmartTV</li> <li>- Isolierung Sprechers für Hörgeräte</li> <li>- Tooltips: digitales Lippenlesen</li> </ul>
Blind oder Sehbehindert	Kann nichts oder eingeschränkt sehen, zB Farbschwäche, Größe, Verzerrung	<ul style="list-style-type: none"> <li>- Ausgabe: Braille, Sprachsynthese</li> <li>- Eingabe: Tasten, Sprache, Gesten</li> <li>- Magic Zooming</li> <li>- Taktile Grafiken</li> <li>- Klausur: Bereitstellung v. Tablet od. Tastatur &amp; Maus &amp; Lupenfkt. Od. Kamera + Extrazeit</li> </ul>
Taubblind (bsp Helen Keller)	Geringe Sprachkompetenz, Braille ungeeignet	<ul style="list-style-type: none"> <li>- Gebärdensprache → Handschreiben</li> <li>- Alltagsgeräte mit Vibration</li> <li>- Gestenerkennung mit Zoneneinteilung</li> </ul>
Körperlich eingeschränkt, Bsp. Armlos?	Handsteuerung wegen fehlenden Händen, oder zittern, Rollstuhlfahrer erreicht Höhe nicht	<ul style="list-style-type: none"> <li>- Sprachsteuerung</li> <li>- Eyescrapping</li> <li>- Bluetoothcontroller</li> <li>- Tablet für Smarthomesteuerung –Gesten statt Maus greifen</li> <li>- Tastaturgröße anpassen, Kopfmaus</li> <li>- Mehrfachanschläge</li> <li>- Scanning</li> </ul>
Kognitiv eingeschränkt	Lernbehinderung (zB Dyslexie) bis zu schwerer geistigen Einschränkung (BSP Autismus, Schlaganfall, Parkinson, chirurgischer Eingriff, Epilepsie)	<ul style="list-style-type: none"> <li>- Strategien: Sprachsynthese, Wörterbücher</li> <li>- Videos: Bilder und Texte zur Erklärung, farbliche Unterschiede für Wortarten</li> </ul>
Ältere Personen	Verlust von Fähigkeiten (Sehen, Hören, Motorik) bis zu Demenz	<ul style="list-style-type: none"> <li>- Seniorennfreundliche Websites (Größe, Kontraste, Layout. Navigationsfreundlichkeit erhöhen, Medieneinbindung)</li> </ul>

Neue Formen der MCI	Barrieren bei unbekannter Adaptierung
Grafische Benutzungsoberflächen	<ul style="list-style-type: none"> <li>- Pixelbarriere</li> <li>- Maus und andere Zeigeeinstrumente</li> </ul>
Hypertext/ Internet	<ul style="list-style-type: none"> <li>- Mangel an Überblick</li> </ul>
Multimedia/ interactive Medien	<ul style="list-style-type: none"> <li>- Multimedia Barriere</li> <li>- Mangel an temporaler Steuerbarkeit</li> </ul>
Virtual Reality	<ul style="list-style-type: none"> <li>- Nicht visuelle Immersion</li> </ul>
Sprachassistenten	<ul style="list-style-type: none"> <li>- Sprachvermögen als Barriere</li> </ul>

Handy	Display, Ausgabe, Steuerbarkeit
Telefon	Behinderungsarten s. Übung zu VOIP-Oberfläche
Smart TV	
Smart Watch	Hätte Oscar Nobel ein Problem gehabt diese zu bedienen? (er hatte Epilepsie)
Taschenrechner	

# 2. WAI CONTENT

WCAG

Barrierearten

ARIA



Was sind Barrieren?

Reglungen Barrierefreiheit im Internet (WCAG, BITV) und welche Kriterien (Benutzbarkeit, Wahrnehmbarkeit,...);

ARIA & WCAG

dynamische Inhalte (wie kenntlich machen: Stichwort ARIA. Beim Chat: wie auf Braillezeile immer das aktuelle?)

## WCAG 2.0 (USA)

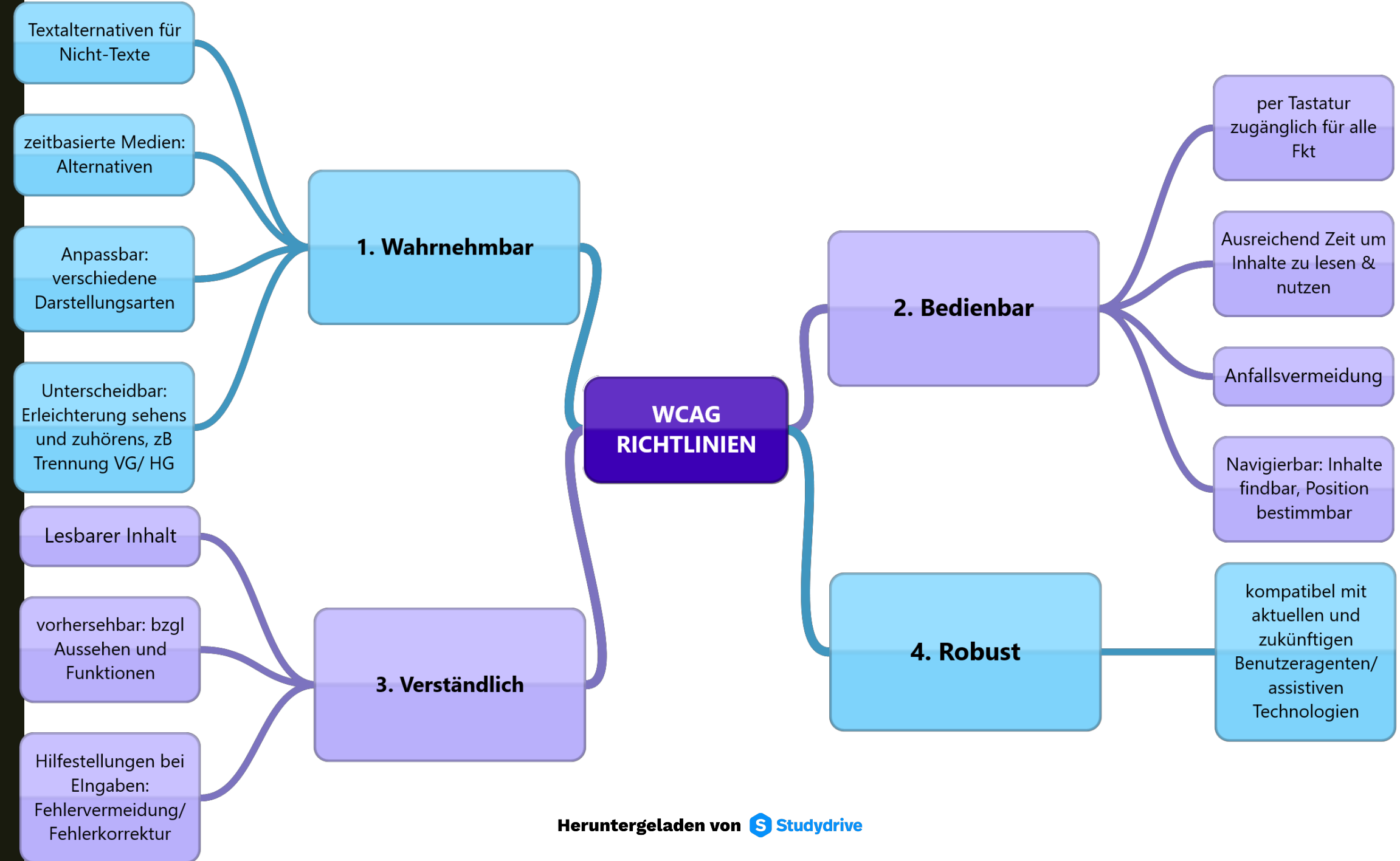
4 Prinzipien der web content accessibility guidelines

- Wahrnehmbarkeit (perceivable)
- Bedienbarkeit (operable)
- Verstehbarkeit (understandable)
- Robustheit (robust)

- Konformitätsstufen: A grobe Fehler, AA Forderungen, AAA ???

## BITV

- Behindertengleichstellungsgesetz: Objekt Apps, Formulare, Terminplanung, PDF bei öffentl. Stellen
- Enthält 4 Prinzipien nach WCAG
- Gebärdensprache auf Startseite
- Überwachungsstelle veröffentlicht Standards



# Einteilung nach Barrieren

1	Farbe	Farbsehschwäche → Styleswitcher, Kontraste → Unterschiede von 1:3 in Farbhelligkeit in RGB Raum; Farbwahl in Triaden
2	Textlesbarkeit	Erkennbarkeit durch Größe, kein Autorefresh!, semantische Markups für Screenreader (CSS: span & div für Abtrennung Textbereiche); keine Asciigrafiken
3	Seitenlayout	Nähe & Gruppenbildung, Anordnung in Blöcken, aber KEINE Tabellen (sondern CSS) → nutze Templates; Responsive Design: Flexibilität bei verschiedenen Displaygrößen; HTML 5: Struktur im Body aufbauen:
4	Überschriften	Html von Screenreadern gesondert behandelt, leicht per Tastatur zu besuchen, mittels CSS gestaltbar: <h1> einmal benutzen, danach zwingend <h2> (nicht zu h3 überspringen), in Formularen <legend> statt <h*>
5	Listen	MarkUp wichtig! Usability Problem; besser: Tags (zB ul, ol, dl) Problem: tief verschachtelte Listen mit >5 Ebenen, schwer zu verbalisieren
6	Sprachinhalt	Sprachwechsel : <span lang “eng” xml: lang=“en”> Browser </span> Zitate: <blockquote> für längere, <q> für kürzere, <cite> für Quellen Einfache & klare Sprache nutzen, kurze Sätze (6-12 W.), Alltagsvokabular; Problem: Messbarkeit Linguistische Analyse: Morphologie (Wortstruktur bei Änderungen); Lexikologie (keine seltenen Wörter, keine Mehrdeutigkeiten. Einfache Verben); Syntax (keine Verschachtelungen, Satzlänge begrenzen, einfache Sprache = 3 Substantive/Satz); Semantik ( wenige Präpositionen oder Pronomen); Diskurs (Verbindungen zu vorherigen Sätzen wie “daraus folgt”)
7	Tabellen	Nicht fürs Layout verwenden! Serialisierung durch Screenreader durch Autor zu unterstützen; <summary> für Zusammenfassungen od. <th abbr =“Kilometer”> Länge der Grenze in km </th>; Datentabellen verwenden <captions> <th> <td> Problem: mehrere Ebenen in Spalten und Zeilenüberschriften → Normalisierung um Verschachtelung mittels Markup aufzuheben, zusätzl. Überschriften nötig

8	Verweise & Navigation	<p>Warum kein Menütag im Screenreader? → nach Selektion müsste man Reaktion vorhersehen, bräuchte Verknüpfung von html zu Javascript; aber dafür nur nicht standardisierte Events → Lösung: in html5 Nahbereich mittels invisible Labels Klasse (inkl. Größe und Position)</p> <p>Navigation unsichtbar einbauen, &lt;li&gt; wird verbalisiert</p> <p>Aufgabengerechte Tab-Folge, Angabe des tab-index; niemals “mehr...” nutzen!, Trennzeichen zw. Horizontalen Gruppen;</p> <p>Verweise zu Inhalt unsichtbar gestalten (class hidden von Screenreader gesehen, sonst nicht)</p> <p>Navigation in Seite: Leisten verwenden, Gruppierungen, keine Ascii Grafiken, Bedienalternativen</p> <p>Benannte Anker: Sprünge zw. Abschnitten -&gt; für Screenreader Anfang &amp; Ende gut, Achtung verlassen der Seite</p> <p>Navigation zw. Seiten: zB Inhaltsverzeichnis, Sitemap – besser keine PopUps, informative Titel, Bookmarks erkennbar, Bedeutungsvolle Beschriftung</p> <p>Breadcrumbs trails: Weg als Struktur, fordern Fehlerrobustheit, aber ok für Screenreader</p> <p>Suche. Techniken in Suchmaschinen: Alternativtexte, keine unnötige Ausgabe, Suchoptionen, vermeiden weiterer naher Eingabefelder, Anpassung Trefferliste, Erläuterung, Hilfe bei erfolgloser Suche</p>
9	Grafik	<p>Unterlegung mit IMG, AREA, OBJECT</p> <p>Alternativtexte: alt=“...” → von Screenreader ignoriert! Alternativ: title für ausführliche Darstellung, longdesc nur teils beherrscht;</p> <p>Richtlinie Bildbeschreibung; 1-3 Worte, max. 150 Zeichen, kurze und lange Beschreibung bereitstellen</p> <p>Lange Beschreibung in &lt;p class= “portfolio”... → bettet Bild ein</p>
10	Audio	<p>Kanaltrennung Sprache vs Geräusche vs Musik (letzteres kursiv)</p> <p>Einfache Satzstruktur (Subjekt-Prädikat-Objekt), max 2 Zeilen, Rhythmus &amp; Dauer einhalten, Personentrennung mit Farbe</p> <p>Caption = Textbeschreibung, Subtitle = Übersetzung</p>
11	Video/ Animation	<p>Auch Mimik, Gestik, Emotionen beschreiben; Videos ohne Audio zusammenfassen; Animationen: pausieren ermöglichen für Zusatzzeit Erkundens (zB Vergrößern); mehrfache Videos seriell darstellen</p>

	Multimedia WCAG	<p>Audio: Alternative für zeitbasierte Medien bereitstellen, die äquivalente Infos für aufgezeichneten reinen Audioinhalt bietet</p> <p>Video: entweder Alternative für zeitbasierte Medien ODER Audiospur mit äquivalenten Infos für aufgezeichneten reinen Videoinhalt</p> <p>Untertitel: für alle Audioinhalte, außer diese sind bereits Medienalternativen, bei live auch Audiodeskription für Videos, Gebärdensynch. Für Audio</p> <p>Wenn Pausen in Vordergrundaudio für Deskription zu kurz, dann erweiterte Bereitstellung</p> <p>Kriterien Gebärdensprache: mind. Bildgröße Gebärdensprachfilme, nicht ruckartig, mind. Datenrate, Verständlichkeit: erkennbar, verständlich?, Gebärdensraum gesamter Oberkörper, Beleuchtung, Kontraste, beleuchteter Mundraum, keine graf. Elemente</p>
13	Komponenten/Plugins	<p>→ Erweiterungen vom Browser; Problem: Html kein Zugang für selfbrailling/selfvoicing</p> <p>Applets werden von Jaws per accessibility Bridge beherrscht – Seite soll auch ohne Applet bedienbar sein – AWT bietet weniger Unterstützung für Screenreader als Swing – Prüfung wie für DesktopApp erforderlich (Fokusverfolgung, Tastaturunterstützung, visuelle Unterstützung)</p> <p>Rolle Treeitem kennen, kenne Aria polite Konzept</p>
14	Interaktion	<p>Tastaturbedienung durch AccessKey, Nutzer muss aber informiert werden, welche Tasten bedienbar! Problem: verschiedene OS und Browser</p> <p>Ereignisbehandlung: log. Behandlung vs. geräteabhängige Behandlung</p> <p>PopUps vermeiden: da kein Kontext, Umgehung ermöglichen bei Einbettung</p> <p>Captchas: Telefon schlecht, besser Audio Captcha</p>
15	Formulare	<p>Problem: Verbindung zw. Beschriftung und Inhalt</p> <p>Dialogstruktur: Beschriftung und Bedienelemente assoziieren</p> <p>Vorbesetzen der leeren Texteingabefelder abhängig vom Screenreader</p> <p>Verschiedene Arten von SuFu vorsehen</p> <p>Fehlerbehandlung beachten: Problem: * nicht als Pflichtfeld erkannt, dann Fehlermeldung und mit Tab-1 navigieren</p> <p>Fokusverfolgung über mehrere Seiten nicht in Richtlinien behandelt</p> <p>Nutze Label!</p>

# 12. Dynamische Inhalte

## Screenreader & AJAX

### Dyn. Inhalte & Javascript

- dyn. Inhalte entstehen bei Verwendung von JavaScript (verwendet Ereignismodell)
  - Abhängig von Benutzereingabe: zB nach ersten 3 Zeichen Vorschlag
  - Unabhängig: Timer, dh man ruft Ereignisse programmiert auf, keine Möglichkeit Filter
  - Fast beliebige Abänderung des DOM (tags, label änderbar)
  - PROBLEM: keine Fokusverfolgung mgl, Tastaturbedienung oft nicht vorgesehen
  - Häufige Barrieren Javascript: ausklappbare Bäume = Listen, Orientierungspunkte mit Überschriften; od. zB Schieberegler mit generischen Elementen erstellt
- Asynchronous JavaScript and XML bezeichnet ein Konzept der asynchronen Datenübertragung zwischen einem Browser und dem Server. Dieses ermöglicht es, HTTP-Anfragen durchzuführen, während eine HTML-Seite angezeigt wird, und die Seite zu verändern, ohne sie komplett neu zu laden
  - → ermöglicht Dynamik (asynchrony), indem immer andere Server/ Clients kontaktiert werden
  - Screenreader: asynchrony. http xml request
  - Nachteile erfordern zusätzl Programmieraufwand (keine bookmarks oder history)
  - DOM in spez. Buffer verwaltet um Erkundung zu ermöglichen
  - Probleme: Fokus setzen nur extrem eingeschränkt mgl; Lösungsmglk: tabindex=-1, (de)aktivierung virt. PC Cursor Modus

# ARIA

- Bisher: Websites sollen ohne javascript bedienbar sein; Aber: Markup und Javascript sollen widgets zugänglich machen
- „Jedes Element oder Widget ist mit einer vollständigen und korrigierten Semantik gekennzeichnet, die sein Verhalten vollständig beschreibt (unter Verwendung von Elementnamen oder Rollen). Die Beziehungen zwischen Elementen und Gruppen sind bekannt Zustände, Eigenschaften und Beziehungen gelten für jedes Elementverhalten und sind über das DOM zugänglich. Es gibt ein Element mit dem richtigen Eingabefokus.“
- Aria= accessible rich internet applications(xml Sprache)
- **Landmarks:** semantische Auszeichnungstechnik für Navigation: leicht ergänzbar, erreichbar über Sondertasten, Warum? Einfach, nur drübergelegt, fügt Infos über Website hinzu, ohne Präsi für Sehende zu beeinflussen
- Technik des Roaming tabindex: setze tabindex für alle Elemente auf -1, bis eins 0 hat, dann korrigiere

# Aria Live Regions

- **aria-live="polite" oder aria-live="assertive"** Polite: soll alles oder nur eine Sache vorgelesen werden? Bei polite erst alles und DANN Neuigkeit
- aria-atomic: Das Attribut aria-atomic=BOOLEAN wird eingesetzt, um festzulegen, ob der Screenreader die Live-Regionen als Ganzes präsentieren soll, auch wenn sich nur ein Teil dieser Region ändert. Die möglichen Werte sind false oder true, wobei false der Default-Wert ist.
- aria-relevant: Mit aria-relevant=[LIST\_OF\_CHANGES] wird bestimmt, welche Art von Veränderungen relevant für eine Live-Region sind - die möglichen Werte sind additions/removals/text/all. Der Default-Wert ist "additions text".
- aria-labelledby: Mit aria-labelledby=[IDLIST] wird eine Region mit seinen Labels verknüpft. Die Technik ist dieselbe wie bei aria-controls, nur dass hier Labels statt Steuerungselemente mit der Region verknüpft werden. Mehrere Bezeichner können durch Leerzeichen getrennt angegeben werden.
- aria-describedby: Das Attribut aria-describedby=[IDLIST] wird verwendet, um eine Region mit einer Beschreibung zu verknüpfen. Auch hier ist die Technik dieselbe, wie bei aria-controls, nur dass eine Beschreibung statt einer Steuerung verknüpft wird. Mehrere Bezeichner für Beschreibungen können durch Leerzeichen getrennt angegeben werden.
- `<ul id="roster" aria-live="polite" aria-relevant="additions removals">`  
Hier wird JavaScript zum Hinzufügen und Entfernen von Benutzern eingesetzt --> `</ul>`



- **WAI-ARIA** (**Web Accessibility Initiative – Accessible Rich Internet Applications**) ist eine Initiative zur Verbesserung von Webseiten und [Webanwendungen](#), um sie für Menschen mit Behinderungen besser zugänglich zu machen, insbesondere für blinde Anwender, die [Vorleseprogramme](#) verwenden.
- ARIA ist eine technische Spezifikation, die von Mitgliedern der [Web Accessibility Initiative](#) entwickelt wurde. Seit März 2014 ist ARIA ein empfohlener Webstandard des [World Wide Web Consortium](#) (W3C).
- **Konzept und Funktionsweise**
- ARIA verwendet die Techniken [JavaScript](#) und [Ajax](#). ARIA ist eine rein [semantische](#) Erweiterung für [HTML](#), die das Layout einer [Webseite](#) nicht verändert. Die [Barrierefreiheit](#) dynamischer Seiten wie im [Web 2.0](#) mit seinen [Rich Internet Applications](#) und die allgemeine [Benutzerfreundlichkeit](#) können so verbessert werden.
- ARIA ermöglicht es Webseiten (oder Teilen einer Seite), sich als [Anwendungen](#) zu bezeichnen anstatt als statische Seiten. Dazu werden in dynamischen Webanwendungen Informationen zu Rollen, Eigenschaften und Zuständen hinzugefügt. ARIA ist zur Benutzung durch Entwickler von Webanwendungen, [Browsern](#), [assistiven Technologien](#) und Programmen zur Verifizierung von [Barrierefreiheit](#) vorgesehen.
- WAI-ARIA besteht aus vier Komponenten:
- **Landmark Roles** erlauben die semantische Zuweisung einer Rolle zu HTML-Konstrukten. Dadurch kann für Screenreader die Aufgabe eines Oberflächenelements kenntlich gemacht werden, die sich aus den HTML-Elementen selbst nicht erschließt. Beispiele sind [Slider](#) (Schieberegler) oder [Bäume](#). Für einige dieser Rollen gibt es seit [HTML 5](#) auch dezidierte HTML-Elemente.
- **ARIA-Attribute** ARIA definiert einige zusätzliche [Attribute](#) wie `aria-required` oder `aria-invalid`, die sich für alle HTML-Elemente verwenden lassen. Sie lassen sich beispielsweise dafür verwenden, den Inhalt eines Eingabefeldes als ungültig zu markieren, etwa wenn in einer E-Mail-Adresse kein `@`-Zeichen vorkommt oder zwei Eingaben eines Kennworts (zur Bestätigung) nicht übereinstimmen.<sup>[2]</sup>
- **Live Regions** sind Teile einer Seite, die sich in unregelmäßigen Abständen aktualisieren. Diese Veränderungen können bei implementiertem ARIA von Screenreadern automatisch erkannt und gesprochen werden.
- **States und Properties** werden für richtige JavaScript-[Widgets](#) verwendet (wie beispielsweise einer aus `div`-Elementen bestehenden Liste von Optionen), um semantisch bedeutsame Eigenschaften des jeweils aktuellen Zustands auszuzeichnen. Beispielsweise muss die Tastaturnavigation inklusive der Hervorhebung des gerade aktiven Elements bei eigenen JavaScript-Widgets selbst implementiert werden. Damit die Information, welches Element gerade aktiv ist, nicht nur optisch durch Hervorhebung, sondern etwa auch Navigationshilfen für Sehbehinderte zur Verfügung steht, kann mit `activedescendant` das aktuell fokussierte Element ausgewiesen werden. ARIA stellt als semantische Erweiterung einen Standard für die Auszeichnung solcher Informationen zur Verfügung.



# Genauere Prüfungsfragen

- Barrieren im Web (freie Auswahl, Dinge nennen die man auch genau erklären kann).
- Aufzählen von Behinderungen und Erklären an Beispielen, welche Probleme bei Browsern auftreten und Lösungen aufzeigen
- Genauer: Was macht man bei Bildern? Wie mit Karten umgehen?
- Was soll im Alt-Tag stehen?
- Kurz und knapp oder gar nichts bei Dekoration.
- Was würden sie bei einem Bild von einer Karte der TU Dresden machen?
- Lange Beschreibung "longdesc" bzw. Verweis auf andere Seite (Hat ihm nicht richtig ausgereicht.)
- Ausdruck als taktile Grafik (Er wollte eher digital bleiben. ;) Vllt. gibts hier noch was besseres.)

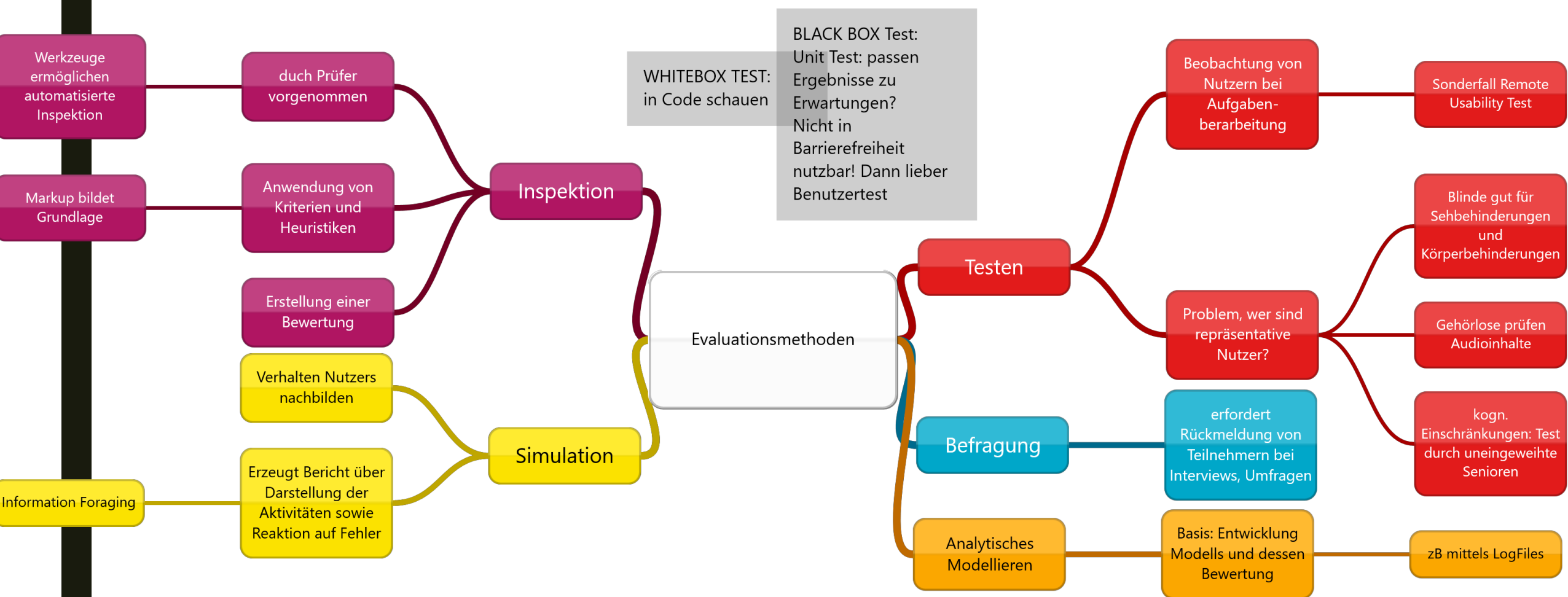
# Fragen zu Medien

- Was ist bei **Videos** ? Wie dort Barrieren lösen? Für wen sind Untertitel? Formate nennen?
- - Untertitel im Web; Unterschied captions und subtitles
- **Tabellen**: Probleme, Leerfelder, Wie lösen?
- Barrierefreie Tabellen (er hat eine aufgemalt mit Leerzelle, mehrfachen Zelleninhalten und verbundenen Zellen: erklären wo Barrieren liegen und Lösungen)
- Was ist der Unterschied zwischen **Hörbüchern** und **Audio/MP3**- Dateien ?
- - Unterschied mp3-Hörbücher zu Daisy-Hörbücher als Einleitung
- - Erklärung Aufbau daisy. SMIL, NGX, ...
- - Was macht der <spline> - Tag in den ncx (navigation control) - Format? (siehe Daisy Hörbücher)
- Was ist mit JavaScript? (Probleme, Wie kann man diese Barrieren lösen?)

- Inspektion für Javascript
- ■Firefox Accessibility Extensions (FAE)
- ■[http://www.accessfirefox.org/Firefox\\_Accessibility\\_Extension.php](http://www.accessfirefox.org/Firefox_Accessibility_Extension.php)
- ■Untersuchung der ARIA Widgets (Role, Tab Index, Wert)
- ■Unsichtbare Navigationsleisten werden einsehbar
- ■Bsp: Karteikarten auf bahn.de
- ■Vorsicht: tab Index wird dynamisch geändert
- ■Fokusverfolgung
- ■Benennen der Änderungen

- Evaluationsmethoden (manuell, automatisiert)
- Javascript
- Crawling
- Beschreibungssprachen
- WAQM
- Simulation

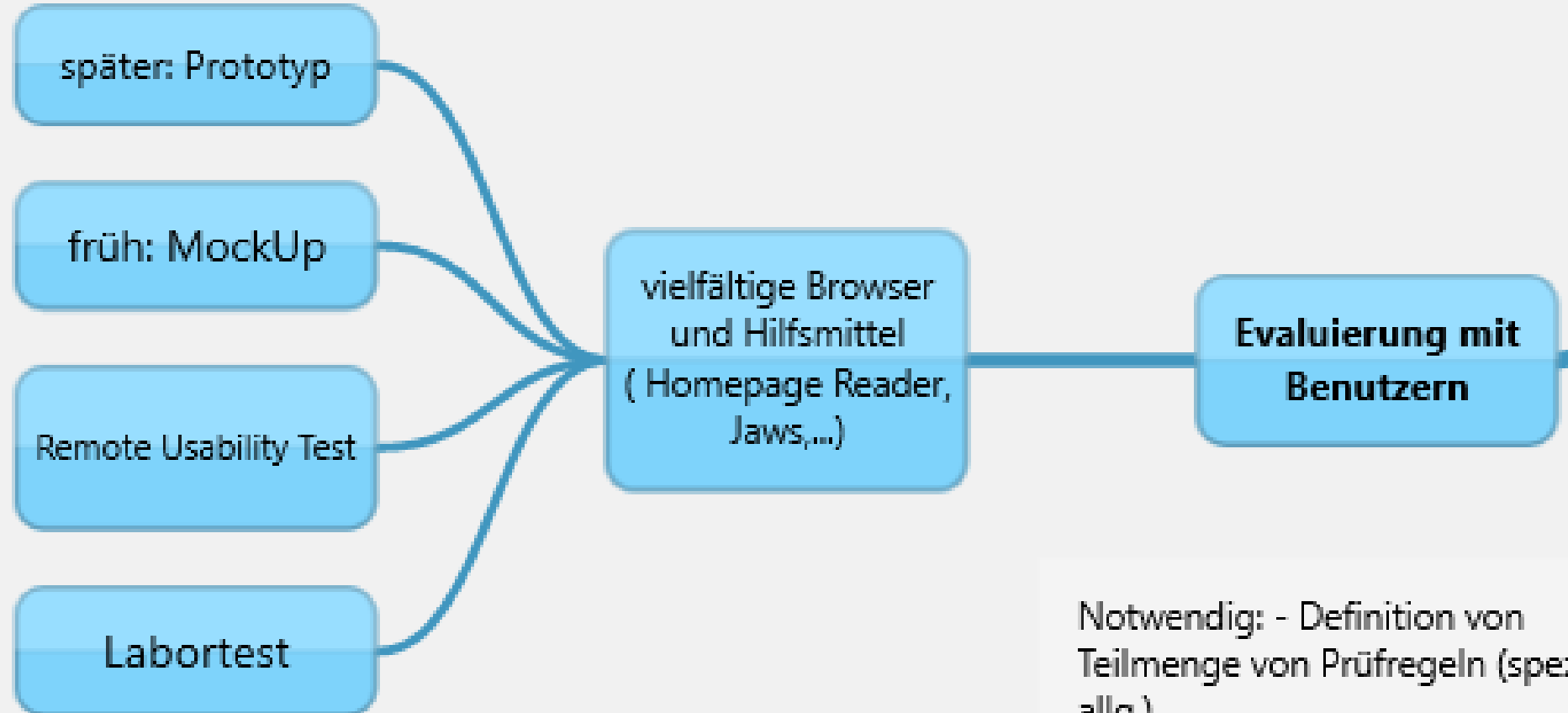
## 3. Prüfen von BF auf Websites



# Evaluierung mit Benutzern

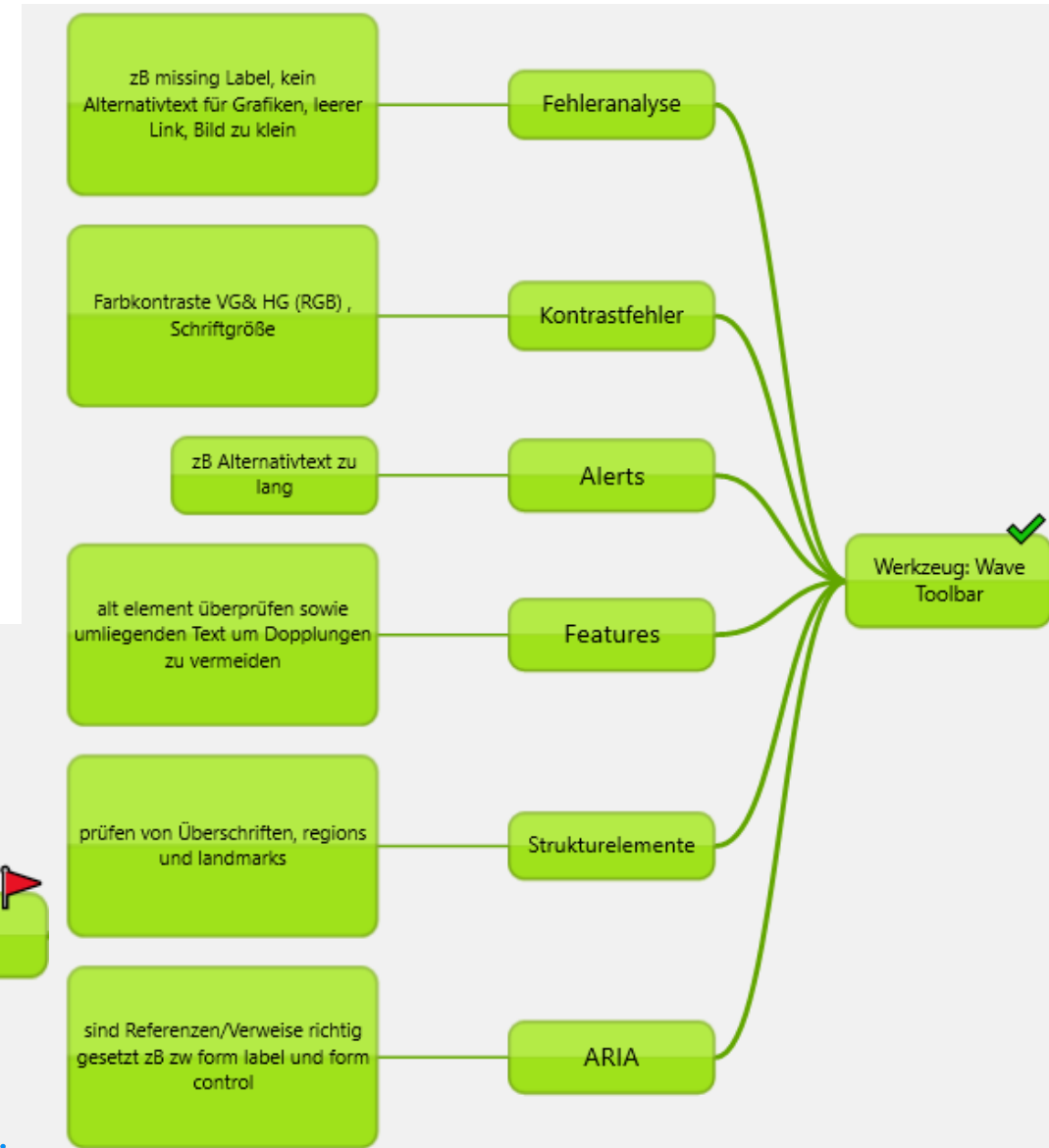
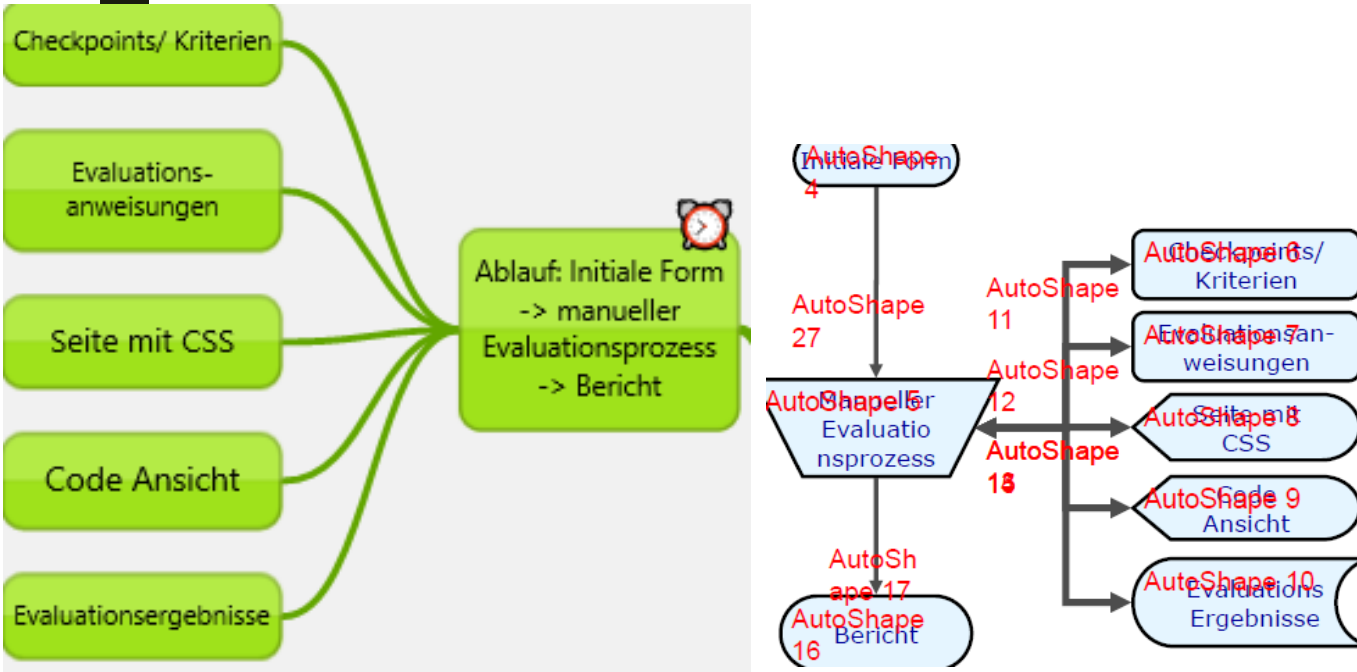
## BARRIEREFREIHEIT EVALUIEREN

Umfang festlegen: Was gehört dazu, Ziele der Evaluation, Konformitätsstufen (A bis AAA)



Notwendig: - Definition von Teilmenge von Prüfregeln (spez./allg.)  
- Eineindeutige Referenz auf Testobjekt (stat./dyn.)  
- Berichtsgenerierung (indv./aggregiert; zB Anordnung Suchergebnisse bei Google)

# Manuelle Evaluation



# Automatische Evaluation

## Werkzeugarten

Evaluationswerkzeuge

statisches Untersuchen,  
allg. nach Richtlinien oder  
spez. zB nach Kontrasten

Reparaturwerkzeuge

in html Text eingreifen  
mgl

Filter-u. Transformations-  
werkzeuge

direkte Kritik und  
Unterstützung beim  
Nutzer

Prüfbare Aspekte

Navigation: nur Aspekte  
der Tastaturbedienung

Schwerpunkte: Codierung  
oder vis. Kriterien

## Fehlerquellen

Fehlerberichte zu viele  
oder zu wenige

Bsp Sprachumfang

HTML Erweiterungen  
nicht geprüft

SMIL, TIME, SVG

meist nicht für gesamte  
Website anwendbar

derzeit nicht  
automatisiert prüfbar:

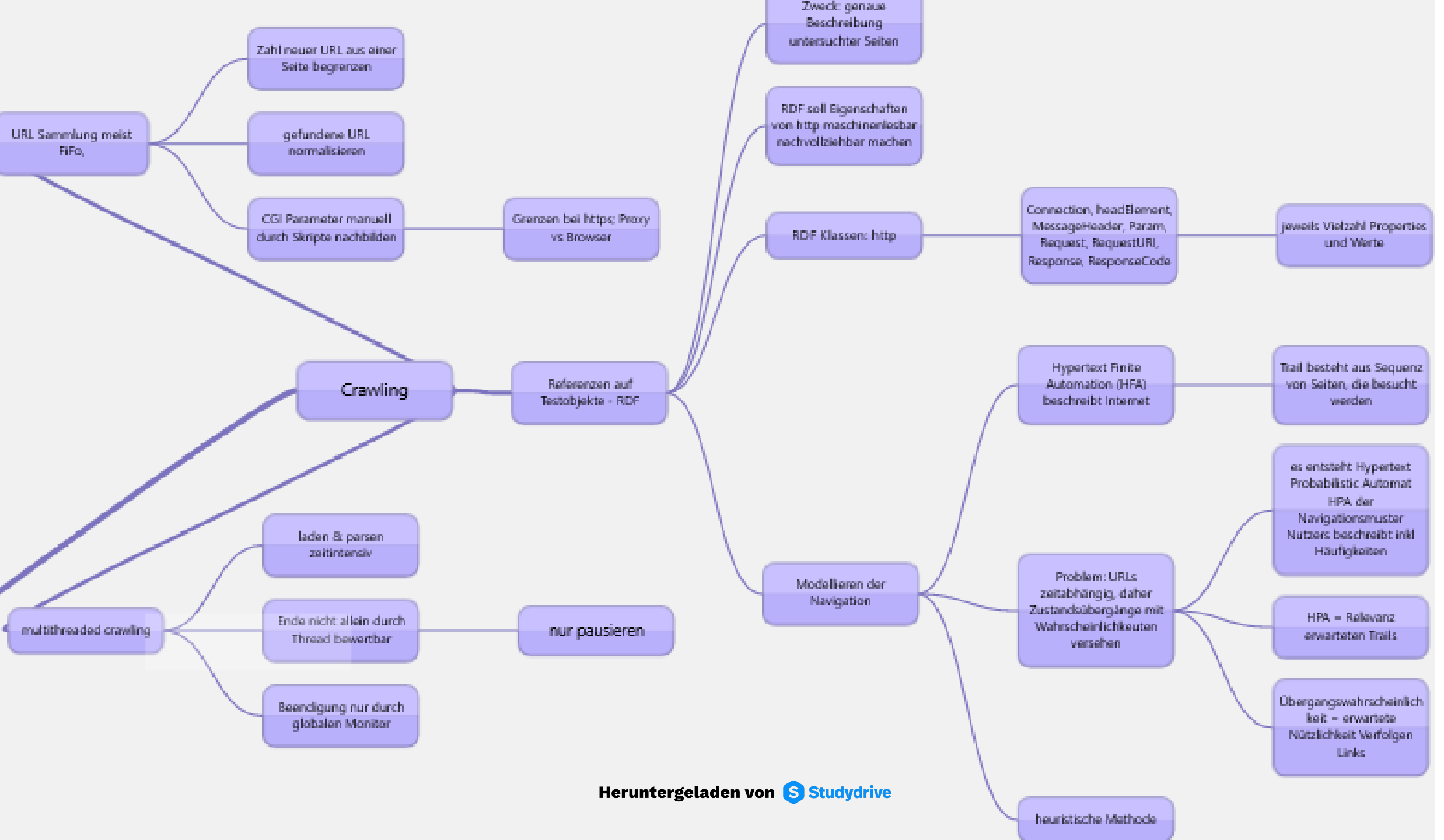
Naviagtion; einfache  
Sprache; Einfluss  
Hilfsmittels

Monitoring von Barrieren  
über Zeitraum

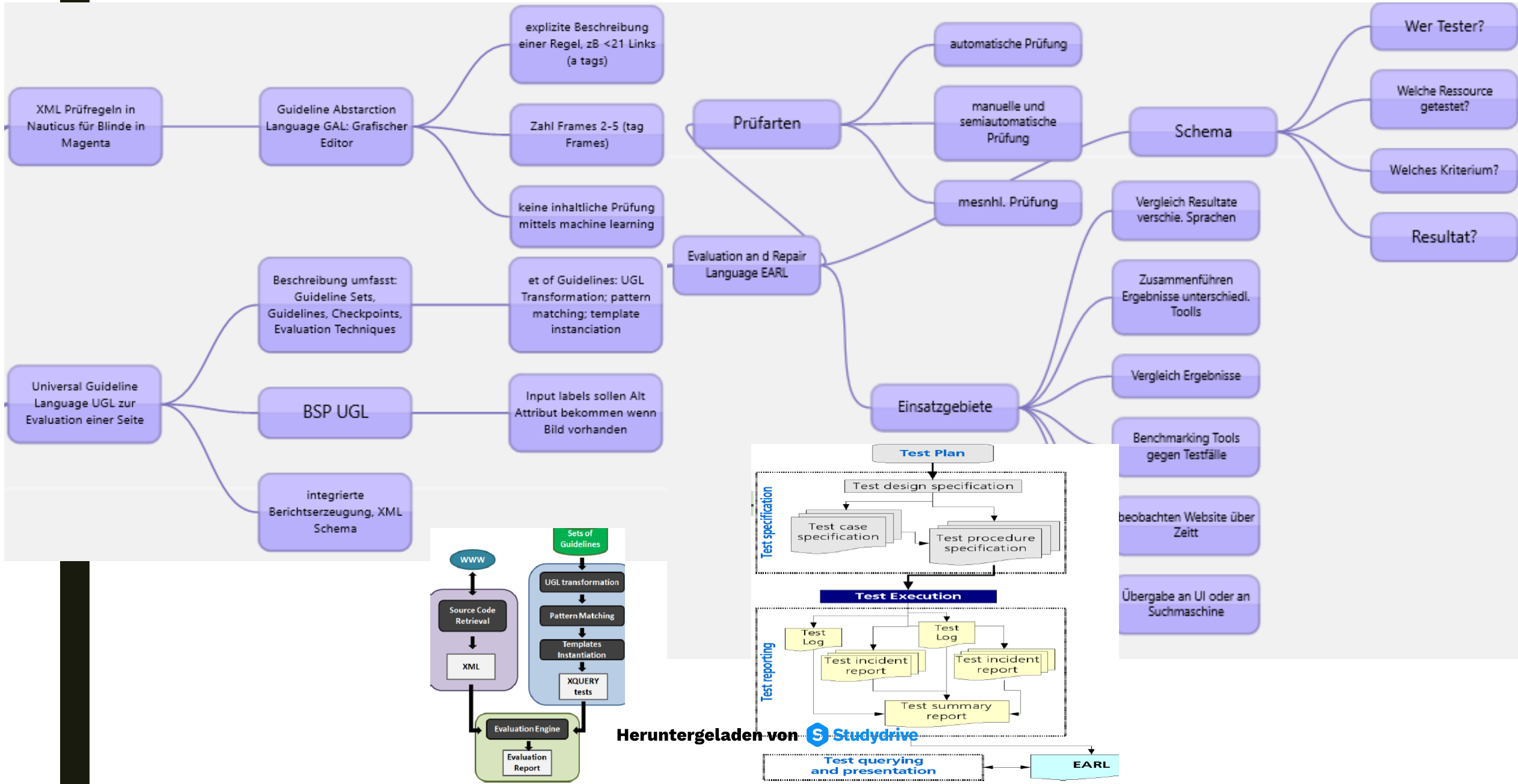
unterstützen skripting,  
Gamification, BITV Test,  
WCAG

Bsp: Mauve: erlaubt  
eigene Regeln anzugeben



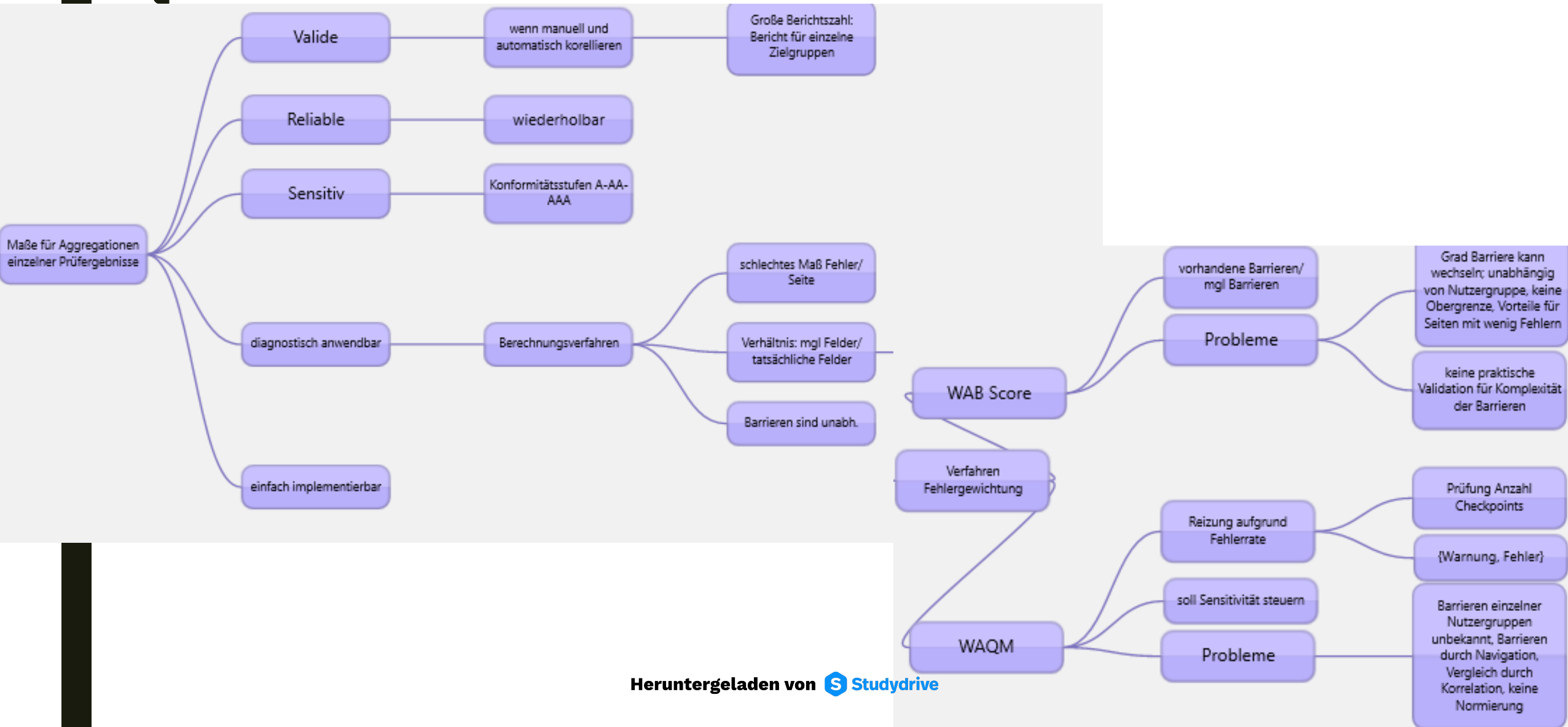


# Automat. Ev.: Beschreibungssprachen 1

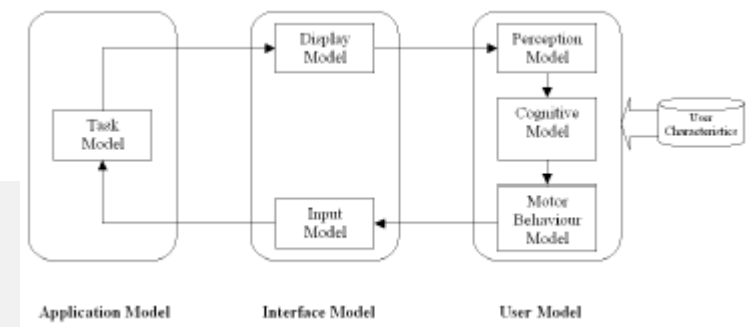


# Automat. Ev.: Beschreibungssprachen 2

## Quantitative Metriken



# SIMULATION



## Simulation

Zweck: bildet Wahrnehmungsdefizite nach, zB Seh-od Hörbehinderung

im allg. nicht gelöst, Versuch durch Filter statt Ausschalten

Modellierung der Infosuche durch Markov Ketten bei einfachen Geräten

aDesigner: zB Buttons und andere Visualisierungen

Grenze: Navigation nicht vorausgesagt

NetbeansIDE simuliert Netzhautveränderungen, Parkinson, Dyslexie,...

Sim

Sim. physikalischer Behinderung

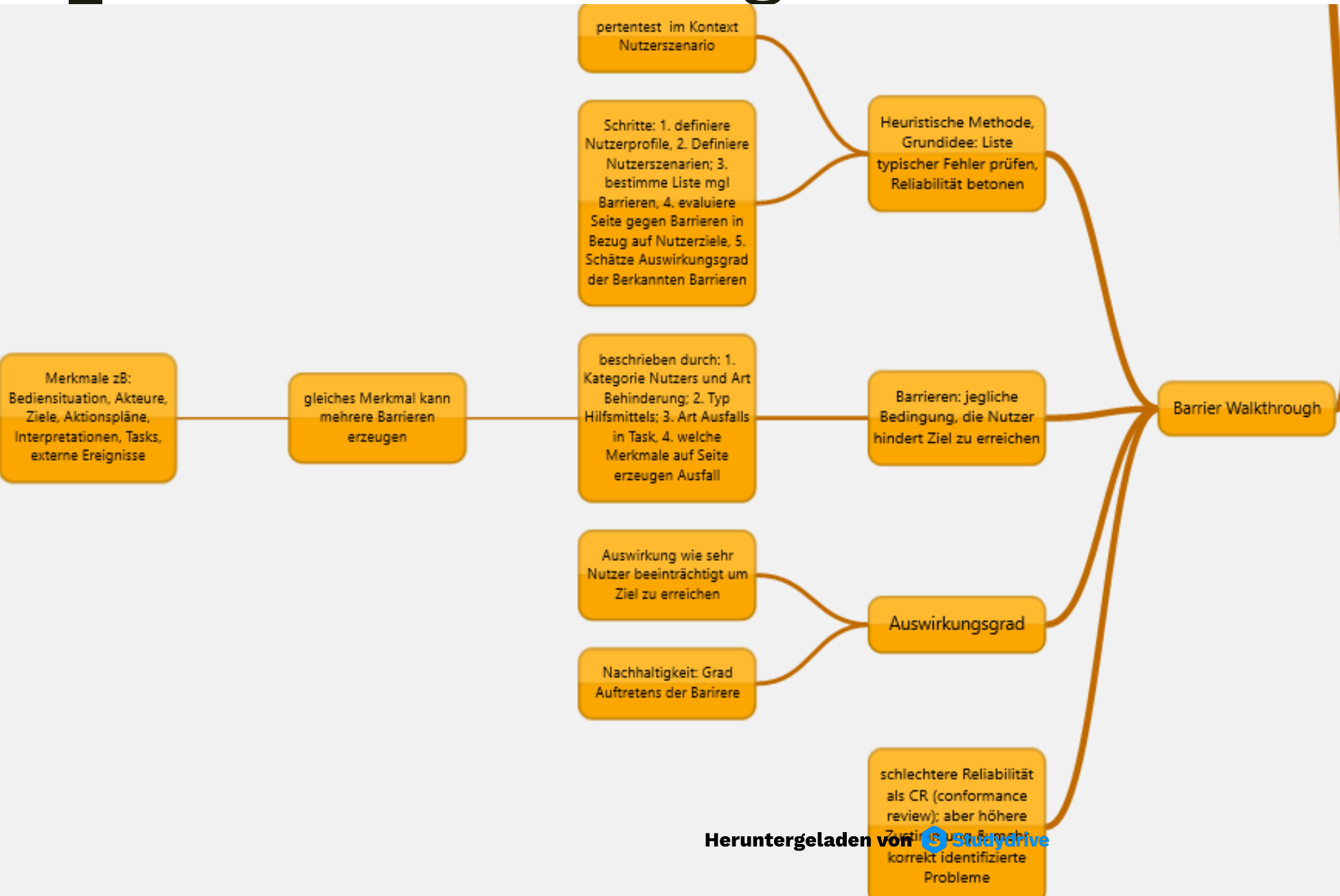
nutzt sequentielle Tastatur (scanning Keyboard)

Basis: visuelles Wahrnehmungsmodell

8 Richtungen berücksichtigt

nur von 8 Nutzern getestet, Benutzertests mit Braillezeile besser als Simulation

# Barrier Walkthrough



# Evaluation

- Wie kann man Barrierefreiheit evaluieren? (manuell, automatisch, Simulation, Evaluierung, ...)
- - wie sieht Simulation aus? Was kann man simulieren?
- - wie lassen sich Resultate verschiedener autom. Evaluations-Tools vergleichen bzgl Barrierefreiheit von Websites ? → EARL
- -dann was bei Opal Anforderungen an eine automatisierte Evaluation (Crawling, Testdaten für Befüllen Opal-Tests notwendig)

# MATHEMATIK

## Entwurfskriterium

### Allgemeine GUI Kriterien

### Mathematik Interaktion

#### Kohärenz

- Layout
- Struktur u. präsentierte Zeichen

- Ausgabe-und Eingabemodalität sollen sich entsprechen
- ZB Braille& Braille oder Sprache&Sprache

#### Erkundung

- räuml. Od. Hierarchische Navigation je nach Art Interaktionsobjekts

- räuml. Od. Hierarchische Navigation je nach math. Konstrukt

#### Graf. Symbole

- Durch Namen verbalisieren

- Verbalisieren durch Namen od. Nicht verbalisierbare Klänge

#### Lernbarkeit

- Erfahrung mit text-basierter Benutzungsoberflächen anwenden

- Einsatz existierender Braille-Notation od. Natürl. Sprache

#### Adaptierbarkeit

- Individualisierbarkeit

- Einsatz der Braille-Notation/ Sprechweise je nach Kenntnissen Lesers



<b>Braille &amp; Markup</b>	<ul style="list-style-type: none"> <li>- 6-8 Pkt, unterschiedliche Berücksichtigung von Klammern;</li> <li>- Problem Überführung Marburger Notation in andere zB AMS;</li> <li>- Grenze: Operatoren ständig neu definierbar</li> </ul>
<b>HTML &amp; Mathe</b>	<ul style="list-style-type: none"> <li>- HTML nicht für Mathe vorbereitet; Image Anweisung + render schlecht, besser: Bild nur zusätzlich anbieten, aber Formel in Alt Tag</li> </ul>
<b>Braille &amp; Computer</b>	<ul style="list-style-type: none"> <li>- Auf Papier: keine Korrektur, keine Struktur, mech. Handhabung</li> <li>- Kohärente Verarbeitung erfordert Brailletastatur → Folge: Textverarbeitung auch mittels Brailletastatur</li> <li>- Math. Ausbildung macht kaum Gebrauch von Computern</li> <li>- Math. Braille für Sehende unzugänglich; Spezialschulen: Latex, Integrierend: AMS, SMSB</li> </ul>
<b>Math. Editoren</b>	<ul style="list-style-type: none"> <li>- Problem für Blinde: Mathe als Grafik behandelt</li> <li>- Schreiben u. editieren?</li> <li>- Ziel: Einsatz von Standards und Zugang für Blinde</li> </ul>
<b>Braille, Sprache, Klänge</b>	<ul style="list-style-type: none"> <li>- Dyn. Braille: Terme ersetzen durch Termbegriffe</li> <li>- Sprachausgabe: mit korrekter Prosodie, dhzB links und rechts vom Gleichheitszeichen Pausen angeben; auch bei Brüchen; 16 Tonhöhenstufen</li> <li>- Klangausgabe zum Überblick, begleitend zur Kontrolle</li> </ul>
<b>Beispielausdruck</b>	<ul style="list-style-type: none"> <li>- Baumstruktur ausbilden: Mathe <math>\rightarrow</math> x und <math>\rightarrow</math> rechter Bruch; diesen in <math>\rightarrow</math> oben und <math>\rightarrow</math> unten</li> </ul>
<b>Lambda Projekt</b>	<ul style="list-style-type: none"> <li>- Zwischensprache zw Braille und MathML, Echtzeit und graf. Transformierbar</li> </ul>
<b>OCR Analyse</b>	<ul style="list-style-type: none"> <li>- Analysieren von Bildschirmhalten durch Schrifterkennung; erzeugt aus Pixeln</li> <li>- Ausgangspunkt: Papier oder PP Folien mit Formeln <math>\rightarrow</math> Screenshot <math>\rightarrow</math> Latex generieren</li> </ul>

# MATH ML

Mathematical Markup Language	<ul style="list-style-type: none"><li>- Math. Doks haben vielfältige Strukturen da sowohl menschl. Als auch maschin. Erstellung</li><li>- <b>Presentation Markup</b>: etwa 30 Elemente u. 50 Attribute, sowie Katalog math. Symbole (Token Elemente; mi für Identitäten/Variablen, mo für Operatoren, mn für Zahlen)</li><li>- <b>Content Markup</b>: 100 Elemente und 10 Attribute, beschreibt Funktionen; zB sin, plus, set, vector; ideal für Sprachsynthese</li></ul>
Presentation vs Content	<ul style="list-style-type: none"><li>- msuphat zwei Argumente: Basis und Exponent; mfencedbeschreibt die Klammerung</li><li>- applyhat zwei Argumente: eine Funktion und deren Argument</li><li>- Msup → apply nicht umwandelbar wegen Klammern; je nach Stärke der Bindung umgekehrt gut mgl., a &amp; b sind Identitäten oder Argumente, mi ungleich ci!</li></ul>
Box Model	<ul style="list-style-type: none"><li>- Ähnlich Baumstruktur: Ziel: visuelle Zusammenhänge in Boxes abbilden</li><li>- Aber: was ist „integral x d x von x= null bis unendlich“?</li></ul>
Layout steuern	<ul style="list-style-type: none"><li>- Indizes: hochgestellt msup; tiefgestellt msub</li><li>- Balken und Pfeile unten: munder, oben mover</li><li>- Allg. Algebra: mmultiscripts</li></ul>
Content Markup	<ul style="list-style-type: none"><li>- Apply = Funktion anwenden;</li><li>- Token Elemente ci, cn – Container bzw Konstruktoren</li><li>- Mengen= sets mit conditions, Intervalle mit Attribut closure, Vektoren; Matritzen</li></ul>
Grenzen erweitern	<ul style="list-style-type: none"><li>- Presentation Markup: visuelle Darstellung – gemischt PM und CM erläutert jeweils eine Darstellung</li><li>- Schwierig: maschinelle Lesbarkeit; besser Menschen einsetzen</li><li>- Neue Operatoren nicht prozedural beschrieben; aber es gibt Erweiterungsmechanismen</li></ul>
MathML im Browser	<ul style="list-style-type: none"><li>- W3C Browser Amaya unterstützt MathML</li><li>- Alternative: XSLT</li></ul>

- Mögliche Wahl zwischen MathML und Web (ich wählte Web, obwohl Prof. Weber dann trotzdem kurz auf MathML einging, aber nur erzählte und nicht viel wissen wollte)
- Mathematische Formeln als Bild -> Wie funktioniert MathML?
- 4.2. Wie sieht es bei mathematischen Formeln aus?
  - *Einfach: Latex in alt-Tag*
  - *Erweitert: MathML (Content & Presentation Markup)*
- 4.3. Hier wollte er nochmal wissen, wie man Gleichungen umsetzt.
  - *Funktion-Tags + ci / cn*

# MULTIMEDIA BARRIERE

# Personalisierung

- Problem: Bedienelemente (buttons) größer in Videoabspieler; Screenreader versagt, keine Untertitel oder Gebärdensprache
- → Multireader
- Benutzer haben verschiedene Anforderungen, zB Blinde (Geschwindigkeit, Pausen steuern; Bildbeschreibungen, Audiodeskriptionen); Gehörlose (textbasierte Beschreibungen; Bilder & Filme statt Text; Gebärdensprachlexikon; kurze Texte, dyn. Hervorhebungen synch. Mit Sprache, var. Zeichensätze Farben Abstände); Dyslexiker (var. und dyn); Sehbehinderte (var, Vergrößerungen, Sprache)
- Zugang zu Büchern: Anreicherung mit PDF, Großdruck, Braille, Sprachsynthese, Druck, HTML & CSS
- Bsp Gehörlose: Erst Video, dann Video mit Gebärden, und parallel Text hervorhebungen in Zsmfassung

# Multireader Dokumente

- Ein MR Dokument ermöglicht die Betrachtung eines personalisierten Transformationsergebnisses des Inhalts durch
  - Auswählen des Inhalts (Video mit Gebärdensprache, Zusammenfassung) basierend auf einem Editor für die gewünschten Eigenschaften,
  - adaptierbaren Sichten (Schriftgröße, Farbe, temporale Adaptierungen) des angereicherten Inhalts, und
  - Navigationstechniken basierend auf semantisch modellierten Interaktionsobjekten
- Für den Einsatz ist zu unterscheiden
  - Server-basierte Anpassung (transaktions-basierte Verarbeitung der Profildaten)
  - Client-basierte Anpassung (lokale Anwendung der Profildaten)

## MarkUp für MRDocs

- Ein „Container“ identifiziert redundante Medienobjekte → dh man benutzt Klassen aus Javascript für semantische Typisierung
- Meta-widgets für Navigationstechnik
  - *nächstes/voritges/Start*
  - *Inhaltsverzeichnis*
  - *Index*

Inhalte so zsmstellen, dass barrierefreies Buch ergibt.  
Problem: Versuchsleiter macht Einstellungen, nicht Nutzer

### Layout

- Starke Trennung von Inhalt und Präsentation, Layout per CSS (außer temporales Layout)


### Personalisierung

- *MS Windows Profile*
- *Hilfsmittelprofile*
- *Stereotypen*
- *individuelle Einstellungen*
- *(Identitätsmanagement)*

# Benutzerprofile

- Dienen der Erstellung adaptierbarer und adaptiver Systeme
  - symbolische Merkmale
  - statistische Merkmale
  - Kategorienbildung und -verdichtung nach Bayes
  - Kollaborative Systeme verdichten Benutzerprofile
- hier: Benutzerprofil mit behinderungsspezifischen Merkmalen
- Problem des Systems: Datensicherheit erhalten vs. Barrierefreiheit
- Problem der Benutzer: Verknüpfung von Usability und Accessibility (Needs and Preferences)

## EN 1332-4 (2000)

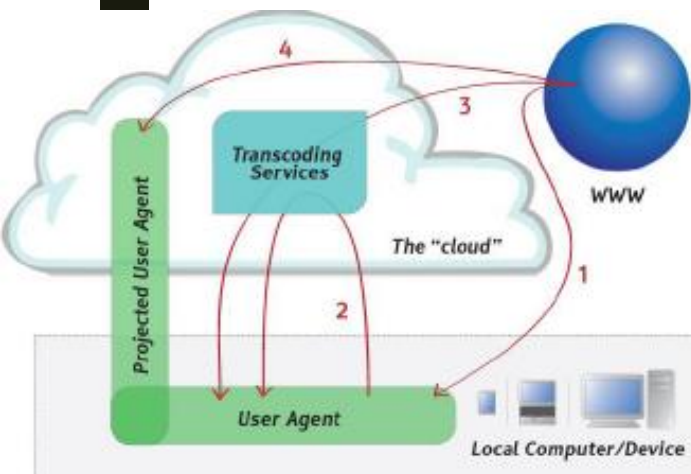
- ISO Standard legt Benutzerprofile für Identifikationskarten fest
- In sicheren Umgebungen einsetzbar
- Benutzer-Merkmale sind Anforderungen
  - *Buchstabenhöhe (2 BCD Stellen)*
  - *Bildschirmfarbe (1 Byte für Text und Hintergrundfarbe)*
  - *Farbvermeidung (1 Byte für spezifische Farben z.B. rot, rot/grün, grün/gelb)*
  - *Ca. 20 Merkmale zur Bildschirmtastatur, Tonhöhen, Spracheingabe, einfacher Sprache*
- **Heruntergeladen von  Studydrive**  
*Nachteil: zu wenig Profilinfos*

# Infrastruktur für Inklusion

## Global Public Inclusive Infrastructure GPII

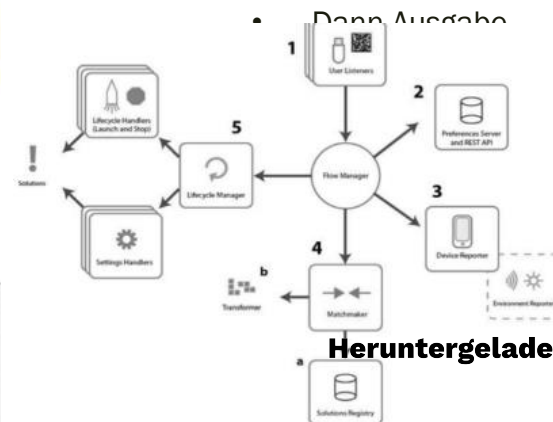
Man bietet auf Server im Web User Agent an = Browser zum Runterladen → hat Erweiterungen die es erlauben Inhalte runterzuladen und anzupassen

Client wie Browser, Screenreader, andere assistive Technologien



## Regelbasierter Ansatz für Bedarfe

- Präferenzen P1 ... P8 des Benutzers werden anhand existierender Regeln als Bedarfe B1 ... B3 bestimmt (Eingabe & Schlussfolgerung)
- mit den Merkmalen der verfügbaren assistiven Technologien (solutions) S1 ... S5 verglichen. (Konfliktdetektion)
- Dabei können K1...K2 Konflikte entstehen und daraus Konfliktlösungsarten KS1 ... KS4 abgeleitet werden. (Konfliktreduktion)



Heruntergeladen von Studydrive

## Konflikte in Profilen

- Konflikte entstehen durch
  - andere Präferenzen des selben Bedarfs (z.B. 12 pt vs. 18pt Font)
  - semantisch relevante Bedarfe anderer Benutzer

## KS1: Benutzer wählen aus Empfehlungen

- Wichtige Impulse zur Akzeptanz von Empfehlungen sind
  - Bewusstsein,
  - Kennenlernen und
  - soziale Einflüsse
- Bewusstsein und Kennenlernen erfordern barrierefreie Präsentation und Interaktion mit der Anwendung



# Kollaborative Barrierefreiheit

## Crowdsourcing

- Nach Surowiecki, 2004 entsteht C.S. wenn
  - Vielfalt der Meinungen,
  - Unabhängigkeit,
  - Dezentralisierung und
  - Aggregation möglich werden
- Kollaborative Barrierefreiheit wurde von Takagi, et.al. 2008 für die Beseitigung von Barrieren in Webauftritten entwickelt (Proxy-Server)
- in einem Feldtest sind 2009 innerhalb weniger Tage ca. 2000 Reparaturen durchgeführt worden

## Karten

- Kollaborative Bf.: Barrierefreiheit wird durch Beteiligung anderer Menschen hergestellt
- Ziel hier: bessere Karten für Rollstuhlfahrer
- Mehrstufige Bewertung:
- zugänglich (grün), teilweise zugänglich (gelb), unzugänglich (rot), unbekannt (grau)

## Zukünftige Probleme

Kollaborative Barrierefreiheit ist ein noch junges Gebiet in dem es viele Probleme zu lösen gibt:

- Kosten
  - *Systemkosten*
  - *Kosten für Benutzer (z.B. iPhone ist notwendig)*
  - *Kosten für Entwicklung und Nachhaltigkeit*
- Aufbau eines Pools von Bearbeitern
- Einbeziehung von Menschen mit einer Behinderung
  - *mehr zu kollaborativen Erstellung taktiler Grafiken (Projekt Tangram) später in dieser Vorlesung*
- Messung der Qualität eines Service
- Hybride Systeme mit besserer Automatisierung

# PDF

# Page Objects

- 10.1.2008: PDF 1.7 wird ISO Standard

- Dokumente enthalten Dictionaries, Page Objekte und Aktionen
- Strukturen vererben Attribute

- meist hierarchisch

Seiten enthalten Elemente; Seiten sind Grundelemente, darin Objekte

- Pfadobjekte: Beliebige Kombination aus Geraden, Rechtecken und kubischen Bezierkurven. Kann als Clippingpfad benutzt werden
- Textobjekte: Kombinationen aus mehreren Buchstaben. Textobjekte können gezeichnet, gefüllt oder als Clippingpfad benutzt werden.
- Externe Objekte (XObjects): Externe Objekte werden außerhalb des Content-Streams definiert und können anschließend innerhalb eines Content-Streams verwendet werden. XObjects werden hauptsächlich dazu benutzt, Grafiken in PDF einzubinden.
  - *Bilder zusammen mit einer Transformationsmatrix zur Anpassung des Koordinatensystems bzgl. des graphic states . Binärdaten transcodieren die ASCII-Filter ASCIIHexDecode und ASCII85Decode bzw. verwenden die Dekompressionsfilter LZWDecode , FlateDecode , RunLengthDecode , CCITTFaxDecode und DCTDecode*
  - *Forms sind aus Postscript übernommene Datenstrukturen zur Wiederverwendung*
- Inline-Images: Eine Möglichkeit um kleine Grafiken innerhalb von PDF einzubinden.
- Shading Objekte: Shading Objekte bestehen aus einem beliebigen Umriss, wobei die Farbe abhängig von der Position innerhalb dieses Umrisses bestimmt wird. Ein Shading Object könnte z.B. verwendet werden, um Farbverläufe darzustellen.

- Interaktive Elemente wie z.B. ein hypermediales Inhaltsverzeichnis
- Annotationen
  - *Text Annotation: Die Annotation wird im geschlossenen Zustand als Icon dargestellt (Kategorien Comment , Help oder Note)*
  - *Free Text Annotation: ständig auf der Seite angezeigt*
  - *Line Annotation: eine einfache gerade Linie*
  - *Square und Circle Annotation: Im geschlossenen Zustand wird diese Annotation durch ein Rechteck bzw. eine Ellipse dargestellt, die den der Annotation zugewiesenen Bereich einnimmt.*
- Verweise (Hyperlinks)
  - *Go-To-Action*
  - *Remote-Go-To-Action*
- File-Attachment-Annotation
- Audio-Annotation und die
- Video-Annotation

# PDF/A= archivierbar

- Portable Document Format(PDF) 1.4 zur Langzeitarchivierung von elektronischen Dokumenten
- geräteunabhängig
  - Kann verlässlich präsentiert werden, ohne von der HW/SW abhängig zu sein
- abgeschlossen
  - Enthält alle Ressourcen die für den Renderer notwendig sind
- selbstdokumentierend
  - Enthält seine eigenen Beschreibungen/Metadaten (XMP)
- transparent
  - Zugänglich für unmittelbare Auswertungen mittels einfachen Werkzeugen
- keine technischen Schutzmassnahmen
  - *keine Verschlüsselung, Passwörter, usw.*
- offen
  - *autorisierte Spezifikation ist öffentlich verfügbar*
- eingesetzt
  - *verbreiteter Einsatz dürfte der beste Schutz vor Verlust sein*
- Grenzen von PDF/A
  - *kann nicht allein die Langzeitarchivierung ermöglichen*
  - *ist noch nirgends gerichtsfest*
    - dt. Projekte: ArchiSig, TransiDoc

## PDF/UA= nur Aufbau, nicht API, nicht unbedingt archivierbar

- ISO/DIS 14289-1 - Dokumentenverwaltungsanwendungen - Erweiterung des elektronischen Dokumentendateiformats für Barrierefreiheit - Teil 1: Verwendung von ISO 32000-1 (PDF/UA-1), 2009 und 2014
- Bf. wird durch Prüfprotokoll (Matterhorn) abgesichert (2015)
- Ca 130 Prüfpunkte, 31 Gruppen spez. Inhaltsarten
  - *mathematische Formeln*
  - *laufende Seitenüberschriften und -unterschriften*
  - *Verweise*
  - *Festlegungen der Lesereihenfolge (article threads)*
  - *digitale Unterschriften: analog zu Formularen*
  - *Non-Interactive Forms*
  - *XML Form Architecture (XFA)*
  - *Sicherheit*
  - *Navigation*
  - *Annotationen*
  - *Aktionen*
  - *XObjects*
  - *Fonts*

Checkpoint	Kurzbeschreibung
Index	Numerisches Kennzeichen
Fehler	Voraussetzungen für einen Fehler
Section	Bezug zum Abschnitt im PDF/UA Standard
Typ	Elemente von PDF: <i>Doc, Page, Object, JS</i> (eingebettetes Script), <i>Font</i>
Wie erfolgt der Test	Mensch oder Maschine

## PDF/UA vs WCAG

- PDF/UA definiert technische Merkmale und bietet einen technischen Rahmen
- PDF/UA enthält keine Anforderungen an die Auszeichnung des Inhalts (z.B. Akronyme)
- PDF/UA legt kein Mindestkontrastverhältnis fest
- PDF/UA beschreibt keine zeitlichen Anforderungen
- PDF/UA erweitert WCAG teilweise (Encoding, aktive Inhalte)
- PDF/UA nicht für Endanwender gedacht (teil von „Speichern unter“)
- PDF/UA betrifft auch das Leseprogramm
- PDF/UA kennt (außer bei Bildern) keine alternative Darstellungen (z.B. Redetext)

## Checkliste Schnellprüfung

- PDF mit Tags versehen
- Dokumententitel vorhanden
- Sprache festgelegt
- Semantik korrekt
  - Überschriften
  - Listen
  - Tabellen (Kopf vs. Zellen)
  - Bilder
- Lesereihenfolge sinnvoll
- Bilder enthalten keinen Text
- Bilder beschriftet
- Formularfelder beschriftet
- Schriftart korrekt verwendet (Encoding)
- Schmuckelemente ausgeblendet
- ü Sicherheitseinstellungen lassen Zugriff mittel AT zu

# PDF und Barrierefreiheit

## Schrittweise Verbesserung

## Tags in Pdf

## Aufbau Tags

- Version 4: Extrahierung des Textes (außer bei Scannerergebnissen)
- Version 5: plugin unterstützt Screenreader
- Adobe Version 6: tagging wurde eingeführt:
  - Automatisches und manuelles Erstellen
  - Automatisches Prüfen
- PDF/A (ISO 19005-1) berücksichtigt
- Def.: Tags sind eine hierarchische Struktur des Dokuments. Das erste Element dieser Struktur ist der Tag-Stamm. Alle anderen Elemente sind Tags; sie sind dem Tag-Stamm untergeordnet. Tags nutzen kodierte Elementtypen, die in spitzen Klammern (< >) angezeigt werden.

- Seitenstruktur muss linearisiert werden
- Tags werden hierarchisch durch AP angelegt,
- nur wenige Screenreader unterstützen PDF
  - *WindowEyes ab 4.5*
  - *Jaws ab 4.51 (auch verschlüsselt)*
  - *Blindows*
- Noch unvollständige Implementierung auch in Jaws 7
- ++ Überschriften
- ++ Listen
- -- Zitate (<blockquote>)
- ++ Sprache
- -- Sprachwechsel (evt. anlegen eines Span und Text tags)
- - Links
- + Bilder
- ++ Tabellen
- ++ Lesezeichen

### Liste der vordefinierten Tags

- Article <Art>, Annotation <Annot>, Bibliography Entry <BibEntry>, Block Quote <BlockQuote>, Caption <Caption>, Code <Code>, Division <Div>, Document <Document>, Figure <Figure>, Form <Form>, Formula <Formula>,
- Heading <H>, <H1>, <H2>, <H3>, <H4>, <H5>, <H6>
- Index <Index>, Label <Lbl>, Link <Link>, List <L>, List Item <LI>, List Item Body <Lbody>, Note <Note>, Paragraph <P>, Part <Part>, Quote <Quote>, Reference <Reference>, Section <Sect>, Span <Span>,
- Table <Table>, Table Data Cell <TD>, Table Header Cell <TH>, Table Row <TR>
- Table of Contents <TOC>, Table of Contents Item <TOCI>

### Spezielle Textelemente

- Block quote element <BlockQuote>: Zitat
- Caption element <Caption>: Beschreibung einer Tabelle oder eines Bildes (<lbl> ist für den Namen)
- Index element <Index>: Liste von Text in <Reference> Tags die zum Auftreten von Text verweisen
- Table of contents element <TOC>: Liste von strukturierten items und labels (kann verschachtelt sein)
- Table of contents item element <TOCI>: ein item

### Überschriften und Paragraphen

- <p>, <H> für verschachtelte <sections>
- <h1>-<h6> für lineare <sect>

### Label und Listenelemente

- <L> enthält <LI>. <LI> enthält <LBI> (optional) und <LBody>

### Block-level Elemente: Container

- Division element <Div>
  - Ein generischer Block oder eine Gruppe von Block-level Elemente
- Section element <Sect>
  - Ein generischer Block, vergleichbar zu Division (DIV Class="Sect") in HTML, üblicherweise ein Teil von <part> oder <Art>
- Article element <Art>
  - Ein abgeschlossener Textabschnitt
- Part element <Part>
  - Für größere Abschnitte, fasst andere tags zusammen wie article elements, division elements, oder section elements
- Document element <Document>
  - Die Wurzel eines Dokuments

### Tabellen

- <Table>, <TR>, <TD> und <TH> entsprechen HTML

### Inline Tags

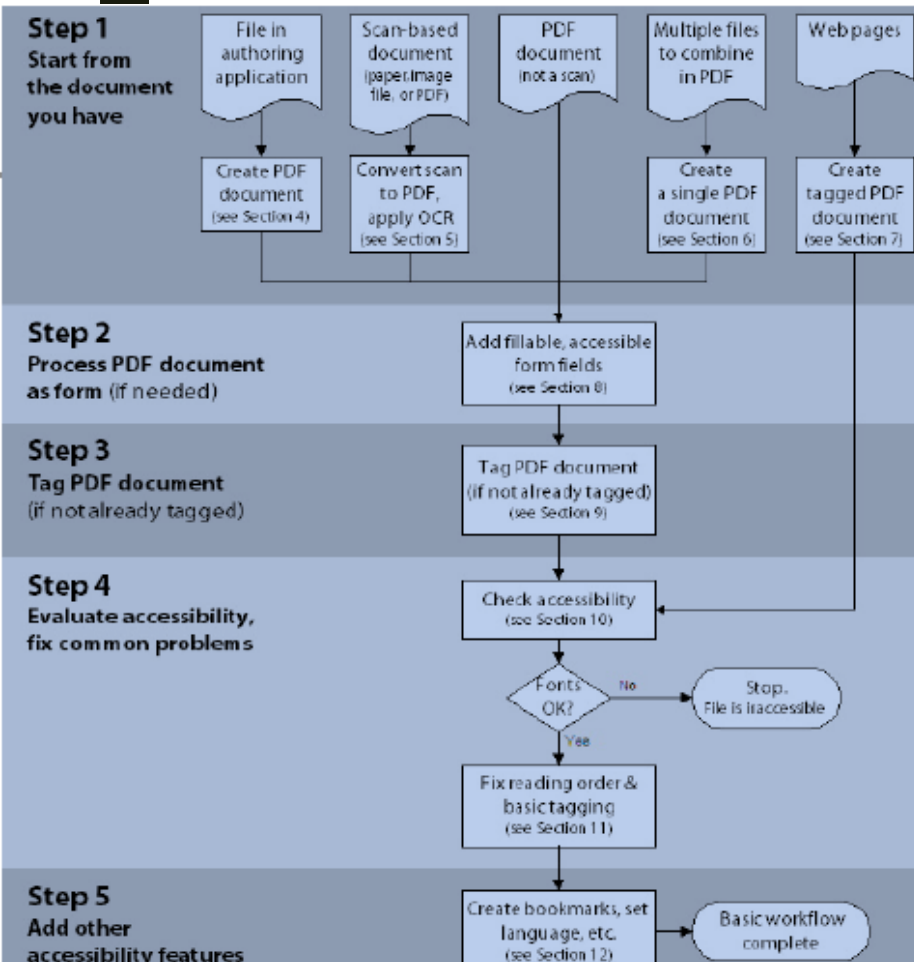
- <BibEntry> für Quellenangaben, kann label enthalten
- <Quote> für Zitate innerhalb von Text
- <Span> begrenzt stilistische Eigenschaften von Text
- <Code> für Quellcode

### Spezielle Tags

- <Figure> zeichnet Bilder im Text aus
- <Form> für Formularelemente
- <Link> für Veweise
- <Note> für Annotationen
- <Reference> für Daten innerhalb des Dokuments



## BF herstellen



## Ausgabehilfeprüfung und Bericht

- Vorhandensein alternativer Beschreibungen für Bilder
- Festlegung einer (!) Sprache eines Texts
- Zeichenkodierung bekannt
- Beschriftung der Formularelemente
- Listen- und Tabellenstruktur entspricht Ordnungsstruktur

Heruntergeladen von Studydrive

## PAC 2.0

Freies Werkzeug für tagged PDF, prüft u.a.

- Titel des Docs verfügbar
- Doc Sprache definiert
- Sicherheitseinstellung
- Tab folgt Tag-Struktur
- konsistente Heading Struktur
- Bookmarks verfügbar
- Font Encoding zugänglich
- Tags vollständig
- Logische Lesereihenfolge
- Alternativer Text
- korrekte Syntax der Tags/Roles
- ausreichender Kontrast
- Leerzeichen vorhanden

- 1) Technische Prüfung („Start“): alle maschinenprüfbaren Checks durchlaufen eine Kurzbericht („Results“) erstellen.
- 2) Detailbericht („Report“): Mit Hilfe des Detailberichts die einzelnen Fehler im Dokument analysieren.
- 3) Vorschau-Ansicht („Screenreader Preview“): vereinfachte Strukturansicht für die Qualität („Tags“) und der logischen Reihenfolge („Lese-Reihenfolge“).
- 4) Dokumentstatistik („Document Statistics“): Übersicht der Anzahl der verwendeten Struktur-Elemente.
- 5) Logische Struktur („Logical Structure“): Expertenansicht des kompletten Tag-Baums, um sich korrespondierende Elemente zu einem Tag im Dokument anzeigen lassen oder die Rollenzuweisungen zu kontrollieren.

## Verweise

- Verweise werden vor den allgemeinen Tags erstellt
- Automatische Erkennung aus OCR Ergebnis möglich (Menü: Erweitert/Verknüpfungen/alle URL erstellen)
- Verweise müssen explizit beschriftet werden (Optionen/Tag aus Auswahl erstellen/Typ „Verweis“)
- Verweise aus MS Office/OpenOffice bleiben erhalten

## Word: Exportieren als PDF

- Voraussetzung: vollständig mit Stilvorlagen arbeiten
- Eigene Stilvorlagen durch ableiten von vorhandenen Stilvorlagen bilden
- Lesezeichen erhalten durch Festlegen beim Transcodieren

## MS PowerPoint: Exportieren als PDF

- Bilder müssen mit Alternativ beschrieben werden (Grafik formatieren/Web)
- Masterfolienlayout für Schmuckgrafiken nutzen
- Texte werden nicht als Überschriften strukturiert, evt. manuelle Überarbeitung Notwendig
- Im PDF werdentrotzdem einige Fehler möglich:
  - *Inhalt ohne Zuordnung (OLE)*
  - *Tabellen falsch erkannt bei zu enger Spaltenwahl*
  - *Tabreihenfolge fehlerhaft (?)*
  - *unzulässige Zeichen (Bullets) und*
  - *Alternativtext der Inhalte verdeckt (!)*

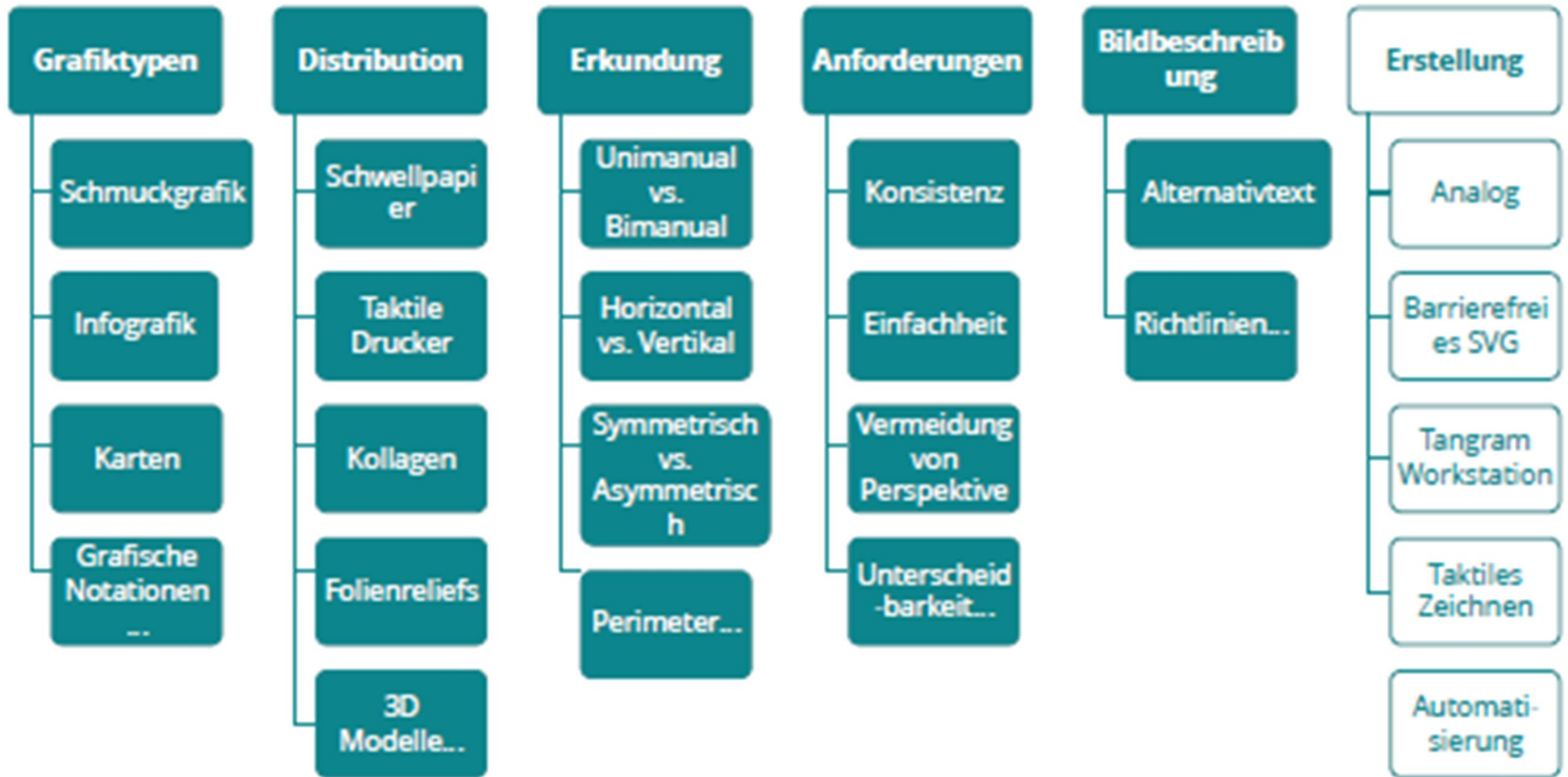
# GRAFIKEN

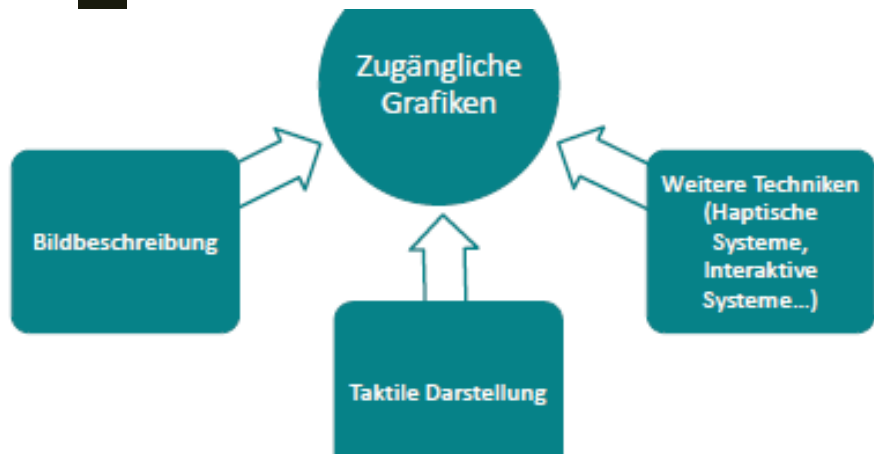
taktile Graphiken:  
was gibt es da so?  
(taktil plus Audio:  
wie realisieren?)



berührungsempfin  
dliche Displays =  
hörbare  
Rückmeldung  
möglich)







## Bildbeschreibung

- Prinzipiell: Alle Nicht-Text-Inhalte
- Ausnahme: Reine Schmuckgrafiken
- Sonst: Mindestens Alternativtext unterstützen

### WAS beschreiben?

- Grafiktyp
- Absicht/ Zweck des Bildes
- Ort, Objekte, Gebäude, Menschen
- Farben (wenn relevant)
- Atmosphäre
- Handlungen
- Kontext (keine redundanten Informationen geben, Fundort, Autor:innen...)

### WIE beschreiben?

- vom Allgemeinen zum Speziellen
- zielgruppenangepasst (Vokabular, Expertise...)
- objektiv (keine Interpretationen, Meinungen, Auslassungen oder Emotionen)
- kurz, prägnant und verständlich → inhaltstragende Wörter, Aufzählungen/ Listen
- Ton und Sprache (Terminologie, beschreibend, aktive Verben)

Heruntergeladen von Studydrive

## Detailgrad

### Drill-Down Organisation:

**Alternativtext:** Kurzer Überblick max. 1-2 Sätze

→ Sollte immer vorhanden sein (außer bei Schmuckgrafiken)

**Bildunterschrift:** Kurze Beschreibung mit zusätzlichen Informationen, die nicht auf visuelle Elemente fokussiert sein muss (für alle Menschen sichtbar)

**Bildbeschreibung:** Detaillierte Beschreibung der Bildinhalte, was den Zugang zu visuellen Konzepten unterstützt

Grafiken = Barriere v.a. für Menschen mit Blindheit und Sehbeeinträchtigung

Zugang zu Grafiken unausweichlich für gleichberechtigte, gesellschaftliche Teilhabe, z.B. für Bildung, soziale Bereiche, SocialMedia, Kultur & Kunst...

gleichwertigen Zugang durch alternative Darstellungsweise des Inhalts gewähren

# Bereitstellung HTML

## HTML

### ALT-Attribut

Pflichtattribut für Grafiken

Zweck: Kurze Inhaltsbeschreibung, Verweis auf Kontext bzw. Langbeschreibung

Leeres Alt-Attribut wird von Screenreadern ignoriert (z.B. für Schmuckgrafiken)

Title-Attribut ist keine Alternative!

Hinweis zum Ort der detaillierten Beschreibung geben

### Longdesc-Attribut

Link zu externen (ausführlichen) Beschreibungen auf der gleichen oder einer anderen Seite

für alle HTML-Elemente möglich

**Nachteile:** nicht mit allen Screenreadern kompatibel

nur für Screenreader zugänglich (nicht visuell ersichtlich)

### Link zur Beschreibung

Sprungmarke zur langen Beschreibung direkt neben dem Bild

#### Vorteile:

sichtbar für alle Nutzende

kompatibel mit allen Browsern und assistiven Technologien

#### Nachteile:

keine semantische Verbindung zwischen Bild und Text

### ARIA „aria-describedby“ = semantische Auszeichnungssprache für HTML

Referenzieren von (langen) Beschreibungen auf der gleichen Seite

versteckte Beschreibungen möglich (Offscreendiv)

**Vorteile:** keine Nutzerinteraktion nötig (wird nach alt-Attribut vorgelesen)

gute Screenreaderkompatibilität für alle sichtbar

**Nachteile:** Beschreibung ist reiner String (keine semantische Gliederung) muss auf gleicher Seite sein

### Sonderfall: Bild im Linktext

wird häufig u.a. für Logos („Home-Button“) verwendet

Bild im Linktext eingebettet

Alt-Text sollte die Funktion und Ziel des Links enthalten

## SVG

- Erhöht Verständnis von Grafiken für alle Betrachtenden
- Lesbarkeit ohne Grafikprogramm möglich
- alle Elemente semantisch kennzeichnen:
  - wenn möglich Basistypen statt `path` verwenden (z.B. `circle`, `rect`, `line`, `polygon` etc.)
  - Textalternativen und -beschreibungen: `title`, `desc`, `meta`
  - Gruppierungen von Elementen mit `g`
  - sinnvolle Wiederverwendung gleichbedeutender, separat definierter Elemente mit `use`
  - Objekttransformationen vermeiden (Linienstile, Schriftgrößen etc. werden mitskaliert)

## Editoren

### **Bildbeschreibung hinzufügen**

unterstützt von diversen Programmen (z.B. Word, Powerpoint, Acrobat Reader DC)

Beschreibungen  
gegebenenfalls in separater Datei mitliefern (mit entsprechendem Verweis darauf)

## Herausforderungen

Oft nur eine *mögliche* Interpretation → subjektiv, abhängig von Wissen und Fähigkeiten des Erstellenden  
Notationscharakteristik  
schwer verbalisierbar

Detaillierungsgrad (Farben? Hintergrundwissen?)  
Erstellung sehr aufwändig, bspw. AGSBS → meist manuelle Erstellung von Personen mit Fachexpertise  
eingeschränkte selbstständige Erfassung und Deutung des Inhalts von Personen mit Blindheit  
Verstehen komplexer Beschreibungen ist anstrengend und benötigt viel Zeit

# Taktile Grafiken

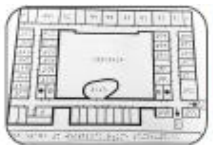
## Definition

fühlbare Grafiken, die mit dem Tastsinn wahrgenommen werden können

bestehen aus erhabenen Punktsymbolen, Linien und Texturen  
→ Unterscheidung (Farbersatz)

häufig in Kombination mit Braille-Beschriftungen

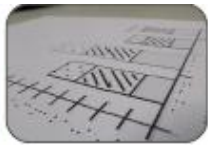
verschiedene Erstellungsverfahren und Techniken verfügbar



Schwellpapier



Kollagen



Prägedruck



Folienreliefs



Digitale  
Stiftplatte

Verfahren	Erstellung (Institutionen) *	Verwendung (Nutzer:innen) **
Schwellpapier	80 %	90 %
Brailledrucker	55 %	72 %
Kollage	20 %	62 %
Tiefziehfolie	55 %	87 %
3D-Model	20 %	63 %

## Distribution

- Schwellpapier (Helligkeit = Höhe, spez. Papier erhitzt + jeder Drucker – schlecht Braille – braucht Fuser – Kosten)
- Braille Drucker (optimiert für Braille, änderbarer Punktabstand + tiefe Prägung + var Auflösung, + aus Text generierbar, - geringe Auflösung – nur eine Reliefhöhe)
- Taktile Drucker: gut für Grafikdruck (+ scharfe Kanten und Linien, + großes Papier, + Kombi Schwarzschrift mgl; - Auflösung, - Kosten Hardware, - unüblich)
- Kollagen: Komposition verschiedene Materialien (+ Details, + realitätsnah, + Strukturen, - hoher Aufwand, - Vervielfältigung)
- Punktreiefs punzieren = Metallprägung, mit Platte auf Papier (+ Vervielfältigung, - keine Korrektur mgl, Zinkverbrauch)
- 3D Modelle (+ realitätsnah haptisch, beliebige Höhen und Details, Braille mgl, Farben mgl, unterschiedl Materialien; - zeitintensive Erstellung, lange Druckzeit, Expertise nötig)



# Erkundung

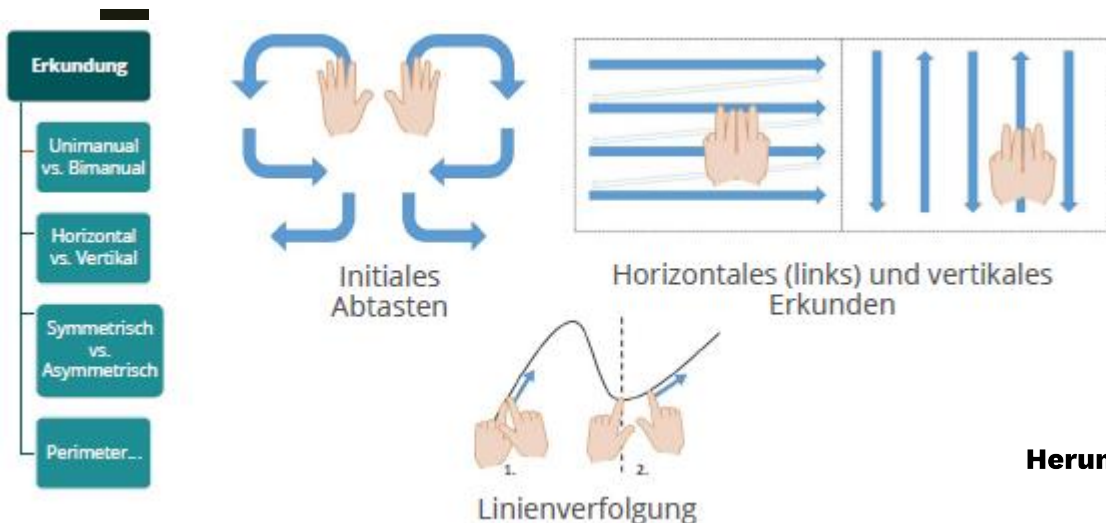
Erkundung taktiler Grafiken muss erlernt werden

Kognitiver Prozess: Zusammensetzen eines Bildes aus vielen Einzelbildern  
unterschiedliche Strategien zum Bildverständnis

wird meist in Schule unterrichtet

Minimalverständnis des Grafiktyps wichtig (v.a. bei komplexen Grafiken)

meist Braillekenntnisse erforderlich



## RICHTLINIEN

Wahrung der ursprünglichen Aussage

Reduzierung der Komplexität

Texturen, Linienstile und Punktsymbole sparsam verwenden

Perspektive vermeiden

Aufteilen komplexer Objekte

Unterscheidbarkeit

Verwendung von Braille-Schrift

## MINDESTMAßE

- Linienlänge
- Versenkungen
- Kreuzende Linien

## TEXTUREN

- Verwechslungen ausschließen
- Wiedererkennung ermöglichen (konsistente Nutzung)
- Ausgabemedium einbeziehen
- Flächengröße beachten
- Verwendung evaluierter Texturen-Sets empfohlen

# Erstellung



## Digital

häufig aufwendiger,  
manueller Prozess

Richtlinien für taktile  
Grafiken müssen eingehalten  
werden

häufig als Transkription von  
visuellen Grafiken

Optimalfall: Autor:nerstellt  
taktile und visuelle Grafik

## Teilautomatisiert

**Häufigster Ansatz:** Anpassung  
einer bestehenden visuellen  
Grafik für die taktile Ausgabe  
(Transkription)

*Beispiel: TGA*(TactileGraphics  
Assistant)

Algorithmus zur  
automatischen Vereinfachung  
und Optimierung  
herkömmlicher Grafiken in  
taktile Grafiken:manuelles  
Eingruppieren der Bilder in  
Klassen sowie Training

Separieren und Entfernen von  
Text innerhalb einer Grafik

OGRund-Umwandlung in Braille

## Vollautomatisch

geeignet für wohldefinierte  
Grafiktypen (z.B. Diagramme)

wenige Anwendungen mit gutem  
Ergebnis vorhanden

Qualitätskontrolle sollte dennoch  
sichergestellt sein

Ermöglicht selbstständige  
Erstellung durch Menschen mit  
Blindheit oder  
Sehbeeinträchtigung

Kontrolle des Ergebnisses ohne  
Ausdruck schwierig

Beispiele beim Thema  
„Diagramme“

# Taktile Interaktion

## Audio-haptische Systeme

### Nachteile taktiler Grafiken

- begrenzte Auflösung  
→ geringe Informationsdichte
- Unterscheidbarkeit der Elemente (max. 5 Texturen/ Symbole/ Linienstile)
- Muss erlernt werden  
→ hoher kognitiver Aufwand

### Ansatz

- Multimodale Systeme  
→ Kombination verschiedener Ein- und Ausgabemöglichkeiten, z.B. haptischer und auditiver Elemente
- Ansprechen verschiedener Sinne

Technologien  
Zahlreiche Ansätze, um Interaktion zu ermöglichen, z.B. Videobasiertes Tracken des Fingers bei der Exploration der Grafik

Verwendung digitaler Stifte, die Position erkennen

Einbetten von RFID Tags

3D-Druck mit leitfähigen Filamenten

## HyperBraille-Fenstersystem

Konzept zur taktilen Darstellung und Interaktion mit Anwendungen (Fenstersystemen)  
– Evaluation mit der Zielgruppe innerhalb von empirischen Studien

– Unterstützung thematischer Ansichten für verschiedene Anwendungsfälle

– äquidistantes Braille erfordert Änderungen der Lesegewohnheiten

### Nutzende

- Fähigkeiten
- Vorkenntnisse

### Anwendungskontext

- Mobilität
- Limitierungen der Umwelt

### Interaktion

- Konzept (Führung, Exploration, Überblick)
- unterstützte Modalität (Touch, Sprache, Tastatur...)

### Limitierungen der Grafik

- Reproduzierbarkeit
- Auflösung und Größe der Anzeigefläche
- Darstellbarkeit
- Mobilität

Heruntergeladen von  Studydrive

Kopf

Struktur

Detail

Ansichtenliste

Fensterleiste

Darstellung

Symbol

Überblick

Original

Arbeit



# Zeichensysteme

Zeichnen und visualisieren von Gedanken wichtig für Kommunikation, persönliche Ausdruckskraft, künstlerische Darstellungen etc.

Zeichnen ist schwierig, erfordert handwerkliches Können

Problem: kein Feedback des Gezeichneten

## **Ansatz:**

Entwicklung von Werkzeugen zur Unterstützung des Zeichenprozesses

## Analog

- Steckbrett
- Formkasten
- Kopierrädchen
- Quickdrawpaper
- Wollfaden auf Klett
- 3D Painting
- Zeichenfolie

## Digital

### Zeichnen durch Programmieren

Braille-Buchstaben werden zu Bildpunkten  
BPLOT

- Erzeugung von Ausdrucken für Brailledruckermittels *plottercontrollanguage* → keine Überprüfung während des Zeichnens möglich
- Abpausen von taktilen Objekten über Touchpad

IC2D(Integrated Communication 2 Draw)

- Navigation und Malen auf dem Bildschirm mit Hilfe von Sprachausgabe und Musik
- Punktauswahl durch rekursives Schema basierend auf 3 x 3 Gitter (Telefontasten) → Bedienung durch Tasten 1-9 bzw. Pfeilnavigation

**Analog**

- schnell und einfach
- günstig
- detaillierte und naturgetreue Darstellung möglich

**Digital**

- gute Fehlerkorrektur
- hohe Veränderbarkeit
- hohe Reproduzierbarkeit
- leichte Distribution

**Analog**

- schwierige/ keine Fehlerkorrektur
- schwer reproduzierbar
- wenig Unterstützung beim Zeichnen

**Digital**

- Erfordert oft hohe kognitive Ressourcen
- oft begrenztes Anwendungsgebiet
- Spezialequipment/-hardware notwendig (Kosten)

Kontra

## Zeichensysteme | Anforderungen

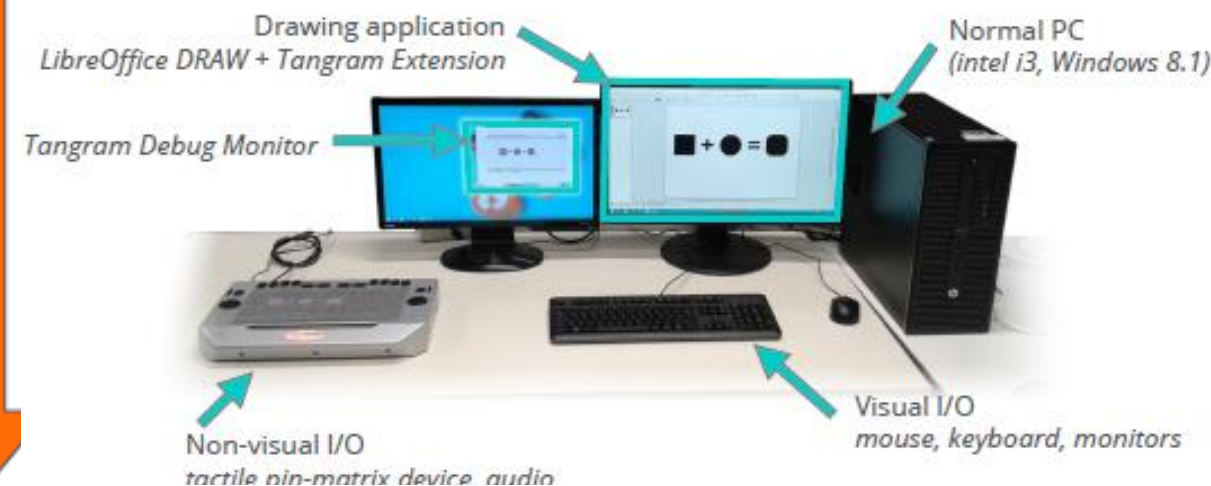
Bewertung der Anforderungen von Menschen mit Blindheit (Top 8):

1. Taktils Feedback
2. Elemente löschen können
3. Export visuelles Bild
4. Editieren von Elementen – Bewegen, Größe ändern, Transformieren, Anpassen grafischer Eigenschaften
5. Methoden zur Fehlerbehebung (undo/redo)
6. Export taktils Bild
7. Selbständiges Erstellen
8. Kompatibel mit anderer assistiver Technologie (z.B. Screenreader)

## Zeichensysteme | Kollaboration

**TangramWorkstation**

Einbezug der Zielgruppe in Grafikerstellung → Kollaborative Grafikerstellung mit blinden und sehenden Lektor:innen



## Zeichensysteme | Haptische Systeme (Phantom)

**AHEAD (Audio-Haptic drawing Editor and Explorer for Education)**

- Force-Feedback-System → Feedback durch Kraftrückkopplung
- Führung durch Mausbewegung (Abzugskraft bringt Phantom-Stift zum Mauscursor)
- Erkundungsmodus (haptisches Bild als positives oder negatives Relief, Berührung zur Auswahl und Audioausgabe)
- Sättiermodus zum Zeichnen und zur Manipulation vorhandener Objekte



# Diagramme

Kriterien:

Titel, Achsen,  
Legende, Raster,  
Beschriftungen

## Beschreibung

- 1. Überblick:** Diagrammtyp (Bezeichnung), Besonderheiten (z.B. horizontal vs. vertikal), Titel
- 2. Achsen:** Anordnung/Lage, Beschriftung, Einheit, Skala (Wertebereich, Intervalle)
- 3. Daten:** je nach Diagrammtyp, z.B. Anzahl der Datenreihen, Name und Anordnung Daten, konkrete Datenwerte (wenn möglich tabellarisch)



## Sonifikation

- Synthetisieren der Datenwerte in Töne
- ein Tongraph je Datenwertgruppe
- keine direkte Beeinflussung der Sonifikation möglich
- bei Liniendiagrammen Sonifikation der vertikalen Werte
- Abspielen des Tongraphen per Tastendruck

**Ziel:** Menschen mit Blindheit oder Sehbeeinträchtigung die Erstellung zugänglicher, taktiler Diagramme ermöglichen

## Anforderungen

- Software zur automatischen Erzeugung zugänglicher Diagramme, z.B. in SVG
- versch. Eingabeformate denkbar: z.B. Datentabellen, grafisches Diagramm, Mark Up
- barrierefreie Bedienung
- geeignete Formate für zugängliche Grafiken: SVG, HTML
- verschiedene Diagrammtypen und Designs bereitstellen
- zugängliche Bildschirmdarstellung für sehbehinderte Menschen integrieren
- Erzeugung auf Grundlage von Gestaltungsrichtlinien (css)

## Anforderungen und Ziele der automatischen Erstellung

blinden und sehbehinderten Menschen selbständigen Zugang zu Diagrammen ermöglichen zur Kontrolle eines selbst erstellten Diagramms

um Zugang zu einem visuellen Diagramm zu bekommen

Automatisierte Erstellung von Diagrammbeschreibungen für verschiedene Anwendungsfälle

Integration in bestehenden Workflow

Heruntergeladen von Studydrive

## Auditive Zugängliche Diagramme in SVG

- **Idee:** Erstellung interaktiver Diagramme in SVG zur nicht-visuellen Exploration
- Erreichbarkeit aller Informationen ermöglichen
- Zugänglichkeit der Informationen für den Screenreader
- Navigierbarkeit per Tastatur
- Gruppierung von Inhalten zur effizienten Navigation
- Sonifikation der Daten, um generellen Trend darzustellen

## Taktile Diagramme

- Verwendung vor allem im Bildungskontext üblich
- vorrangig klassische Diagramme wie Balken-, Linien-, Punkt- oder Kreisdiagramme taktil umgesetzt
- meist mit Schwellpapier, Brailledruck oder als Prägedruck erzeugt
- keine speziellen Richtlinien zur Gestaltung taktiler Diagramme, ABER
- die Gestaltung der Diagramme muss an den taktilen Sinn angepasst sein
- → keine reine „Übersetzung“ von grafischen in taktile Diagramme sinnvoll

# Barrierefreie Karten

## Anforderungen

### Datenbasis

- Allgemeine Informationen (z. B. Adresse, Routen, Objekte, POIs)
- Erreichbarkeitsmerkmale (z. B. taktile Beläge, Rampen)
- Aktualität der Daten

### Anwendung

- (Grafische) Benutzungsoberfläche
- Interaktionsmodalitäten
- Adaptivität und Benutzungsfreundlichkeit
- Funktionale Anforderungen (z. B. Filter, Zoomen, Editieren, Navigation)

### Karte

- Typ (z. B. innen, außen, öffentliche Verkehrsmittel)
- Barrierefreie Darstellung (z. B. taktile Karten, audio-taktile Karten)
- Mobilitätsgrad
- Interaktion

starke Vereinfachung notwendig

Erfahrungen fast ausschließlich für Outdoor (Straßen, Kreuzungen, Wege, Gebäude etc.)

selten bis keine Verfügbarkeit von Gebäudekarten (Räume, Stockwerke, Hindernisse, Einrichtung, Türen etc.)

## Probleme

geringe Auflösung

kleiner Bildausschnitt

viele Kartenelemente

Position und Ausrichtung des Nutzenden

unterschiedliche Sichtweisen auf die Karten

# Taktile Karten ZSMFassung

## Taktile Karten

- **Taktile Stadtpläne (Outdoor)**
- Abbildung von Straßen, Gebäuden, Kreuzungen
- Beispiel: YAH Maps
- Mögliche Datengrundlage: OpenStreetMap Data
- **Taktile Gebäudepläne (Indoor)**
- Abbildung von Räumen, Wänden, Treppen, Fahrstühlen usw.
- verschiedene Datenformate, z.B. OSM, IndoorGML
- Mögliche Datengrundlage: Gebäudepläne, Bilderkennungsverfahren usw.
- Taktile Karten können die Sicherheit und Mobilität von blinden und sehbehinderten Menschen erhöhen bzw. verbessern
- Müssen an die Bedürfnisse der Zielgruppe angepasst werden (insbesondere in Bezug auf Kontext der Nutzung (zu Hause vs. vor Ort), Informationsbedarf und Ausgabeformat

## Zugängliche Grafiken

- Mindestanforderung: Bildbeschreibung
- besser als die nachträgliche Anpassung, ist die barrierefreie Erzeugung von Grafiken
- Methodestark abhängig vom Grafiktyp und Anwendungsfall
- Barrierefreie Erstellungsprozesse von Grafiken ermöglichen (z.B. durch Automatisierung)
- Einbezug der Nutzenden unabdinglich
- verschiedene Medien, um Grafiken zugänglich zu gestalten, z.B.
  - Bildbeschreibungen
  - barrierefreies SVG
  - taktile Grafiken
  - audio-taktile Grafiken
  - Sonifikation
  - Force-Feedback...