

# Bachelor-Thesis

## Integration von Sprachsteuerung für Redakteurstätigkeiten in Content- Management-Systemen unter besonderer Berücksichtigung von TYPO3

Vorgelegt von: Michael Helthuis

Fachbereich: Elektrotechnik und Informatik

Studiengang: Medieninformatik Bachelor

Erstprüfer: Sebastian Kreideweiß, M.Sc

Ausgabedatum: 27. August 2024

Abgabedatum: 28. November 2024

**Aufgabenstellung:**

Diese Arbeit fokussiert sich auf die Machbarkeit einer Implementierung von Sprachsteuerung in ein Content-Management-System. Hierfür soll eine Erweiterung eines TYPO3 Systems prototypisch erstellt werden, welche darstellen soll, ob Redakteurstätigkeiten durch Sprachsteuerung ausgeführt werden können.

# Eigenständigkeitserklärung

Declaration of Originality

---

**Name, Vorname**

*Last name, first name*

---

**Matrikelnummer**

*Matriculation number*

**Ich versichere hiermit, dass ich die vorliegende**

I hereby declare that this

**Hausarbeit**  
term paper

**Bachelorarbeit**  
bachelor's thesis

**Masterarbeit**  
master's thesis

**mit dem Titel**

with the title

---

---

---

**eigenständig und ohne unerlaubte fremde Hilfe angefertigt habe. Ich habe keine anderen als die angegebenen Quellen und Hilfsmittel verwendet und Entlehnungen aus anderen Arbeiten kenntlich gemacht. Für den Fall, dass die Arbeit zusätzlich elektronisch und/ oder digital eingereicht wird, erkläre ich, dass die schriftliche und die elektronische und/ oder digitale Form identisch sind. Die Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen.**

is my own original work and any assistance from third parties has been acknowledged. I have clearly indicated and acknowledged all sources and resources as well as any borrowings from other works. In case of an additional electronic and/or digital submission of this work, I declare that the written form and the electronic and/or digital form are identical. This work has not previously been submitted either in the same or in a similar form to another examination office.

**Ich bin damit einverstanden, dass die vorliegende Hausarbeit/ Bachelorarbeit/ Masterarbeit für Veröffentlichungen, Ausstellungen und Wettbewerbe des Fachbereiches verwendet und Dritten zur Einsichtnahme vorgelegt werden kann.**

I agree that this work can be used for publishing, exhibition or competition purposes and can be inspected by third parties.

**ja**  
Yes

**nein**  
no

**es liegt ein Sperrvermerk bis \_\_\_\_\_ vor**  
there is an embargo period until

---

**Ort, Datum**

Place, Date

---

**Unterschrift**

Signature

Zusammenfassung der Arbeit

Abstract of Thesis

Fachbereich: **Elektrotechnik und Informatik**

Department: :

Studiengang: **Studiengang Medieninformatik**

University course:

Thema: **Integration von Sprachsteuerung für Redakteurstätigkeiten in**

Subject: **Content-Management-Systemen unter besonderer Berücksichtigung von TYPO3**

Zusammenfassung:

Die vorliegende Arbeit ist eine Analyse der Möglichkeiten einer Integration von Sprachsteuerung im Content-Management-System TYPO3. Eingegrenzt auf die Tätigkeiten von Redakteur:innen und die Navigation innerhalb des Backends.

Abstract:

This thesis is an analysis of the possibilities for integrating voice control into the TYPO3 content management system, focusing on the activities of editors and navigation within the backend.

Verfasser:

Author: **Michael Helthuis**

Betreuender Professor/in:

Attending Professor: **Sebastian Kreideweiß, M.Sc.**

WS / SS:

**WS 2024/2025**

# Inhaltsverzeichnis

<b>Zusammenfassung der Arbeit .....</b>	<b>1</b>
<b>Inhaltsverzeichnis.....</b>	<b>2</b>
<b>1 Einleitung .....</b>	<b>1</b>
1.1 Relevanz und Motivation .....	1
1.2 Ziel der Arbeit .....	1
<b>2 Grundlagen und Definitionen .....</b>	<b>2</b>
2.1 Content-Management-Systeme (CMS) .....	2
2.1.1 TYPO3 .....	2
2.1.2 Backend .....	2
2.2 TYPO3 Erweiterungen .....	3
2.3 Sprachassistenten und Sprachsteuerung .....	3
2.4 API (Application Programming Interface) .....	3
2.5 Barrierefreiheit (Accessibility).....	4
2.6 Nutzbarkeit .....	4
2.7 Modal .....	5
2.8 Redakteurstätigkeiten .....	5
2.8.1 Aufgabenbereiche .....	5
2.8.2 Redakteurstätigkeiten in TYPO3 .....	5
<b>3 Grundlagen der Integration von Sprachsteuerung.....</b>	<b>6</b>
3.1 Relevanz von Sprachsteuerung in Content-Management-Systemen .....	6
3.2 Konzeptionelle Überlegungen zur Integration von Sprachsteuerung .....	6
<b>4 Aktueller Stand der Forschung.....</b>	<b>7</b>

4.1	Relevante Studien zu Sprachassistenten.....	7
4.1.1	Betrachtete Studien .....	7
4.2	Aktuelle Nutzung von Sprachassistenten .....	8
4.2.1	Sprachassistenz im Gesamtbild.....	8
4.2.2	Entwicklung der individuellen Studien .....	9
4.2.3	Teilnehmerprofile der Studien .....	9
4.2.4	Verwendung von Sprachassistenten nach Altersgruppen.....	10
4.2.5	Sprache als Ersatz zu herkömmlichen Mitteln .....	11
4.2.6	Bedenken zur Sprachsteuerung .....	11
4.3	Auswertung und Trends .....	13
<b>5</b>	<b>Methodik .....</b>	<b>15</b>
5.1	Anforderungsanalyse.....	15
5.2	Entwicklungsprozess .....	16
5.3	Implementierung Google Home .....	17
5.4	Implementierung Amazon Alexa .....	17
5.4.1	Erstellung von TYPO3-Endpunkten .....	18
5.4.2	Amazon Developer Console .....	18
5.4.3	Individuelle Sprachbefehle.....	18
5.4.4	Codeimplementierung von Sprachbefehlen .....	18
5.4.5	Authentifizierung Alexa und TYPO3 .....	18
5.4.6	Entwicklung der TYPO3 Erweiterung in PHP .....	19
5.5	Implementierung Web Speech API .....	20
5.5.1	JavaScript.....	21
5.5.2	PHP .....	25

5.6	Funktionstests.....	27
5.6.1	Definition der Äquivalenzklassen .....	28
5.6.2	Abgeleitete Testfälle .....	28
5.6.3	Testfall: Arbeitsfluss .....	29
5.7	Nutzbarkeitstests .....	30
5.7.1	Genauigkeit der Eingabe .....	30
5.7.2	Latenzzeit.....	31
5.7.3	Verfügbare Steuerungsmethoden .....	31
5.7.4	Sprachunterstützung.....	31
<b>6</b>	<b>Ergebnisse der Implementierungen und Tests .....</b>	<b>31</b>
6.1	Implementierung Google Home .....	31
6.2	Implementierung Alexa .....	32
6.3	Implementierung Web Speech API .....	34
6.4	Ergebnisse Funktionstests .....	35
6.4.1	Grundlegende Funktionen.....	35
6.4.2	Sprachsteuerung im Arbeitsfluss.....	36
6.5	Ergebnisse Nutzbarkeitstests .....	37
6.5.1	Überprüfung der Genauigkeit .....	37
6.5.2	Gemessene Latenzzeit.....	38
6.5.3	Funktionsweise der Steuerungsmethoden .....	38
6.5.4	Sprachkomplikationen .....	38
<b>7</b>	<b>Schlussfolgerung.....</b>	<b>39</b>
7.1	Ausblick .....	40
<b>8</b>	<b>Danksagung.....</b>	<b>41</b>

<b>9</b>	<b>Literaturverzeichnis.....</b>	<b>42</b>
<b>10</b>	<b>Abbildungsverzeichnis .....</b>	<b>45</b>

# 1 Einleitung

## 1.1 Relevanz und Motivation

Die Entwicklung der Sprachsteuerung von elektronischen Geräten hat besonders in den vergangenen Jahren große Fortschritte gezeigt. So erweitern sich die Nutzungsmöglichkeiten von virtuellen Sprachassistenten vom ursprünglichen Gebrauch, indem grundlegende Fragen beantwortet werden konnten. Nun ist es den Nutzenden ebenfalls möglich, Konversationen mit dem Assistenten zu führen, Termine im Kalender eintragen zu lassen und sogar durch künstliche Intelligenz Anrufe tätigen zu lassen. Es werden demnach weitreichende Möglichkeiten geboten, ein Gerät per Sprache zu steuern. Durch virtuelle Sprachassistenten wie Google Assistant, Amazon Alexa, Apples Siri oder seit neuestem auch die konversationelle künstliche Intelligenz von OpenAI, ist die Nutzung dessen bei einem Großteil der Bevölkerung stets in Reichweite. Sei es durch die Integrierung im Smartphone, durch so genannte Smart Speaker im Zuhause oder am Computer, der persönliche Sprachassistent ist meist lediglich ein Aktivierungswort oder ein Knopf entfernt.

Durch stetige Weiterentwicklung und tieferer Integrierung von Sprachassistenten in alltäglichen Geräten steigt auch deren Akzeptanz in der Bevölkerung. So geben diverse Umfragen mit steigender Prognose an, mehr als die Hälfte der Befragten nutzen mindestens einen Voice Assistant. Bevorzugt werden hierbei grundsätzlich Amazon Alexa, Google Assistant und Siri von Apple. Jedoch wird die Sprachsteuerung nicht nur weiterentwickelt, um den Alltag zu unterstützen, sondern bietet auch einen Grundbaustein der Barrierefreiheit. Als benutzerfreundliche Schnittstelle können Nutzer:innen mit Technologien interagieren, ohne notwendigen Einsatz traditioneller Eingabegeräte. Somit kann die Barrierefreiheit für Menschen mit körperlichen Einschränkungen, sei es motorisch oder visuell, gefördert werden.

Die Akzeptanz und das Maß der Nutzung von Sprachsteuerung steigen. Eine Implementierung von Sprache zur Navigation einer Website bietet ebenfalls eine Erweiterung der Barrierefreiheit. So könnte durch Implementierung dessen auch eine benutzerfreundliche, vereinfachte Schnittstelle für Content-Management-Systeme geboten werden.

## 1.2 Ziel der Arbeit

Das Ziel dieser Arbeit ist es, die Machbarkeit einer Integration von Sprachsteuerung in Content-Management-Systeme (CMS) zu analysieren und prototypisch zu entwickeln. Im Mittelpunkt steht die Frage, inwiefern eine Sprachsteuerung Redakteurstätigkeiten innerhalb des CMS TYPO3 unterstützen kann und welche Herausforderungen hierbei auftreten.

Dazu soll eine TYPO-Erweiterung entwickelt werden, die grundlegende Tätigkeiten als Sprachsteuerung implementieren soll. Der Schwerpunkt liegt dabei auf der praktischen Umsetzung und der Evaluation, ob Sprachbefehle erfolgreich in Redakteursprozesse integriert werden können. Demnach soll nicht zwingend eine marktreife Lösung geboten werden, sondern es steht vielmehr die Untersuchung der technischen und funktionalen Rahmenbedingungen im Vordergrund.

## **2 Grundlagen und Definitionen**

### **2.1 Content-Management-Systeme (CMS)**

[1] Ein Content-Management-System bietet Nutzenden als Software die Möglichkeit, Inhalte von diversen Webseiten wie Blogs oder Online-Shops zu erstellen, zu verwalten und zu veröffentlichen. Hierbei ist es den Nutzenden nicht notwendig, umfassende technische Kenntnisse zu besitzen. Zudem trennen CMS den Inhalt von der Verwaltung des Layouts und der Struktur, wodurch Inhalte unabhängig von der Code Grundlage bearbeitet werden können. CMS bieten durch geringe Einstiegshürden also eine strukturierte Möglichkeit, Webseiten zu erstellen und zu gestalten. Als Werkzeug zur Inhaltsverwaltung sind CMS insbesondere für große Websites und für Unternehmen unerlässlich.

#### **2.1.1 TYPO3**

TYPO3 [2] ist ein leistungsstarkes, kostenfreies und quelloffenes Content-Management-System, welches insbesondere durch dessen Flexibilität und Anpassbarkeit hervorsteicht. Das CMS bietet umfangreiche Funktionen zur Verwaltung von Webseiten und ebenfalls zur Verwaltung von mehrsprachigen Inhalten. [3] Der Aufbau von TYPO3 ist modular, demnach werden so genannte Extensions, auf deutsch Erweiterungen, zu dem System hinzugefügt, um zusätzliche Funktionen zu ermöglichen. [4] Auch wird ein extensives Rechteverwaltung geboten. Hierbei können die Nutzenden diverse Benutzergruppen zugewiesen werden, wie Redakteur:in oder Administrator:in. Außerdem können durch die native Unterstützung von Mehrsprachigkeit mehrere Websites innerhalb von einer TYPO3 Installation auf verschiedenen Sprachen verwaltet werden.

#### **2.1.2 Backend**

Das Backend [5] eines CMS wie TYPO3 bezeichnet die administrative Oberfläche, über die Inhalte sowohl erstellt, bearbeitet, als auch verwaltet werden können. Es dient als Plattform für redaktionelle und administrative Tätigkeiten, wie das Anlegen neuer Seiten, das Veröffentlichen von Inhalten oder das Konfigurieren von Benutzerrollen.

Zusätzlich bietet das Backend [5] auch Zugriff auf weitere Funktionen, die über spezifische Module zu verwalten sind. Beispiele hierfür sind das Seitenmodul, das die Organisation und Bearbeitung der Seitenstruktur ermöglicht, oder der Extension Manager, über den Erweiterungen installiert und verwaltet werden können. Das Backend stellt also die Schnittstelle dar, über die Administratoren und Administratorinnen, sowie Redakteure und Redakteurinnen das gesamte System verwalten können, ohne auf den Quellcode zugreifen zu müssen.

## **2.2 TYPO3 Erweiterungen**

Eine TYPO3 Erweiterung [3] kann über den Extension Manager im TYPO3-Backend installiert werden. Solch eine Extension wird in der Regel in der Programmiersprache PHP entwickelt und ermöglicht benutzerdefinierte Lösungen für konkrete, individuelle Anforderungen. Sowohl Frontend- als auch Backend-Extensions können entwickelt und implementiert werden. Frontend-Extensions beeinflussen hierbei die Darstellung oder die Funktionalität des Frontends einer Website, wobei Backend-Extensions das Backend von TYPO3 erweitern.

## **2.3 Sprachassistenten und Sprachsteuerung**

Sprachassistenten [6] verarbeiten natürliche Sprache und bieten Nutzenden die Möglichkeit, über gesprochene Befehle mit Geräten und Software zu interagieren. Die bekanntesten Sprachassistenten wie Amazon Alexa, Google Assistant oder Apple Siri sind oftmals direkt in Smartphones integriert oder durch Smart Speaker verfügbar. In der Regel nutzen Sprachassistenten so genannte Wake Words, woraufhin auf die Anweisung der Nutzenden angenommen und verarbeitet wird. Beispiele hierfür wären „Hey Google“ oder „Alexa“. [7] Zur Steuerung von Websites steht die Web Speech API zur Verfügung, welche vom W3C verwaltet und unter anderem von Google maßgeblich mitentwickelt wurde. [8] W3C, also das World Wide Web Consortium, ist eine internationale Gemeinschaft, die Webstandards entwickelt und pflegt. Diese API ermöglicht es, Sprachbefehle im Browser zu erkennen und zu verarbeiten.

## **2.4 API (Application Programming Interface)**

Eine API (Application Programming Interface) [9] ist eine Schnittstelle, die es ermöglicht, dass verschiedenste Software miteinander kommunizieren kann. Hierfür werden definierte Regeln und Protokolle geboten, die festlegen, wie Software miteinander interagieren kann, ohne, dass interne Strukturen offengelegt werden. So bietet die Web Speech API [10] eine Schnittstelle zur Spracherkennung, die es der Anwendung ermöglicht, Sprache in Text umzuwandeln und zu verarbeiten. Während die API clientseitig im Browser genutzt wird, erfolgt die eigentliche Spracherkennung in der Regel serverseitig durch Cloud-Dienste. Weitere Hardware oder Sprachassistenten wären bei dessen Verwendung nicht erforderlich. Aktuell

wird die Web Speech API nicht von Browsern wie Firefox unterstützt, da Mozilla Bedenken hinsichtlich Datenschutzes und der Abhängigkeit von externen Cloud-Diensten geäußert hat.

## 2.5 Barrierefreiheit (Accessibility)

[11] Eine herkömmliche Beschreibung der Barrierefreiheit besagt, dass Gebäude, Verkehrsmittel, öffentliche Einrichtungen oder Ähnliches barrierefrei, also für alle Menschen, einschließlich Menschen mit diversen Einschränkungen, zugänglich ist. Jedoch reicht die Definition des Begriffs noch wesentlich weiter. So wird der Begriff der Barrierefreiheit im Behindertengleichstellungsgesetz wie folgt definiert.

*„Barrierefrei sind bauliche und sonstige Anlagen, Verkehrsmittel, technische Gebrauchsgegenstände, Systeme der Informationsverarbeitung, akustische und visuelle Informationsquellen und Kommunikationseinrichtungen sowie andere gestaltete Lebensbereiche, wenn sie für Menschen mit Behinderungen in der allgemein üblichen Weise, ohne besondere Erschwernis und grundsätzlich ohne fremde Hilfe auffindbar, zugänglich und nutzbar sind. Hierbei ist die Nutzung behinderungsbedingt notwendiger Hilfsmittel zulässig.“ [11]*

Demnach können sowohl elektronische Geräte wie Smartphones oder Computer, als auch deren Software barrierefrei gestaltet werden. So sollte auch in der Gestaltung von Software wie Betriebssystemen oder auch Webseiten darauf geachtet werden, dass diese barrierefrei gestaltet sind. Inhalte müssen auch für Menschen mit visuellen oder akustischen Einschränkungen zugänglich sein. Weiterhin können Systeme so aufgebaut sein, dass diese auch ohne Maus oder Touchscreen bedient werden können, sei es durch die Verwendung von Tastaturen oder durch die Implementierung von Sprachsteuerung.

## 2.6 Nutzbarkeit

Der Begriff Nutzbarkeit, auch als Usability bezeichnet, wird unter anderem in der Bewertung und Gestaltung von Systemen verwendet. Die internationale Norm ISO 9241-11 definiert Usability wie folgt.

*„The objective of designing and evaluating systems, products and services for usability is to enable users to achieve goals effectively, efficiently and with satisfaction, taking account of the context of use.“ [12]*

Dies bedeutet, dass das Ziel der Nutzbarkeit darin besteht, ein System so zu gestalten, dass Nutzende ihre Ziele effektiv, effizient und zufriedenstellend erreichen können, wobei der Kontext berücksichtigt wird. Demnach steht die Nutzbarkeit stets im Zusammenhang mit den Anforderungen und Gegebenheiten der Umgebung des Systems.

## 2.7 Modal

[13] Ein Modalfenster ist ein überlagerndes Element in einer Benutzeroberfläche. Häufig wird dies verwendet, um spezifische Funktionen in den Fokus zu setzen, wie bei einer Auswahl von diversen Optionen. Modalfenster werden in der Regel für Eingaben oder Bestätigungen genutzt, ohne die Seite vollständig neu laden zu müssen.

## 2.8 Redakteurstätigkeiten

[14] Der Begriff Redakteurstätigkeiten umfasst im Kontext eines Content-Management-Systems wie TYPO3 die Aufgaben, die mit der Erstellung, Bearbeitung, Organisation und Veröffentlichung von Inhalten auf einer Webseite verbunden sind. Diese Tätigkeiten werden in der Regel von Redakteuren und Redakteurinnen durchgeführt, die für die Pflege und Verwaltung von Inhalten verantwortlich sind.

### 2.8.1 Aufgabenbereiche

[14] Redakteurstätigkeiten lassen sich also in verschiedene Bereiche unterteilen. So stehen bei der Inhaltserstellung das Verfassen und Eingeben von Texten im Mittelpunkt, sowie das Hochladen und Einbinden von Medien wie Bildern oder Videos. Außerdem gilt hier das Erstellen neuer Inhaltselemente wie Textabschnitte, Formulare oder Tabelle zum Bestandteil der Inhaltserstellung.

Als weiteres Aufgabengebiet gilt die Inhaltsbearbeitung. Diese umfasst unter anderem das Aktualisieren bereits bestehender Inhalte und auch das Überprüfen und Anpassen von Layouts. Ebenso lässt sich die Optimierung von Inhalten für Suchmaschinen durch Keywords oder Metadaten diesem Bereich zuordnen.

Redakteurstätigkeiten umfassen noch weitere Aufgaben wie ein integraler Bestandteil der Veröffentlichungsprozesse von neu erstellten oder bearbeiteten Inhaltselementen zu sein. Jedoch wird im Verlauf der Arbeit insbesondere ein Schwerpunkt auf die Bereiche der Inhaltserstellung und -bearbeitung gelegt. Konkret wird sich stark auf das Erstellen neuer Inhaltselemente in Form von Text und Medien bezogen. Ebenfalls wird das Bearbeiten bereits bestehender Elemente eine Rolle spielen, seien es Überschriften oder sonstige Elemente.

### 2.8.2 Redakteurstätigkeiten in TYPO3

[15] In TYPO3 werden Redakteurstätigkeiten innerhalb der Weboberfläche ausgeführt. Da TYPO3 ein extensives Rollenmanagement anbietet, kann vorab festgelegt werden, auf welche Module des CMS ein:e Redakteur:in Zugriff erhält. Befindet man sich im Seitenmodul, um beispielsweise neue Seiten anzulegen oder bestehende Seiten und dessen Inhaltselemente zu

bearbeiten oder zu erweitern, so kann ein Großteil dieser Aufgaben durchgeführt werden, ohne Programmierkenntnisse zu besitzen.

### **3 Grundlagen der Integration von Sprachsteuerung**

In diesem Abschnitt wird der theoretische Rahmen für die Integration von Sprachsteuerung in CMS mit Fokus auf TYPO3 dargelegt. Dabei werden grundlegende Konzepte und relevante Lösungsmöglichkeiten untersucht, die für die spätere Umsetzung und Bewertung beachtet werden.

#### **3.1 Relevanz von Sprachsteuerung in Content-Management-Systemen**

[16] Die Sprachsteuerung hat in den letzten Jahren erhebliche Fortschritte mit sich getragen und sich als benutzerfreundliche Schnittstelle etabliert. Nutzende können mit technischen Systemen interagieren, ohne herkömmliche Eingabemethoden wie Tastatur und Maus zu verwenden. In Content-Management-Systemen, die von Redakteuren und Redakteurinnen mit unterschiedlichen technischen Vorkenntnissen genutzt werden, kann die Integration einer Sprachsteuerung den redaktionellen Arbeitsfluss unterstützen. Die Frage danach, ob sich diese Technologie in ein bestehendes CMS einbinden lässt, ist von besonderem Interesse.

Nach aktueller Recherche bieten weder WordPress noch TYPO3 eine native Sprachsteuerung des Backends an. Auch Erweiterungen, die eine umfassende Sprachsteuerung ermöglichen, wurden kaum gefunden. Für TYPO3 konnte jedoch die Erweiterung „speech\_input“ identifiziert werden, die es Nutzenden ermöglicht, Text per Sprache in Feldern einzufügen, welche zuvor manuell ausgewählt werden müssen. Die Logik dieser Erweiterung basiert auf der Web Speech API. Allerdings wurde diese Erweiterung seit TYPO3 Version 9 nicht mehr aktualisiert, wodurch ihre Verwendbarkeit in neueren Version nicht garantiert wird.

Für Wordpress zeigen sich ähnliche Ergebnisse. Hier ergab sich, dass falls Erweiterungen eine Sprachintegration anbieten, diese fast ausschließlich für das Frontend der Website ausgerichtet ist. Sei es in Form von der Integration für Chatbots oder für das Befüllen von Textfeldern per Sprache.

#### **3.2 Konzeptionelle Überlegungen zur Integration von Sprachsteuerung**

Eine erfolgreiche Implementierung einer Sprachsteuerung in ein bestehendes Content-Management-System setzt eine konkrete Konzeption und die Berücksichtigung technischer und funktionaler Anforderungen heraus. Hierbei sollten insbesondere geeignete Schnittstellen in Form von APIs in Betracht gezogen werden. Diese Schnittstellen sollen verwendet werden, um Sprachbefehle zu empfangen, damit diese folgend in für TYPO3 spezifische Aktionen

umgesetzt werden können. In diesem Zusammenhang werden in weiterer Ausarbeitung unterschiedliche Möglichkeiten dargestellt. So könnte beispielsweise eine Einbindung der Alexa Development Tools [17] oder die vom W3C verwalteten Web Speech API [10] als Lösungsmöglichkeit angewendet werden.

[18] Da Content-Management-Systeme wie TYPO3 von Personen mit unterschiedlichem technischem Wissen verwendet werden, soll eine Implementierung von Sprachsteuerung besonders auf eine klare Art und Weise verwendet werden können, ohne weitreichendes technisches Wissen vorauszusetzen. Dadurch soll eine Nutzbarkeit mit möglichst wenig Aufwand garantiert werden.

## **4 Aktueller Stand der Forschung**

### **4.1 Relevante Studien zu Sprachassistenten**

Zahlreiche Studien belegen die wachsende Verbreitung von Sprachassistenten. Ein Großteil der deutschen Bevölkerung verwendet Sprachassistenten in diversen Ausführungen. Ob im Smartphone integriert oder als dedizierter Smart Speaker im Haushalt, Voice Assistants gewinnen an Bedeutung und Verwendungsmöglichkeiten. Umfragen [19] [20] zeigen, dass in etwa die Hälfte der deutschen Bevölkerung einen Sprachassistenten verwendet. Bevorzugt werden im Gesamtbild der Befragten grundsätzlich Amazon Alexa, Google Assistant und Siri von Apple. Besonders im Bereich der Smart Speaker sticht Alexa von Amazon mit einem Anteil von 65 Prozent hervor. [19] Sowohl im Gebrauch, als auch in der Entwicklung der Sprachassistenten gibt es stets Fortschritte. [21] Kontextbasierte Antworten, die Möglichkeit Dialogen zu folgen und der Einsatz von leistungsstarken Large Language Model wie ChatGPT von OpenAI sorgen dafür, dass der Umgang mit diversen Assistenten dem Menschen gegenüber natürlicher wird.

#### **4.1.1 Betrachtete Studien**

[21] Die Anzahl der weltweit verwendeten Sprachassistenten hat sich in den vergangenen Jahren signifikant erhöht. Zur Betrachtung der Entwicklung dessen wird ein großer Fokus auf den Markt in Deutschland gelegt. [22] Dies hängt damit zusammen, dass TYPO3 als Content-Management-System besonders in deutschsprachigen Gebieten weit verbreitet ist. So kann TYPO3 weltweit aus dem Standpunkt des Jahres 2020 einen Marktanteil von 0,6 Prozent aufzeigen. In Deutschland sieht es etwas anders aus, hier kann ein Marktanteil nach Stand 2019 von etwa 13 Prozent ausgemacht werden und in Österreich sogar einen Anteil von etwa 17 Prozent.

Zur Analyse des Wachstums der Nutzung von Sprachassistenten werden einige Studien als Grundlage dienen. Wird sich im weiteren Verlauf auf die Studie von MUUUH! [23] bezogen, handelt es sich um die im November 2023 veröffentlichte Studie der MUUUH! Group. Außerdem wird sich auf die Studie mit dem Titel „The Age of Voice 3.0“ [19], veröffentlicht von OMD Germany bezogen. Diese Studie wurde am 23. September 2021 veröffentlicht. Zudem werden Daten der WIK Kurzstudie [20], veröffentlicht im Dezember 2022, in Betracht gezogen.

Die Studie der MUUUH! Group [23] beruht sich auf eine durch die YouGov Deutschland GmbH durchgeführte Online-Umfrage. Dessen Teilnahmegruppe betrug eine Anzahl von 2014 Personen. Da diese Umfrage im Jahr 2023 durchgeführt wurde, stellt diese Studie die gegenwärtigsten Ergebnisse dar.

Eine Umfrage von 2014 Personen kann bereits einen guten Einblick in die aktuelle Nutzung von Sprachassistenten in Deutschland bieten. Um jedoch ein tatkräftigeres Ergebnis dahingegen zu erhalten, wie Voice Assistants in Deutschland heute aufgestellt sind und zukünftig wachsen könnten, sollen weitere Studien ebenfalls analysiert werden.

So bietet eine Stichprobe von 1504 Personen die Grundlage der Studie zu Sprachassistenten der OMD Germany GmbH [19]. Der Befragungszeitraum betrug sich hier auf den Zeitraum vom 07. Juli 2021 bis zum 15. Juli 2021.

Damit der Zeitraum zwischen den Jahren 2021 und 2023 ebenfalls mit einbezogen wird und das Ergebnis der Analyse auf diversifizierten Daten beruht, wird die WIK Kurzstudie [20] ebenfalls Gewicht tragen. Dessen Online-Befragung umschließt eine Stichprobe von 3254 Personen und wurde im November 2022 durchgeführt.

## **4.2 Aktuelle Nutzung von Sprachassistenten**

In diesem Kapitel zur aktuellen Nutzung von Sprachassistenten wird in verschiedenen Kategorien betrachtet, inwiefern und durch welche Methoden Sprachsteuerung heute verwendet wird.

### **4.2.1 Sprachassistenz im Gesamtbild**

Betrachtet man den aktuellsten Stand der zuvor vorgeführten Studien, so wird deutlich, dass stets ein Wert von etwa 50 Prozent aufgezeigt wird. Dieser Wert bezieht sich auf die existierende Nutzerbasis von Sprachsteuerung. So zeichnet die Studie von OMD Germany [19, p. 4] auf, 52 Prozent der befragten Personen seien im Jahr 2020 aktive Nutzende von Voice Assistants gewesen. Als aktive Nutzende werden Personen definiert, die mindestens drei- oder viermal im Monat Gebrauch eines Voice Assistants machen. Zusätzlich würden sich weitere sieben Prozent als zukünftige Nutzende also bis zum Ende des Jahres 2021, definieren. Eben-

falls haben laut dieser Befragung 69 Prozent der Stichprobe einen Voice Assistant in Vergangenheit bereits verwendet.

Wirft man einen Blick auf die WIK Kurzstudie [20] aus dem Jahr 2022, so ist zu erkennen, dass ebenfalls ein Wert von etwa 50 Prozent bezüglich der Nutzung von Sprachassistenten vorliegt. Hier wird kenntlich gemacht, dass 47 Prozent der insgesamt 3.254 befragten Personen mindestens einen Sprachassistenten nutzen. Weiterhin wird verdeutlicht, dass insgesamt 92 Prozent der Befragten jedoch mindestens ein Sprachassistent bekannt ist.

Die Studie der MUUUH! Group [23] hängt sich ebenfalls an die 50 Prozent Marke. Hier wird gekennzeichnet, dass 53 Prozent der 2.014 befragten Personen Geräte mit Voice-Funktionalität nutzen. Diese werden als Personen definiert, die regelmäßig oder gelegentlich mindestens ein Gerät mit Sprachfunktion nutzen.

#### **4.2.2 Entwicklung der individuellen Studien**

Sowohl die Studie der OMD Germany GmbH [19], als auch die WIK Kurzstudie [20] referenzieren Werte von durchgeführten Befragungen der vergangenen Jahre. So wird in der Studie der OMD Germany GmbH angegeben, 38 Prozent der im Jahr 2018 befragten Personen werden als aktive Nutzende [19] definiert. Zudem haben hier 61 Prozent der insgesamt 3.255 Befragten angegeben, einen Voice Assistant bereits verwendet zu haben. Im Jahr 2019 beträgt die Stichprobe eine Menge von 3.002 Personen, wobei 44 Prozent dessen als aktive Nutzende gekennzeichnet werden. Weitere sechs Prozent gelten als zukünftige Nutzende, welche innerhalb des ersten Quartals des Jahres 2020 planen, einen Sprachassistenten zu verwenden. Insgesamt geben 64 Prozent der Befragten an, einen Voice Assistant bereits verwendet zu haben.

Innerhalb der WIK Kurzstudie [20] aus dem Jahr 2022 mit der Angabe, 47 Prozent der befragten Personen nutzen mindestens einen Sprachassistenten, werden Werte aus dem Jahr 2018 referenziert. Hier wird angegeben, 26 Prozent der Stichprobe aus 3.254 Personen nutzen mindestens einen Sprachassistenten.

#### **4.2.3 Teilnehmerprofile der Studien**

Um die angegebenen Daten und den späteren Verlauf der Arbeit besser einordnen zu können ist es wichtig, einen Blick auf die Charakteristika der Stichproben zu werfen. Dies kann ein ausschlaggebender Bestandteil der weiteren Auswertung sein und sollte demnach nicht vernachlässigt werden.

So gehört zu der Stichprobenbeschreibung der Studie von OMD Germany [19, p. 18] aus dem Jahr 2021 eine Menge von 1.504 Personen, welche sich in ihrer Geschlechterverteilung zwischen 51 Prozent weiblich und 49 Prozent männlich einteilen lassen. In der Altersgruppe wird zwischen den Jahren 18 bis 29, 30 bis 39, 40 bis 49 und 50 bis 65 unterschieden. Hierbei lassen sich 21 Prozent der befragten Personen zur Altersgruppe von 18 bis 29 einordnen. Weite-

re 19 Prozent jeweils zu den Altersgruppen von 30 bis 39 und 40 bis 49. Die noch übrigen 41 Prozent der Stichprobe lassen sich zu einem Alter von 50 bis 65 zuordnen.

Innerhalb der WIK Kurzstudie [20, p. 5] wird die Gesamtstichprobe von 3.254 Befragten als 54 Prozent weiblich und 46 Prozent männlich angegeben. Sieben Prozent der Personen lassen sich einem Alter zwischen 18 und 24 Jahren zuordnen. Zu sowohl der Altersgruppe von 25 bis 34 Jahren, als auch von 35 bis 44 Jahren lässt sich ein Prozentsatz von 15 zuordnen. Die Altersgruppe von 45 bis 54 macht einen Anteil von 20 Prozent aus, wohingegen ein Alter von 55 Jahren oder über diesem, die restlichen 43 Prozent ausmachen.

Als letztes wird die Stichprobe der MUUUH! Voice Studie 2023 betrachtet. Hier [23, p. 13] wird angegeben, dass wie auch in der OMD Germany Studie, die Geschlechterverteilung zwischen 51 Prozent weiblich und 49 Prozent männlich eingeteilt wird. Bezüglich der Altersgruppen wird zwischen 18 bis 34 Jahren, 35 bis 54 Jahren und 55 Jahren und älter unterschieden. Ab einem Alter von 18 mit der Grenze von 34 Jahren lassen sich 24 Prozent der insgesamt 2.014 Befragten einteilen. Insgesamt 35 Prozent der befragten Personen gelten zu einem Alter von 35 bis 54 Jahre. Die restlichen 41 Prozent lassen sich als 55 Jahre oder älter zuordnen.

#### **4.2.4 Verwendung von Sprachassistenten nach Altersgruppen**

Laut den drei Studien aus den Jahren 2021 [19], 2022 [20] und 2023 [23] nutzen etwa die Hälfte der deutschen Bevölkerung eine Form von Sprachassistenten. Damit das Gesamtbild nachvollziehbarer gestaltet wird, wird nun die Nutzung von Sprachassistenten nach dem Alter eingeordnet. Dazu bieten die vorliegenden Studien unterschiedliche Datenpunkte. Die WIK Kurzstudie [20] gibt eine klare Aufteilung der Altersgruppen hinsichtlich deren Gebrauch an Sprachassistenten an. Ebenso unterscheidet die MUUUH! Studie [23] zwischen dem Alter in Betracht zu der Verfügbarkeit von Voice Assistants. Dahingegen differenziert die Studie der OMD Germany GmbH [19] nicht direkt hinsichtlich der vorliegenden Altersgruppen und deren Sprachsteuerungsgebrauch.

Die WIK Kurzstudie [20] gibt an, dass sich Personen in den Altersgruppen von 18 bis hin zu 44 Jahren, überwiegend als Nutzende von Sprachassistenten definieren lassen. Dies bedeutet, dass in dieser Altersspanne stets mehr als die Hälfte der Personen angibt, Nutzer:in von Sprachassistenten zu sein. Dieser Wert überschreitet jedoch in keiner der Altersgruppen einen Wert von 60 Prozent und bewegt sich grundsätzlich eher im mittleren Bereich der 50-Prozent-Marke. Es bezieht sich lediglich auf die Stichprobe der Studie. In der Spanne von 45 bis 54 Jahren erkennen sich 48 Prozent der Befragten als Nutzer:in von Sprachassistenten, wohingegen der Wert auf knapp unter 40 Prozent bei Personen im Alter von 55 oder älter fällt.

Wie bereits erwähnt weist die Studie der MUUUH! Group [23] Daten darüber auf, wie sich die Verfügbarkeit von dem Besitz mindestens eines Gerätes mit Sprachfunktion verteilt. So

geben 76 Prozent der 2.014 befragten Personen an, mindestens ein solches Gerät zu besitzen. Weitere elf Prozent geben an keines zu besitzen, jedoch Interesse vorzuzeigen. In der Altersgruppe von 35 bis 54 Jahren besitzen hier 66 Prozent der Befragten mindestens ein Gerät mit Voice-Funktionalität, wobei addierende zwölf Prozent angeben, keines zu besitzen aber Interesse zu zeigen. Betrachtet man die Altersgruppe von 55 Jahren und älter, so geben 49 Prozent der Personen an, mindestens ein Gerät mit Sprachfunktion zu besitzen, während weitere 19 Prozent keines besitzen, jedoch Interesse an eines haben.

#### **4.2.5 Sprache als Ersatz zu herkömmlichen Mitteln**

[23, p. 15] Spracheingaben können von diversen elektronischen Geräten verarbeitet werden. Ob ein Smart Speaker von Amazon, welcher Alexa integriert und auf eine Frage antwortet, oder am Smartphone per Sprache ein Wecker gestellt wird, Sprachassistenten sind den meisten Menschen in Deutschland in Reichweite. Grundsätzlich kann eine implementierte Spracherkennung und dessen Verarbeitung als Ersatz von herkömmlichen, traditionellen Geräten wie Maus und Tastatur dienen.

Demnach stellt die Studie der OMD Germany GmbH [19] die Frage, ob Sprachsteuerung einen Mehrwert in der Gegenüberstellung zu dem Eingabemittel der Tastatur bieten kann. Die Antworten auf diese Frage wurden folglich in den Kategorien von Intensivnutzende von Sprachsteuerung, Gelegenheitsnutzende und Nichtnutzende unterschieden. Betrachtet werden diesbezüglich fast ausschließlich angegebene Daten aus dem Jahr 2021, da die Datenpunkte aus den vergangenen Jahren oftmals keine großen Unterschiede aufweisen, ausgenommen der gegebenen Daten bezüglich der Nichtnutzenden.

Personen aus der Kategorie von Intensivnutzenden [19] geben im Jahr 2021 zu 30 Prozent an, Sprachsteuerung würde keinen Mehrwert im Vergleich zu Tastatureingaben bieten. Die Stichprobe beträgt hierbei eine Anzahl von 328 Personen. Im Jahr 2018 betrug dieser Wert 29 Prozent bei einer Stichprobenmenge von 580 Personen. Hinsichtlich der Gelegenheitsnutzende, welche aus einer Personenmenge von 449 im Jahr 2021 bestehen, geben 26 Prozent an, Sprachsteuerung würde in diesem Kontext keinen Mehrwert bieten. Im Jahr 2018 lag dieser Wert bei 25 Prozent. Bezüglich der Personen, die sich als keine Nutzenden von Sprachsteuerung definieren lassen, gaben im Jahr 2018 insgesamt 35 Prozent der Personen an, Sprachsteuerung würde keinen Mehrwert gegenüber Eingaben durch Tastaturen bieten. In der Umfrage aus dem Jahr 2019 lag dieser Wert bei 44 Prozent und im Jahr 2021 bei 50 Prozent.

#### **4.2.6 Bedenken zur Sprachsteuerung**

Um in der Implementierung einer Sprachsteuerung jeglicher Art die Bedenken von Nutzenden in Betracht ziehen zu können, bieten Studien Informationen dazu, in welchen Bereichen Skepsis gegenüber Voice-Technologien herrschen.

[19, pp. 15, 16] So stimmen im Jahr 2018 43 Prozent der befragten Intensivnutzenden der Aussage zu, Angst davor zu haben, durch Smart Speaker überwacht zu werden. Dieser Wert liegt bei den Gelegenheitsnutzenden bei 54 Prozent, während Nichtnutzende dieser Aussage zu 64 Prozent zustimmen. Im Jahr 2019 steigt der Wert bei den Intensivnutzenden auf 45 Prozent an. Ebenso stimmen in diesem Jahr 59 Prozent der Gelegenheitsnutzenden und ganze 70 Prozent der Nichtnutzenden dieser Aussage zu. Folglich sinkt dieser Wert im Jahr 2021 bei Intensivnutzenden auf 38 Prozent, während 45 Prozent der Gelegenheitsnutzenden der Befragung zustimmen. In dem Fall der Nichtnutzenden fällt der Datenpunkt auf dieselbe Höhe wie bereits im Jahr 2018, nämlich auf 64 Prozent.

Die WIK Kurzstudie [20, pp. 15, 16] bietet einen tieferen Einblick darin, aus welchen Gründen Nichtnutzende von Sprachassistenten diese Technologie nicht verwenden. So geben 36 Prozent der befragten Nichtnutzenden an, der Technologie schlichtweg nicht zu vertrauen. Weitere 34 Prozent geben an, Angst davor zu haben, dass dessen Daten an Dritte weitergegeben werden. Zudem gibt mit 33 Prozent ein Drittel der Stichprobe an, es wäre unangenehm mit Geräten zu sprechen. Ebenso relevant für die weiterführende Ausarbeitung ist die Angabe von 29 Prozent der Befragten, sie hätten Angst abgehört zu werden. Zudem geben 17 Prozent der Personen an, die Sprachassistenten erkennen die Sprachbefehle nicht einwandfrei. Weitere angegebene Gründe für die Enthaltung von Nutzung gegebener Sprachassistenten können sich der folgenden Abbildung 1 [20, p. 15] entnommen werden.

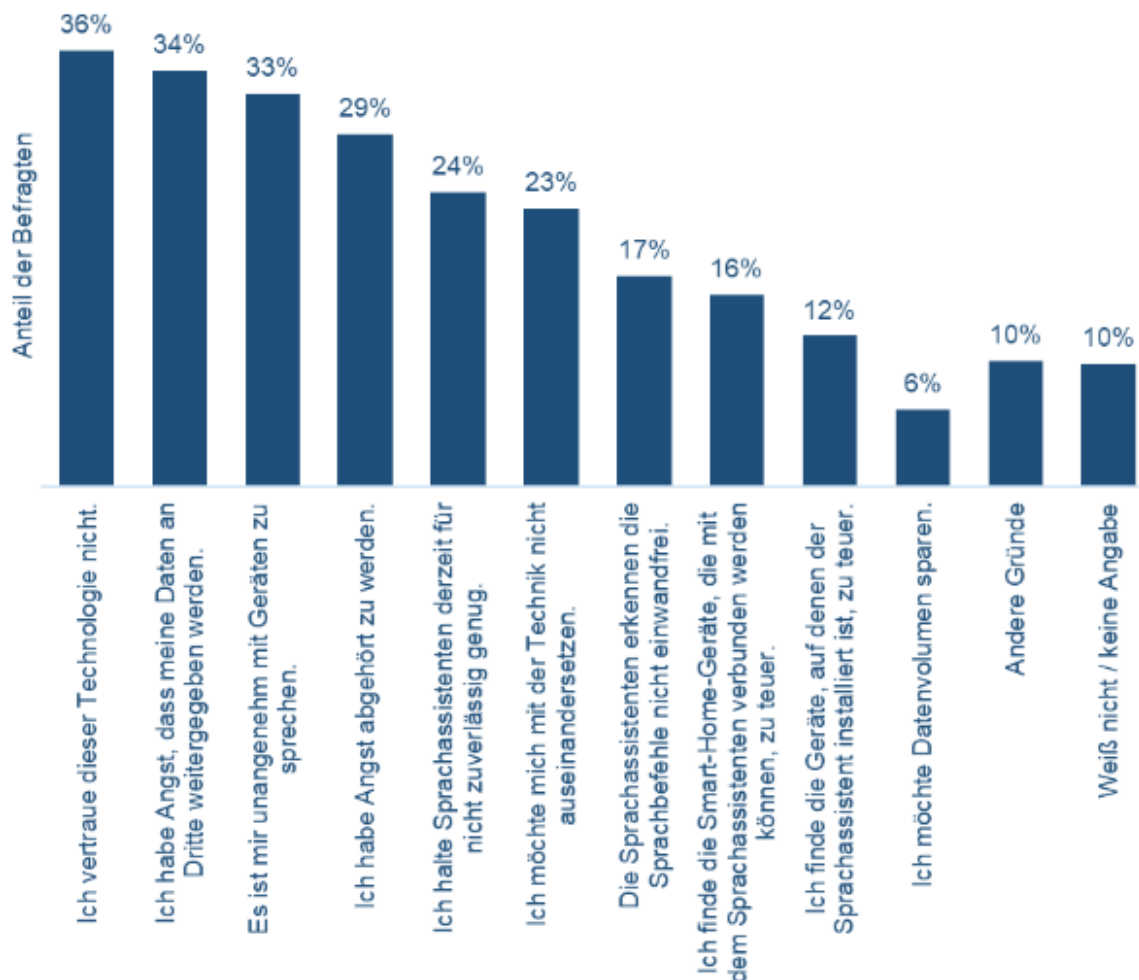


Abbildung 1: Gründe für Nichtnutzung WIK Studie

Auch gibt die MUUUH! Studie [23] aus dem Jahr 2023 Daten darüber, dass 55 Prozent der Stichprobenmenge von 2.014 Personen dem Bedenken hinsichtlich der Datensicherheit mindestens eher zustimmen. Diesem stimmen weitere 30 Prozent teils zu. Die direkte Aussage, dem die Befragten ausgesetzt waren, wäre in diesem Falle „Ich habe Bedenken in punkto Datensicherheit, wenn ich in der Kommunikation mit Geräten oder Unternehmen nur die Stimme nutze. Lediglich 15 Prozent geben hier an, es treffe eher nicht zu, bis es treffe gar nicht zu.

### 4.3 Auswertung und Trends

Die Studien sind sich einig, der Gebrauch von Sprachsteuerung steigt. So zeigt jede Studie, mit Gegenüberstellung von Referenzdaten aus vorherigen Befragungen, ein Wachstum zu den vorherigen Jahren. In der Stichprobe der OMD Germany [19] wäre der Anteil von Nutzenden der Sprachsteuerung jeglicher Art im Jahr 2021 bereits bei 59 Prozent.

Ob in dieser Studie, in der der prozentuale Anteil von Nutzenden im Jahr 2018 noch bei 38 lag, oder in der WIK Kurzstudie [20], es ist ein enormer Wachstum der vergangenen Jahre zu

verzeichnen. Denn auch hier ist zu sehen, dass im Jahr 2018 lediglich 26 Prozent der Befragten angegeben haben, Nutzer:in von Sprachassistenten zu sein. Im Jahr 2022 stieg dieser Wert auf 47 Prozent an. Dies zeigt einen Anstieg von über 80 Prozent in nur vier Jahren.

Jede dieser Studien wurde mit einer diversifizierten Altersgruppe durchgeführt. Demnach wurden stets Personen mit einem Alter ab 18 Jahren bis hin zu mindestens 50 Jahren oder älter einbezogen. Insbesondere durch die Auswertung der WIK Kurzstudie [20] wird deutlich, dass Personen im Alter von 18 Jahren bis 44 Jahren zu einer Mehrheit Sprachassistenten nutzen. Darüber hinaus verwenden Personen in der Altersgruppe von 45 Jahren bis 54 Jahren knapp unter 50 Prozent Sprachassistenten. Dieser Wert fällt ab einem Alter von 55 Jahren der befragten Personen auf unter 40 Prozent.

Dies könnte eine Antwortmöglichkeit dort hingegen bieten, wieso die Studien sich hinsichtlich der Befragung danach, ob Personen Sprache als Eingabemittel jeglicher Art nutzen, kaum unterscheiden. Betrachtet man jede der drei Studien individuell, so ist ein klares Wachstum zu vermerken. Die Studie der OMD Germany GmbH [19] gibt an, bis zu 59 Prozent der Stichprobe würden bis zum Ende des Jahres 2021 mindestens einen Voice Assistant nutzen.

So auch teilt die WIK Kurzstudie [20, pp. 13, 17] mit, dass in etwa 33 Prozent der Ablehner von Sprachassistenten aus dieser Stichprobe angeben, die Nutzung eines Sprachassistenten in den nächsten zwölf Monaten nicht auszuschließen. Dies spiegelt sich in der MUUUH! Studie [23] wider, da auch hier 36 Prozent der Befragten, welche zu dem Zeitpunkt als keine privaten Voice-Nutzenden definiert wurden, der Aussage zustimmen, Sprache würde im privaten Umfeld in Zukunft eine wichtige Rolle spielen.

Dennoch wird in jeder der Studien hinsichtlich der Frage danach, ob befragte Personen Nutzende von Sprachsteuerung oder Sprachassistenten sind, ein Wert von etwa 50 Prozent vermerkt. In der Studie aus dem Jahr 2022 lag dieser Wert sogar unter der 50-Prozent-Marke.

Dies widerspricht dem prognostizierten Wachstum. Es ist auf der einen Seite schwierig zu sagen, ob der Anteil der deutschen Bevölkerung, welcher Sprache als Eingabemedium verwendet, tatsächlich stagniert, oder ob es lediglich Diskrepanzen zwischen den Datenpunkten gibt. Der Aspekt des Alters könnte hier eventuell auch eine Rolle spielen. Grundsätzlich ergibt die Auswertung der Altersgruppen, dass je jünger die Person, desto höher ist die Wahrscheinlichkeit, dass Sprachsteuerung in irgendeiner Form verwendet wird. Nun liegen zu den Studien keine genauen Zahlen zu jedem Alter vor, jedoch werden grobe Wertespanssen übermittelt. So wird in der Studie aus dem Jahr 2021 [19, p. 18] angegeben, 41 Prozent der Befragten wären 50 Jahre alt oder älter. In der WIK Studie [20, p. 5] aus dem Jahr 2022 wird angegeben, 22 Prozent der Stichprobe befinden sich unter der Grenze im Lebensjahr von 34 Jahren, während 43 Prozent sich im 55. Lebensjahr befinden. Die Stichprobenmenge der MUUUH! Studie aus dem Jahr 2023 [23, p. 13] setzt sich wiederum aus Personen zusammen,

welche zu 24 Prozent bis zu 34 Jahre alt sind, während 41 Prozent das Alter von 55 Jahren und drüber inkludieren. Nun könnte man dadurch grob davon ausgehen, dass die Stichprobenmenge der WIK Kurzstudie ein höheres Durchschnittsalter nachweisen könnte, in Gegenüberstellung mit den Vergleichsstudien. Dies kann nicht klar belegt werden, da wie bereits erwähnt, keine Teilnehmerliste mit konkretem Lebensjahr individueller Teilnehmer:innen vorliegt. Als weitere Stütze der Behauptung, dass das Durchschnittsalter das Ergebnis beeinflussen könnte, wären Werte, welche von der Bitkom e.V. [24], ein Branchenverband der deutschen Informations- und Telekommunikationsbranche, im Jahr 2023 veröffentlicht wurden. So resultiert hier aus der Umfrage von 1.159 Personen in Deutschland das Ergebnis, dass 57 Prozent zumindest gelegentlich von Sprachsteuerung Gebrauch machen. Die jüngste Altersgruppe wird hier als 16 Jahre bis 29 Jahre definiert, wobei der Wert in diesem Falle auf 74 Prozent ansteigt.

Ob es in Zukunft zu großem Wachstum oder Stagnation kommt, der allgemeine Konsens bleibt: Aktuell nutzt etwa die Hälfte der Menschen in Deutschland Sprachsteuerung in irgendeiner Form. Die Erweiterung eines Systems um Sprachsteuerung, in unserem Fall ein TYPO3-Backend, könnte nicht nur die Barrierefreiheit verbessern, sondern auch potenziell eine große Zielgruppe im Arbeitsfluss unterstützen. Bedenken bezüglich der Verwendung von Sprachsystemen sollten bei der Implementierung berücksichtigt werden, sei es hinsichtlich des Datenschutzes, der Angst vor dem Abhören oder die Genauigkeit der Nutzereingaben.

## **5 Methodik**

Nachdem die Frage nach einer Zielgruppe geklärt wurde, wird nun in diesem Abschnitt auf die Methodik der Implementierung einer Sprachsteuerung in dem Content-Management-System TYPO3 eingegangen.

### **5.1 Anforderungsanalyse**

Zunächst erfolgt eine Anforderungsanalyse dort hingegen, welche Anforderungen an die Software gelegt werden. Dazu gehören Klarstellungen der verwendeten Software und dessen Kompatibilität mit anderen Versionen. So wird insbesondere die verwendete TYPO3 Version verdeutlicht und die Frage nach vorhandener Abwärtskompatibilität beantwortet. Da lediglich in der TYPO3 Version 12.4.19 entwickelt wird, muss die Extension weder abwärtskompatibel zu Vorgängerversionen, noch Kompatibilität zu nachfolgenden Versionen gewährleisten. Dies hängt damit zusammen, dass die Implementierung als Kernpunkt verdeutlichen soll, ob und inwiefern eine Extension für das CMS entwickelt werden kann, die Redakteurstätigkeiten unterstützt. Nutzende sollten bestmöglich die Erweiterung herunterladen und ohne großartige Konfiguration nutzen können. Dadurch fordert das Programm kein extensives technisches

Hintergrundwissen. Ebenso soll eine Implementierung in Betracht gezogen werden, die bis auf ein vorhandenes Mikrofon keine weiteren externen Geräte beansprucht. Wäre dies nicht gegeben, könnte eine Anschaffung eines Smart Speakers als finanzielle Hürde zur Nutzung des Programms darstellen.

Da TYPO3 als CMS über eine Weboberfläche zu erreichen ist, muss abgesehen von der verwendeten Hardware auch die Software in Form des Browsers beachtet werden. [25] Hier gilt mit etwa 44,5 Prozent Google Chrome als meist genutzter Browser der deutschen Bevölkerung für Desktops, gemessen als Marktanteil durch gesamte Page Views, wobei Mozilla Firefox mit rund 18,9 Prozent den zweiten Platz belegt. Damit die Extension von der größtmöglichen Anzahl an Personen verwendet werden könnte, soll das Ergebnis hinsichtlich beider Browser getestet werden.

Zusätzlich muss die Implementierung in Bezug auf Zugänglichkeit und Nutzbarkeit diverse Möglichkeiten darstellen. Wie zuvor durch diverse Umfragen bestätigt, gilt Datenschutz und die Angst vor dem Abhören von Sprachsoftware beziehungsweise Sprachassistenten als Bedenken von Nutzern. Könnte eine kontinuierlich aktivierte Sprachsteuerung sowohl die Nutzbarkeit als auch die Zugänglichkeit erhöhen, so könnte dies jedoch als intrusiv empfunden werden. Um ein konstantes Gefühl des Abhörens zu vermeiden, müssen Alternativen geboten werden. Ebenfalls soll verdeutlicht werden, ob und inwiefern Redakteurstätigkeiten durch implementierte Sprachsteuerung durchgeführt werden können. Zudem sollte die Navigation insbesondere des Seitenmoduls im TYPO3 Backend per Sprachsteuerung durchgeführt werden können, da dieses Modul ein Hauptbestandteil der Redakteurstätigkeiten darstellt.

## **5.2 Entwicklungsprozess**

Um die Möglichkeit einer Implementierung von Sprachsteuerung in TYPO3 hinsichtlich der zuvor definierten Redakteurstätigkeiten und der Navigation der Weboberfläche zu analysieren, sollen diverse Entwicklungsmodelle verwendet werden.

Etwa ein Drittel der deutschen Haushalte besitzen zum Stand des Jahres 2023 mindestens einen Smart Speaker. Betrachtet man die Angaben aus der Studie der OMD Germany GmbH [19, p. 5], so besaßen hier im Jahre 2021 insgesamt 33 Prozent der Befragten mindestens einen dieser intelligenten Lautsprecher. Im Falle der MUUUH! Studie [23, p. 15] sieht es ähnlich aus, denn hier geben 32 Prozent der befragten Personen an, einen Smart Speaker zu besitzen.

Wirft man nun einen Blick auf die Verteilung dessen, welche Art von Sprachassistenten hierbei am häufigsten gewählt wird, so wird deutlich, dass Amazon Alexa die Nummer eins darstellt. Denn die OMD Studie [19, p. 9] gibt an, dass 65 Prozent der Besitzer:innen eines Smart

Speakers primär Alexa verwenden. An zweiter Stelle steht Google Assistant mit einem Wert von 14 Prozent.

Die Verknüpfung vom Sprachassistenten der Wahl und dem Content-Management-System, mit dem gearbeitet wird, könnte eine bereichernde Option für Nutzende darstellen, sollte Bedarf an Sprachsteuerung bestehen. Personen könnten eine TYPO3 Erweiterung und ein bereits vertrautes Gerät verwenden.

Dadurch könnte die Einstiegshürde verringert und eine Grundlage dafür geboten werden, eine Implementierung von Alexa als Sprachsteuerung für Redakteurstätigkeiten und Navigation des Backends durchzuführen.

Um jedoch eine größtmögliche Anzahl an Personen erreichen zu können, sollte weiterhin eine Implementierungsmöglichkeit ohne externe Geräte betrachtet werden.

### **5.3 Implementierung Google Home**

Im Rahmen der Erweiterung des CMS TYPO3 zur Unterstützung von Sprachsteuerungsfunktionen wird im Folgenden die Möglichkeit der Integration eines Google Home Smart Speakers analysiert. Ziel ist es hierbei, den Google Assistant als zentrale Schnittstelle zu nutzen, um die Redakteurstätigkeiten und die Navigation der Weboberfläche zu unterstützen. Die Integration dessen soll Nutzenden eine alternative Methode zur klassischen Bedienung mittels Maus und Tastatur bieten und den Aspekt der Barrierefreiheit fördern.

Jedoch stellt die fehlende Unterstützung seitens Google für Drittanbieterintegrationen ein Hindernis dar. Seit einer Neuausrichtung hinsichtlich Google Assistant ist die direkte Implementierung von Google Home in eigenentwickelte Systeme eingeschränkt. Dies hängt damit zusammen, dass so genannte „Conversational Actions“ [26] seit dem 13.06.2023 nicht mehr unterstützt werden. Stattdessen wird ein neues Projekt eingeführt, das Entwickler für Apps eine Alternative bietet. Dies trägt für uns die Folge, dass keine offizielle Möglichkeit besteht, eigene Sprachbefehle auf Google Home Geräten zu nutzen, um somit TYPO3 per Sprache zu steuern.

Für die Implementierung bedeutet dies, dass Alternativlösungen betrachtet werden müssten.

### **5.4 Implementierung Amazon Alexa**

In diesem Abschnitt wird die Implementierung einer Sprachsteuerung mit Amazon Alexa für das TYPO3 Backend untersucht. [17] Amazon Alexa bietet eine so genannte Developer Console und über die „Alexa Skills Kit (ASK)“-Entwicklerplattform weitreichende Möglichkeiten zur Integration individueller Sprachbefehle, die mit TYPO3 interagieren können. Folglich werden Schritte zu einem Implementierungsweg gekennzeichnet.

### **5.4.1 Erstellung von TYPO3-Endpunkten**

Um die Kommunikation zwischen Alexa und TYPO3 zu ermöglichen, müssen spezielle Endpoints im TYPO3-Backend erstellt werden. Diese Endpoints dienen als Schnittstellen, welche Sprachbefehle von Alexa annehmen und entsprechende Aktionen im TYPO3 Backend auslösen. Hierbei würde es sich um REST-API-Endpunkte handeln, die spezifische Funktionen im Backend durchführen. So könnten beispielsweise Texte eingefügt oder die Navigation zu bestimmten Bereichen der TYPO3 Weboberfläche ermöglicht werden.

### **5.4.2 Amazon Developer Console**

Damit die Sprachbefehle der Nutzenden die Möglichkeit haben, im eigenen TYPO3 Backend verarbeitet zu werden, müssen individuelle Alexa Sprachbefehle erstellt werden. Um mit der Entwicklung eines Alexa Skills zu beginnen, ist eine Registrierung in der Amazon Alexa Developer Console erforderlich. Hier kann nach erfolgreichem registrieren ein neues Projekt angelegt werden, in dem die benötigten Sprachbefehle definiert werden können.

### **5.4.3 Individuelle Sprachbefehle**

In der Entwicklungsumgebung für die Amazon Alexa Smart Speaker werden die benötigten Sprachbefehle (Intents) für die TYPO3 Erweiterung definiert. Diese Sprachbefehle sollten die wichtigsten Redakteurstätigkeiten und Navigationsaktionen umfassen, wie beispielsweise das Anlegen eines neuen Inhaltselements oder das Wechseln von Ansichten innerhalb des Seiten Editors. Hierzu können klare Befehle wie „Erstelle einen neuen Inhalt“ oder „Springe zum edit Tab“ eingerichtet werden. Jeder dieser Sprachbefehle wird mit einem so genannten Intent verknüpft, der wiederum einen vorher erstellten TYPO3 Endpunkt aufruft. Dabei ist die klare und eindeutige Definition von Befehlen zu beachten, um Struktur zu gewährleisten.

### **5.4.4 Codeimplementierung von Sprachbefehlen**

Damit die erstellten Sprachbefehle direkt an das TYPO3 Backend gesendet und dort ausgeführt werden können, muss sichergestellt werden, dass das jeweilige Amazon Alexa Gerät Zugriff auf die eigene TYPO3 Backend URL besitzt. Hierfür wird die URL in die Logik des Alexa Skills eingetragen und somit als Zieladresse für die Anfragen verwendet. Da die Endpunkte, die innerhalb von TYPO3 erstellt werden über HTTP-Anfragen ansprechbar sein sollen, könnte für jeden Alexa Skill eine Logik implementiert werden, die spezifische Aufgaben im TYPO3 Backend ausführt.

### **5.4.5 Authentifizierung Alexa und TYPO3**

[5] Im TYPO3 Backend handeln Nutzende nach erstellten Benutzerprofilen mit individuellen Anmeldedaten. Demnach sollte bei der Implementierung von Alexa zu dem CMS auch ein entsprechendes Authentifizierungsmodell eingebunden werden.

Es muss sichergestellt werden, dass nur berechtigte Anfragen auch vom System verarbeitet werden, um unbefugte Zugriffe oder Manipulationen zu vermeiden. Ein weit verbreitetes Modell für die Authentifizierung solcher Schnittstellen ist die Verwendung des OAuth-Standards. Open Authorization [27] steht für ein offenes Protokoll, welches es ermöglicht, Anwendungen den Zugriff zu gewähren, ohne Passwörter auf direktem Wege auszutauschen. Stattdessen werden sogenannte Zugriffstoken generiert, welche die Berechtigung für spezifische Aktionen darstellen. Solch ein Token würde bei jeder Anfrage vom Alexa Skill an das TYPO3 Backend gesendet und dort überprüft. Anforderungen würden nur ausgeführt werden, wenn das gesendete Token als gültig anerkannt wird.

#### **5.4.6 Entwicklung der TYPO3 Erweiterung in PHP**

[28] Die tatsächliche Integration von Alexa in TYPO3 wird größtenteils in PHP entwickelt, da dies die primäre Sprache für TYPO3 Erweiterungen darstellt. Die Implementierung basiert auf der Erstellung von REST-API-Endpunkten, die spezifische Funktionen im TYPO3 Backend ausführen.

Um dies technisch umsetzen zu können, wird das Routing in TYPO3 [29] genutzt, das in PHP in der `Routes.php` Datei der Extension definiert wird. Das Routing legt fest, welche Controller-Methoden die jeweiligen URLs aufrufen sollen und wie die Anfragen an die richtige Logik weitergeleitet werden. Eine dieser Routen könnte beispielsweise den Pfad `„/alexa/create-content“` umfassen und auf eine spezifische Methode im Controller der Extension zeigen. Diese Methode verarbeitet die Anfrage und führt die entsprechenden Aktionen im TYPO3 Backend durch.

Die Struktur der Endpunkte sollte gemäß den PSR-7-Standards [30] implementiert werden. Hierbei handelt es sich um einen Industriestandard für HTTP-Anfragen und Antworten in PHP. Dieser Standard gibt an, wie HTTP-Nachrichten strukturiert sind und wie Anfragen und Antworten verarbeitet werden, wodurch eine konsistente Grundlage gewährleistet wird. In der Implementierung bedeutet es, dass die HTTP-Anfragen von Alexa an die entsprechenden TYPO3 Endpunkten in einem standardisierten Format eingehen, wodurch eine sichere und zuverlässige Interaktion möglich wird. Anfragen werden als JSON-Body übertragen und in der eigenen Controller Logik verarbeitet, indem relevante Informationen an die TYPO3-API übergeben werden.

Eine konkrete Implementierung dieser Controller Logik erfolgt in der zu entwickelnden Extension in PHP. Um dies an einem Beispiel darzustellen, könnte Alexa den Befehl erhalten, ein neues Inhaltselement in TYPO3 zu erstellen. Der erstellte Alexa Skill hierzu sendet eine HTTP-POST-Anfrage an den entsprechenden Endpunkt der TYPO3 Erweiterung. Daraufhin wird in der Controller Methode die TYPO3-DataHandler-Klasse [31] verwendet, welche als API für Datenoperationen im TYPO3-Backend dient. Hierüber kann ein neues Inhaltselement

mit den übergebenen Parametern erstellt werden. Sobald diese Aktion erfolgreich abgeschlossen wurde, kann Alexa eine Meldung an den Nutzern ausgeben.

Da hierdurch eine Echtzeitdarstellung standardgemäß nicht gegeben wäre, da Alexa nicht den Browser der Nutzende steuert, sondern direkte Anweisungen an das TYPO3 Backend durchführt, müssten weitere Technologien eingesetzt werden. Wird in diesem Arbeitsfluss ein neues Inhaltselement erstellt, so wird es nicht direkt in der eigenen Weboberfläche ersichtlich, sondern erst dann, wenn die Seite manuell neu geladen wird.

## 5.5 Implementierung Web Speech API

Um eine Echtzeitinteraktion mit der TYPO3 Weboberfläche zu ermöglichen, könnte die Integration einer Web Speech API in Verbindung mit JavaScript und PHP betrachtet werden. Die Web Speech API [10], ist eine browserbasierte Technologie, die es ermöglicht, Sprachbefehle in Echtzeit zu erkennen und zu verarbeiten. Zuvor wurde dargelegt, dass Google Chrome mit einem Marktanteil von 44,5 Prozent [25], gemessen an Page Views, als meist verwendeter Browser in Deutschland gilt. Dadurch, dass die zu verwendende API von Google Chrome unterstützt wird, wird dessen Implementierung nun genauer analysiert.

Die Web Speech API [10] ermöglicht die Umwandlung von Sprachbefehlen in Text, der anschließend in der TYPO3-Weboberfläche verarbeitet werden kann. Dadurch, dass TYPO3 bereits eine Vielzahl an API Endpunkten [31] bereitstellt, erleichtert dies die Implementation. Im PHP-Bestandteil der Erweiterung kann die JavaScript Datei, in der die Web Speech API eingebunden wird, integriert werden. Dieser JavaScript Code könnte in verschiedenen Varianten ausgeführt werden. Möchte man, dass Nutzende für jeden Sprachbefehl zuvor eine manuelle Anweisung tätigen müssen, so kann in einem PHP Code ein Button zu der TYPO3 Weboberfläche hinzugefügt werden. Weiterhin kann die Sprachaktivierung integriert werden, indem bereits bei erfolgreichem Laden der Seite, diese aktiviert wird. Somit wäre eine kontinuierliche Nutzung der Sprachsteuerung ohne weitere Hindernisse möglich. Damit Nutzende nicht das Gefühl vom Abhören aufgedrängt wird, könnte hierfür ebenfalls ein Schalter in der Oberfläche hinzugefügt werden. Dadurch kann ein Nutzende stets entscheiden, ob die Hauptfunktion der Erweiterung aktiviert ist, oder nicht.

Wie bereits erwähnt wird die Logik der Web Speech API grundsätzlich im JavaScript Code implementiert. Dies bedeutet, dass hier auch festgelegt wird, welche Sprachbefehle existieren und welche Funktionen entsprechend ausgeführt werden sollen. Der PHP Anteil der Erweiterung sorgt dafür, dass diese Logik als Bestandteil des TYPO3 Backends integriert wird. Sei es hinsichtlich der Implementation in die Weboberfläche in Form eines Buttons oder auch das grundlegende Laden beziehungsweise Ausführen des JavaScript Codes.

Um eine genaue Veranschaulichung zu bieten, wie solch eine Implementierung gestaltet werden kann, werden folglich einige Code Ausschnitte erläutert.

### 5.5.1 JavaScript

```

4  try {
5      // Variablen initialisieren
6      var recognizing :boolean = false;
7      var infiniteMode :boolean = false; // Zustand für den "unendlich" Modus
8      var SpeechRecognition = window.SpeechRecognition || window.webkitSpeechRecognition;
9      var recognition = new SpeechRecognition();
10     recognition.continuous = true;
11     recognition.lang = 'de-DE';
12
13     // Spracherkennung starten
14     function startSpeechRecognition() :void {
15         if (!recognizing) {
16             recognition.start();
17             recognizing = true;
18             console.log("Speech recognition started.");
19         }
20     }
21
22     // Spracherkennung stoppen
23     function stopSpeechRecognition() :void {
24         if (recognizing) {
25             recognition.stop();
26             recognizing = false;
27             console.log("Speech recognition stopped.");
28         }
29     }

```

**Abbildung 2:** Codeausschnitt Grundaufbau JavaScript

Die oben angegebene Abbildung 2 zeigt den grundlegenden Aufbau dazu, wie der JavaScript Code gestartet wird, sobald dies angefordert wird. Hier sind einige Grundbausteine vorhanden, wie das Initialisieren von `SpeechRecognition`, wobei darauf geachtet wird, dass ebenfalls das `webkit` Präfix in Zeile acht erlaubt wird. Dies hängt damit zusammen, dass einige Browser, wie auch ältere Versionen von Google Chrome, das Präfix verwenden. Im Beispiel von Google Chrome wird das Präfix ab Version 33, also seit Februar des Jahres 2014 nicht mehr verwendet. [10] Die offizielle Schnittstelle für die Spracherkennung gemäß den Webstandards wäre `window.SpeechRecognition`.

Damit ein Sprachkanal kontinuierlich geöffnet sein kann, könnte man davon ausgehen, dass dies durch Zeile zehn gelöst ist. Jedoch sorgt dies lediglich dafür, dass Sprachbefehle aneinandergereiht werden können. Wird in einer gewissen, nicht eindeutig dokumentierten Zeit-

spanne keine Spracheingabe aufgezeichnet, so wird der Sprachkanal geschlossen und müsste manuell vom Nutzenden wieder geöffnet werden, beispielsweise durch einen Button.

```
104 // Wenn die Spracherkennung endet
105 recognition.onend = function () :void {
106     recognizing = false;
107     console.log("Speech recognition ended.");
108
109     // Neustart nur im unendlich/toggle Modus
110     if (infiniteMode) {
111         console.log("Restarting speech recognition in infinite mode...");
112         startSpeechRecognition();
113     }
114 };
115
```

Abbildung 3: Codeausschnitt Sprachkanal

Um einen Sprachkanal offen zu halten, wird dies in Zeile 105 bis 114 implementiert. Dort wird festgelegt, dass sobald ein Sprachkanal geschlossen wird, weil seit längerer Zeit keine Sprache festgestellt wurde, die Spracherkennung automatisch neu gestartet werden soll, falls der Toggle aktiviert wurde.

```
31 // Event Listener für den "normalen" button
32 $(document).on('click', '.speech-action', function (event) :void {
33     event.preventDefault();
34     console.log("Speech action button clicked! Starting recognition...");
35     infiniteMode = false;
36     startSpeechRecognition();
37 });
38
39 // Event Listener für den Toggle Button
40 $(document).on('click', '.toggle-action', function (event) :void {
41     event.preventDefault();
42     $(this).toggleClass('active');
43     infiniteMode = $(this).hasClass('active');
44
45     if (infiniteMode) {
46         console.log("Infinite mode activated.");
47         startSpeechRecognition();
48     } else {
49         console.log("Infinite mode deactivated.");
50         stopSpeechRecognition();
51     }
52 });
```

Abbildung 4: Codeausschnitt Sprachsteuerung

Die Code Abschnitte, welche sich durch Zeilen 32 bis 52 strecken, wird der Start der Sprachsteuerung eingeleitet, sobald ein Ereignis auf den entsprechenden Button verzeichnet wird.

```

54 // Definieren der Sprachfehle
55 recognition.onresult = function (event) :void {
56     var transcript :string = event.results[event.resultIndex][0].transcript.trim().toLowerCase(); //immer neuestes Ergebnis verwenden
57     console.log("Recognized speech:", transcript);
58     //wenn die Eingabe 'setze datum' enthält, so verwende alles folgende (bis auf 'auf') als Eingabe
59     if (transcript.includes('setze datum')) {
60         const dateString :string = transcript.replace( searchValue /. *setze datum( auf)?\s*/, replaceValue '').trim();
61         setDateInFlatpickr(dateString);
62     } else if (transcript.includes('create new content') || transcript.includes('neues inhaltselement')) {
63         createNewContentElement();
64     } else if (transcript.includes('speichern')) {
65         saveContent();
66     } else if (transcript.includes('speichern und schließen')) {
67         saveAndCloseContent();
68     } else if (transcript.includes('fenster schließen')) {
69         closeForm();
70     } else if (transcript.includes('seite anzeigen')) {
71         viewPage();
72     } else if (transcript.includes('tab')) {
73         const tabName :string = transcript.replace( searchValue /. *(\w+)\s+tab\b.*/, replaceValue '$1').trim(); //beispielsweise "language tab"
74         clickTab(tabName);

```

Abbildung 5: Codeausschnitt Transkript Spracheingabe

Wird von der API ein Ergebnis der Spracheingabe empfangen, so landet man in der Zeile 55 der Implementierung. Als Transkript wird die Spracheingabe, in Zeile 56, in Kleinbuchstaben konvertiert, um eine einheitliche Verarbeitung durchführen zu können. Dies hängt damit zusammen, dass folglich in der Funktion Sprachbefehle definiert werden. Um hier Fehlern vorzubeugen, welche ihren Ursprung in der Rechtschreibung finden könnten, sollen die Befehle einheitlich in Kleinbuchstaben definiert werden. Die Zeilen 59 bis 61 sorgen dafür, dass in einem Selektionsfeld des Datums ein entsprechendes Datum ausgewählt werden kann. Darauf folgen Sprachbefehle für Funktionen wie das Speichern des Elements oder das Klicken auf den Button, wodurch sich die Seite anzeigen lässt in Zeilen 70 bis 71. Durch den Code der Zeilen 72 bis 74 wird eine dynamische Funktion eingebunden, die auf entsprechende Menüpunkte klickt. Befindet man sich beispielsweise im Seitenmodul vom TYPO3 Backend und bearbeitet dort ein Textelement, so sind entnommen der folgenden Abbildung 6 weitere Unterpunkte auswählbar.

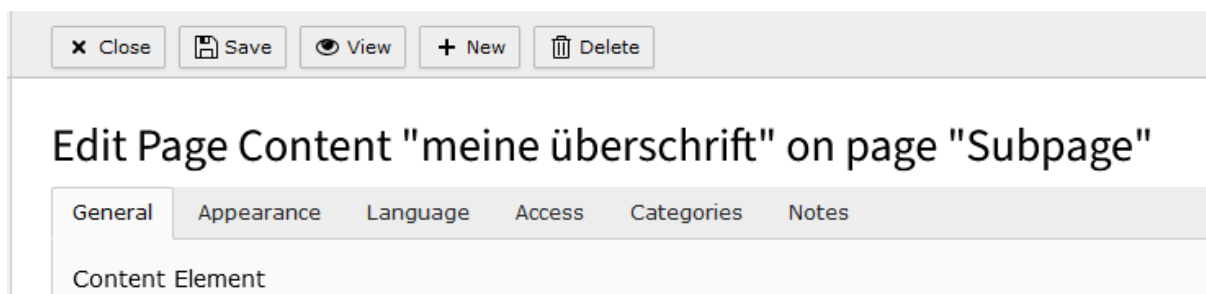


Abbildung 6: TYPO3 Menüband

Die Sprache des Backends ist auf Englisch eingestellt, wodurch dort Abschnitte wie „Appearance“ zur Verfügung stehen. Wird im Sprachbefehl nun von dem Wort „Tab“, gefolgt von einem der Menüpunkte, Gebrauch gemacht, so klickt die Funktion `clickTab()` in Zeile 96 darauf und wechselt die Ansicht entsprechend.

```

105     } else if (transcript.includes('eingabe') || transcript.includes('gebe ein')) {
106         const inputText = transcript.replace(/.*(eingabe|gebe ein)\s*/, '').trim();
107         const focusedElement = document.activeElement;
108         if (focusedElement) {
109             setInputToField(focusedElement, inputText);
110         } else {
111             console.error("No field selected to input text.");
112         }
113     }

```

Abbildung 7: Codeausschnitt Texteingabe

Enthält die Spracheingabe das Wort „Eingabe“ oder die Wortfolge „gebe ein“, so wird dies von dem Transkript getrennt, wodurch lediglich folgende Worte in Textfeldern eingegeben werden können. Daraufhin wird in Zeile 109 eine Funktion ausgeführt, die eine Eingabe des Textes in ein Inhaltselement durchführt, vorausgesetzt es ist ein valides Eingabefeld vorab selektiert worden.

Sprachassistenten sind nicht immer akkurat in ihrer Spracheingabe und -verarbeitung. So ist es auch der Fall bei der Implementierung der Web Speech API. Um Fehlern vorzubeugen, wäre es möglich, durch extensives Testen der Software, definierte Sprachbefehle zu erweitern. Dadurch könnten erwartete Fehler bereits durch leicht abweichende Eingaben aufgefangen und entsprechend korrigiert werden.

```

129     // Generic Methode zum Klicken auf einen Tab basierend auf dem gegebenen Inhalt
130     Show usages   ↳ Michael H
131     function clickTab(tabName) :void {
132         var tabLinks :NodeListOf<Element> = document.querySelectorAll( selectors: 'a.nav-link');
133         var foundTab :boolean = false;
134
135         tabLinks.forEach( callbackfn: function(tab :Element ) :void {
136             var tabText :string = tab.textContent.trim().toLowerCase();
137             if (tabText === tabName.toLowerCase()) {
138                 tab.click();
139                 console.log("Clicked " + tabName + " tab.");
140                 foundTab = true;
141             }
142         });
143         if (!foundTab) {
144             console.error(tabName + " tab not found.");
145         }

```

Abbildung 8: Codeausschnitt clickTab()

In der oben angegebenen Abbildung 8 wird die bereits erwähnte `clickTab()` Methode ausimplementiert dargestellt. Die Funktion nimmt den Namen des Menüpunkts aus der Spracheingabe als Parameter an. Da diese Menüpunkte in der Weboberfläche des TYPO3 Backends als „`a.nav-link`“, wie in Zeile 131 zu sehen, gekennzeichnet sind, werden alle gefundenen Ergebnisse in einer Variable zwischengespeichert. In der Zeile 134 beginnt eine `forEach` Schleife, welche die zuvor zwischengespeicherten Menüpunkte durchläuft und schaut, ob die Spracheingabe dort inkludiert ist. Hierfür wird in Zeile 135 der zwischengespeicherte Inhalt ebenfalls in Kleinbuchstaben umgewandelt, damit die Eingabe und der gefundene Menüpunkt exakt miteinander in Zeile 136 abgeglichen werden können. Stimmen beide Variablen miteinander ein, so wird die `click()` Funktion ausgeführt, wodurch die Ansicht entsprechend gewechselt wird.

### 5.5.2 PHP

Damit es möglich ist, eine implementierte API zur Sprachsteuerung in die Weboberfläche des Backends einzubinden, sollte dies durch PHP geschehen. Da das Seitenmodul von TYPO3 eine Vielfalt an Aufgaben von Redakteuren und Redakteurinnen beinhaltet, könnte hierfür die zur Verfügung gestellte API `ModifyPageLayoutContentEvent` [32] verwendet werden. Somit könnte unter anderem ein Button zu dem Seitenmodul hinzugefügt werden, welcher eine implementierte JavaScript Datei durch Knopfdruck ausführt.

Da Redakteurstätigkeiten jedoch tiefer in das Backend reichen als nur das Seitenmodul, sollte es breiter gefächert betrachtet werden. [33] Die von TYPO3 zur Verfügung gestellte API `ModifyButtonBarEvent` bietet die Möglichkeit, Button beliebig an Kopfzeilen von Modulen hinzuzufügen. Es soll sich bei der Erweiterung um einen Prototypen handeln, weshalb hier nicht nach Modulen eingegrenzt wird, welche von Personen mit der Redakteursrolle im Content-Management-System verwendet werden. Diese Einschränkung wäre gegeben, jedoch ist es für den Fall, eine Sprachsteuerung lediglich hinsichtlich der Möglichkeit zu analysieren, nicht zwingend notwendig.

```

14 class PageLayoutListener
15 {
    no usages  ± Michael H
16 public function __invoke(ModifyButtonBarEvent $event): void
17 {
18     $buttons = $event->getButtons();
19     $buttonBar = $event->getButtonBar();
20     $iconFactory = GeneralUtility::makeInstance(IconFactory::class);
21
22     // Button 1 -> "normaler" button
23     $speechButton = $buttonBar->makeInputButton()
24         ->setName('speech-action')
25         ->setTitle('Speech Input')
26         ->setIcon($iconFactory->getIcon('ext-speechinput-action', Icon::SIZE_SMALL))
27         ->setShowLabelText(true)
28         ->setClasses('speech-action');
29
30     // Position des buttons
31     $buttons[ButtonBar::BUTTON_POSITION_LEFT][3][] = $speechButton;
32
33     // Button 1 -> toggle button
34     $toggleButton = $buttonBar->makeInputButton()
35         ->setName('toggle-action')
36         ->setTitle('Toggle Speech Input')
37         ->setIcon($iconFactory->getIcon('ext-speechinput-action', Icon::SIZE_SMALL))
38         ->setShowLabelText(true)
39         ->setClasses('toggle-action');

```

Abbildung 9: Codeausschnitt Button für die Aktivierung

Demnach ist zu sehen, dass `ModifyButtonBarEvent` in der Zeile 16 des Codes eingebunden wird. In den Zeilen 18 bis 20 werden Variablen für den späteren Gebrauch initialisiert. Folglich wird in den Zeilen 23 bis 31 der erste Button implementiert. Zu erkennen ist, dass Methoden wie `makeInputButton` in Zeile 23 verwendet werden. Diese sind von TYPO3 gegeben, in diesem Falle konkret von der `ButtonBar` Komponente. [34] Zu jedem Button werden einige Werte wie der Titel und das zu verwendende Icon festgelegt. Insbesondere die Klasse wird in den Zeilen 28 und 39 für die jeweiligen Button definiert. Dadurch ist es möglich, auf Ereignisse im JavaScript Code zu hören und entsprechende Funktionen einzuleiten.

```

44     // Buttons aktualisieren
45     $event->setButtons($buttons);
46
47     // CSS und JS für die Buttons hinzufügen
48     $pageRenderer = GeneralUtility::makeInstance(PageRenderer::class);
49     $pageRenderer->loadRequireJsModule('TYPO3\CMS/SpeechExtManual/SpeechInput');
50     $pageRenderer->addCssFile('EXT:speech_ext_manual/Resources/Public/Styles/Styles.css');

```

Abbildung 10: Codeausschnitt Button einfügen

Abschließend werden die Button zu der Leiste hinzugefügt. Zudem wird in der Zeile 49 festgelegt, welche JavaScript Datei eingebunden werden soll.

## 5.6 Funktionstests

Funktionstests sind ein zentraler Bestandteil der Qualitätssicherung in der Softwareentwicklung und dienen dazu, die korrekte Umsetzung und Funktionalitäten einer Anwendung zu überprüfen. Hinsichtlich der zu entwickelnden TYPO3 Erweiterung für Sprachsteuerung werden diese Tests durchgeführt, um sicherzustellen, dass alle implementierten Funktionen zuverlässig sind. Hierfür werden die Funktionstests von dem/der Entwickler:in selbst durchgeführt, um die Erweiterung im Bezug auf mögliche Sprachsteuerungsszenarien gezielt zu testen.

Nach ISO/IEC/IEEE 29119-4 [35] werden Funktionstests genutzt, um durch präzise Testfälle sicherzustellen, dass die Anforderungen eines Systems erfüllt werden. Dabei basieren die Tests auf sogenannten Testmodellen, die testbare Aspekte eines Systems definieren, wie etwa Funktionen. Das Ziel ist also, durch gezielte Eingaben und erwartete Ausgaben nachzuweisen, dass ein System korrekt funktioniert oder mögliche Fehler darzustellen. Im Falle der Extension ist insbesondere die Äquivalenzpartitionierung nach ISO/IEC/IEEE 29119-4 von Bedeutung. Diese wird wie folgt definiert:

*„Equivalence partitioning (BS 7925-2:1998; Myers 1979) uses a model of the test item that partitions the inputs and outputs of the test item into equivalence partitions (also called 'partitions' or 'equivalence classes'), where each equivalence partition shall be defined as a test condition.“ [35, p. 8]*

Diese Methode hilft, Eingaben und Ausgaben in Gruppen zu unterteilen, die ähnlich behandelt werden, wodurch die Effizienz der Tests gesteigert und eine ausreichende Testabdeckung gewährleistet wird.

Die Funktionstests werden also durchgeführt, um zu überprüfen, ob die implementierten Sprachbefehle korrekt erkannt und die entsprechenden Funktionen im TYPO3 Backend ausgeführt werden. Dabei soll die Integration der Erweiterung in ein neu aufgesetztes TYPO3 System mit der Version 12.4.19 getestet werden. Zusätzlich soll die Erweiterung in der Fehlerbehandlung getestet werden, also der Frage, wie das System auf ungültige oder nicht erkannte Sprachbefehle reagiert. Schlussendlich wird auch die Kompatibilität zunächst mit dem primären Browser, Google Chrome, als auch mit Mozilla Firefox überprüft. Im Bezug zur Funktionsweise soll ein einfach dargestellter Arbeitsfluss getestet werden. Dies beinhaltet das Erstellen eines neuen Inhaltelementes, das Einfügen von neuem Text in diversen Feldern und die Navigation innerhalb des neu erstellten Objekts. Daraufhin soll getestet werden, wie

bereits bestehende Inhaltselemente über Sprachsteuerung bearbeitet werden können. Letztlich soll differenziert zu Textelementen getestet werden, ob neue Medien wie Bilder angelegt werden und ob diese eingebunden werden können.

Schlussendlich soll überprüft werden, ob die Erweiterung der vorab durchgeführten Anforderungsanalyse gerecht wird. Da es sich nicht um eine Vollversion einer Erweiterung handelt, sondern lediglich um einen Prototypen, sollen die Funktionstests als Verdeutlichung dort hingegen dienen, zu welchem Grad ein Arbeitsfluss von Redakteuren und Redakteurinnen mit Hilfe von Sprachsteuerung abgedeckt werden kann.

### **5.6.1 Definition der Äquivalenzklassen**

Um die Funktionstests systematisch durchführen zu können, werden mögliche Eingaben und Szenarien der Sprachsteuerung in Äquivalenzklassen unterteilt. Die zuvor definierte Methode der Äquivalenzpartitionierung ermöglicht es, Testfälle abzuleiten, in denen ähnliche Eingaben in Gruppen zusammengefasst werden. Jede dieser Äquivalenzklassen repräsentiert dabei eine Kategorie von Eingaben, die vom System in ähnlicher Art und Weise behandelt werden sollen. Für die TYPO3 Erweiterung wurden folgende Äquivalenzklassen definiert:

1. Valide Sprachbefehle: Korrekte und unterstützte Eingaben, die gewünschte Aktionen auslösen.  
Erwartetes Ergebnis: Aktionen werden korrekt ausgeführt.
2. Ungültige Sprachbefehle: Befehle, die nicht unterstützt werden.  
Erwartetes Ergebnis: Das System ignoriert den Befehl.
3. Randfälle: Sonderfälle wie aneinander gereihete Befehle in einer Eingabe.  
Erwartetes Ergebnis: Das System verarbeitet den Befehl nur teilweise.
4. Fehlerhafte Eingaben: Leere Sprachbefehle oder Hintergrundgeräusche.  
Erwartetes Ergebnis: Das System erkennt keinen Befehl.

Diese Äquivalenzklassen bieten eine Grundlage für die Testfälle, indem die unterschiedlichen Eingabekategorien abgedeckt werden.

### **5.6.2 Abgeleitete Testfälle**

Basierend auf den definierten Äquivalenzklassen wurden konkrete Testfälle erstellt, um die Funktionalität der Sprachsteuerung grundlegend zu prüfen. Diese Testfälle überprüfen, ob die implementierten Funktionen korrekt ausgeführt werden und wie das System auf die verschiedenen Eingabekategorien reagiert.

1. Testfall: Gültiger Sprachbefehl – Erstellen eines neuen Inhaltselements

Eingabe: „Erstelle ein neues Inhaltselement“.

Erwartetes Ergebnis: Das Erstellen eines neuen Inhaltselements wird eingeleitet, woraufhin das Selektionsbild für die Art des Inhaltes angezeigt wird.

2. Testfall: Ungültiger Sprachbefehl – Nicht unterstützte Funktion.

Eingabe: „Lösche alle Inhalte“.

Erwartetes Ergebnis: Das System nimmt den Befehl entgegen, es wird jedoch keine Aktion ausgeführt.

3. Testfall: Randfall – Aneinandergereihte Befehle.

Eingabe: „Select Header mit der Eingabe meine Überschrift“.

Erwartetes Ergebnis: Das System verarbeitet lediglich einen Teil der Eingabe. In diesem Fall, dass die Kopfzeile selektiert werden soll.

4. Testfall: Fehlerhafte Eingabe – Leerer Sprachbefehl.

Eingabe: Laute Hintergrundgeräusche wie Tastatur, Maus, berühren des Mikrofons.

Erwartetes Ergebnis: Das System erkennt keine Eingabe.

Dadurch, dass diese speziellen Testfälle durchgeführt werden, wird verdeutlicht, ob klar definierte Anweisungen die entsprechenden Ergebnisse liefern. Hierdurch soll lediglich die grundlegende Funktionalität des Codes geprüft werden.

### **5.6.3 Testfall: Arbeitsfluss**

Nachdem die Erweiterung hinsichtlich der Integration getestet wurde, soll in diesem Schritt ein vereinfachter Arbeitsfluss getestet werden. Ziel ist es, zentrale Redakteurstätigkeiten im Zusammenhang mit der Erstellung von Inhaltselementen in Form von Text und Medien darzustellen. Darüber hinaus soll getestet werden, inwiefern die Bearbeitung bereits bestehender Elemente durch die Sprachsteuerung unterstützt wird.

Hierfür soll unter anderem das Erstellen eines neuen Inhaltselements durch Sprachsteuerung getestet werden. Daraufhin wird die Auswahl des Inhaltstyps geprüft. In der Weboberfläche des TYPO3 Backends, in der das neu angelegte Seitenelement bearbeitet wird, sollen nun einige Textfelder mit Inhalt befüllt werden. Ebenfalls soll die Navigation innerhalb dieser Ansicht getestet werden, indem diverse Untermenüpunkte und Felder durch Sprache selektiert werden. Weiterhin soll das Hinzufügen von Medien in Form eines Bildes geprüft werden. Das neu angelegte Inhaltselement soll durch einen Sprachbefehl gespeichert und abschließend angezeigt werden.

Dieser Testfall überprüft, ob die Sprachsteuerung grundlegende Redakteurstätigkeiten abdecken kann. Das Testen eines vereinfachten Arbeitsflusses soll also dazu dienen, die Erweiterung umfassend daraufhin zu evaluieren, welche Arbeitsschritte nahtlos anhand von Sprach-

befehlen durchgeführt werden können. Dabei soll insbesondere vorgeführt werden, welche Möglichkeiten die spezifische Integration bieten kann.

## **5.7 Nutzbarkeitstests**

Die Nutzbarkeitstests der TYPO3 Erweiterung dienen dazu, die Benutzerfreundlichkeit und Praxistauglichkeit der Sprachsteuerung zu evaluieren. Demnach werden Aspekte wie die Latenzzeit, Genauigkeit der Eingabe, verfügbare Steuerungsmethoden und die Möglichkeit einer freihändigen Bedienung in Betracht gezogen. Zudem wird die Erweiterung hinsichtlich verschiedener Sprachunterstützungen getestet.

Wie zuvor anhand der Norm ISO 9241-11 definiert, beinhaltet die Nutzbarkeit eines Systems, dass die Ziele der Nutzenden effektiv, effizient und zufriedenstellend erreicht werden können. Im Kontext der Applikation wird die Genauigkeit der Eingabe geprüft, um im Bezug auf Effektivität zu analysieren, ob Sprachbefehle stets präzise erkannt und korrekt ausgeführt werden. Die Latenzzeit wirkt sich auf die Effizienz des Systems aus, weshalb erkannt werden soll, in welchem Rahmen sich diese befindet. Diverse Steuerungsmöglichkeiten und die eventuelle freihändige Bedienung würden die Nutzererfahrung beeinflussen, wodurch der Aspekt der Zufriedenheit geprüft wird.

Da es sich bei der Erweiterung nicht um eine marktreife Implementierung handelt, sollen diese Tests zentral dazu dienen, vorzuzeigen, welche aktuellen Gegebenheiten die Nutzbarkeit in welchem Maß beeinflussen. Aspekte wie die Genauigkeit der Eingabe, sowie die Latenzzeit zeigen, ob die gewählte Art an Implementierung für ein Content-Management-System wie TYPO3 geeignet sein könnte. Weiterhin werden bereits implementierte Steuerungsmethoden analysiert, um einen Einblick darin zu bieten, wie eine zukünftige Version der Erweiterung bedient werden könnte und ob die gewählten Methoden bereits ausreichend sein könnten.

### **5.7.1 Genauigkeit der Eingabe**

Die Genauigkeit der Eingabe ist ein zentraler Bestandteil der Nutzbarkeitstests und steht im direkten Zusammenhang mit der Effektivität des Systems. Sprachbefehle sollten zuverlässig angenommen und entsprechend verarbeitet werden, wodurch gezielte Anweisungen ausgeführt werden.

Hierfür soll Stichprobenartig getestet werden, wie präzise Sprachbefehle vom System erkannt werden. Dabei werden Szenarien wie der Versuch einer klaren Aussprache, einer etwas undeutlicheren Aussprache oder Hintergrundgeräusche berücksichtigt. Das Ziel ist es, zu prüfen, ob eine hohe Genauigkeit stets gegeben ist.

### **5.7.2 Latenzzeit**

Im Aspekt der Latenzzeit soll gemessen werden, wie schnell das System auf Sprachbefehle reagiert und die entsprechenden Anweisungen ausführt. Eine niedrige Latenzzeit ist essenziell, um den Arbeitsfluss nicht großartig im Bezug auf die Effizienz einzuschränken. Ziel ist es also, zu messen, welche Zeitspannen gegeben sind, zwischen Eingabe und Verarbeitung dessen vom System.

### **5.7.3 Verfügbare Steuerungsmethoden**

Die Steuerungsmethoden beeinflussen die Nutzererfahrungen. Verschiedene Optionen wie die Aktivierung der Sprachsteuerung anhand von Button innerhalb der Weboberfläche, durch Toggle oder durch herkömmliche Aktivierungsworte werden evaluiert, um zu prüfen, ob individuelle Bedürfnisse eingebunden werden können. Aktivierungsworte, wie sie in Sprachassistenten wie Google Assistant oder Amazon Alexa vorzufinden sind, würden eine kontinuierliche Eingabe und somit eine freihändige Bedienung ermöglichen, welche besonders hinsichtlich der Unterstützung von Barrierefreiheit von Bedeutung ist.

### **5.7.4 Sprachunterstützung**

Ein zentraler Aspekt der Tests ist die Überprüfung der Sprachunterstützung in unterschiedlichen Konfigurationen. Es wird getestet, ob die Sprachsteuerung korrekt funktioniert, wenn das Backend auf Englisch eingestellt ist, während die Sprachbefehle auf Deutsch definiert sind. Ebenso wird geprüft, ob das System auf englische Eingaben reagieren kann, wenn die Sprachsteuerung auf Deutsch konfiguriert ist. Ziel ist es, die Flexibilität der Sprachunterstützung zu bewerten.

## **6 Ergebnisse der Implementierungen und Tests**

### **6.1 Implementierung Google Home**

Im Verlauf der Analyse wurde schnell deutlich, dass Google seit einigen Änderungen keine Unterstützung mehr für Drittanbieterintegrationen bietet. Wie bereits erwähnt wurden die zuvor genutzten „Conversational Actions“ am 13. Juni 2023 abgeschaltet und durch neue, jedoch auf Android Applikationen fokussierte Elemente, ersetzt.

Dies stellt ein großes Hindernis dar, wodurch von einer Implementierung von Google Assistant abgesehen wird.

## 6.2 Implementierung Alexa

Die Implementierung von Amazon Alexa als Sprachsteuerung für TYPO3 erwies sich als technisch anspruchsvoll und ist mit mehreren Herausforderungen verbunden. Zunächst erforderte die Entwicklung die Erstellung spezifischer REST-API-Endpunkte, welche als Schnittstelle für Sprachbefehle dienen. Diese Endpunkte sollten über das Routing-System von TYPO3 definiert werden und die eingehenden Anfragen mit Controller-Methoden verbinden. [30] Das Routing erfolgt über eine separate `Routes.php` Datei innerhalb der Extension. Hier werden spezifische URL-Pfade definiert, die den entsprechenden Controller-Methoden zugeordnet sind.

Ein konkretes Routing könnte definiert werden, indem ein Pfad wie `/alexa/createContent` festgelegt wird, welcher bei einer POST-Anfrage auf eine Methode verweist, welche ein Inhaltselement anlegt. Im Anschluss erfolgte eine Registrierung in der Alexa Developer Console. Hier wurde testweise ein neuer Intent zum Erstellen einer Seite angelegt.

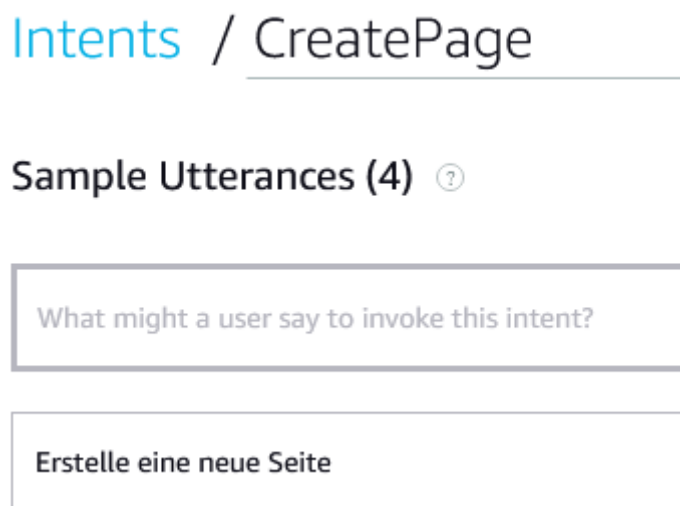


Abbildung 11: Alexa Developer Console

Um diesem eine direkte Anweisung an das Backend des CMS zuzuweisen, wurde ein simpler Code geschrieben.

```
140 ▾ const CreatePageIntentHandler = {
141 ▾   canHandle(handlerInput) {
142     return Alexa.getRequestType(handlerInput.requestEnvelope) === 'IntentRequest'
143     && Alexa.getIntentName(handlerInput.requestEnvelope) === 'CreatePage';
144   },
145 ▾   handle(handlerInput) {
146     return new Promise((resolve, reject) => {
147       const postData = JSON.stringify({ title: 'Neue Seite', content: 'Inhalt der neuen Seite' });
148       const options = {
149         hostname: 'bachelor.902147.jweiland-hosting.de',
150         path: '/alexa/createPage',
151         method: 'POST',
152         headers: {
153           'Content-Type': 'application/json',
154           'Authorization': 'Bearer Rm9rcsN3ZndiQXBpS2V5VGZzdDIwMjM='
155         }
156       };
157     });
158   }
159 }
```

Abbildung 12: Codeausschnitt Alexa

Hier ist zu sehen, dass in Zeile 149 die grundlegende URL des Systems angegeben wird. Folglich wird in Zeile 150 ein Pfad zu einem einzubindenden API-Endpunkt definiert. Dabei ist schnell aufgefallen, dass eine Form von Authentifizierung erforderlich ist. Wäre dies nicht gegeben, könnte es sich als Sicherheitslücke offenbaren. Dementsprechend wurde in diesem Falle testweise in einem String in Zeile 154 ein Token festgelegt.

Dieser Token müsste von der Erweiterung im TYPO3 Backend verifiziert werden. Dies bedeutet, dass eine Token-Generierung bereitgestellt werden müsste. Würde ein weitreichender Alexa Skill erstellt werden, welcher alle möglichen Intents zu Redakteurstätigkeiten enthält, müsste dieser folglich von Amazon geprüft werden, um im öffentlichen Alexa Skill Store für Nutzende zur Verfügung zu stehen. Hier müsste eine Konfiguration eingerichtet werden, in der sowohl ein Token zur Authentifizierung, als auch die anzusprechende URL des CMS eingegeben werden kann.

Bereits an dieser Stelle wurde der weitere Verlauf der Entwicklung abgebrochen. Dies hängt damit zusammen, dass neben Bedenken hinsichtlich der Datensicherheit bei der Übertragung von Daten auch weitere Probleme bestanden. Würde diese Art an Implementierung fortgeführt werden, so wäre eine Hürde der Echtzeitverarbeitung anstehen. Durch Sprachbefehle werden Anweisungen an Endpunkte gesendet, welche Aktionen im Backend ausführen. Dies würde jedoch nicht die Weboberfläche für Nutzende aktualisieren. Es trägt zur Folge, dass wenn beispielsweise ein Textfeld überarbeitet werden soll, dies im Backend ausgeführt, gespeichert und erst nach einem aktualisieren der Seite angezeigt wird. Sprachassistenten und Spracheingabesysteme sind nicht immer akkurat. Demnach müsste ein hoher Grad an Vertrauen an ein etwas instabiles System verlangt werden.

Weiterhin gäbe es einige Komplikationen im Bezug auf der Anforderungsanalyse, in der verdeutlicht wurde, dass kein extensives technisches Hintergrundwissen verlangt werden sollte. Verglichen mit Alternativen wie die folgende Integration mit JavaScript und der Web Speech

API, könnte es ein Hindernis darstellen. Ebenso würde bei dieser Implementation ein sehr spezifisches Zusatzgerät vorausgesetzt werden, in Form eines Amazon Alexa Lautsprechers.

### 6.3 Implementierung Web Speech API

Die Implementierung der Web Speech API zur Sprachsteuerung von TYPO3 verlief insgesamt erfolgreich und zeigte die Vorteile einer browserbasierten Lösung ohne Verwendung externer Geräte. [34] Durch bestehende TYPO3-Endpunkte und -Komponenten, wie zum Verändern der Button innerhalb von Kopfzeilen, konnte die JavaScript Anwendung gut an die Weboberfläche angebunden werden.

Diese vorhandene Struktur reduzierte den Entwicklungsaufwand, da klare Schnittstellen und Funktionen zur Verfügung stehen. Die umfangreiche Dokumentation seitens TYPO3 lieferte hierbei einen großen Mehrwert. Insgesamt leistet das TYPO3-Team in diesem Aspekt hervorragende Arbeit, die es Entwicklern und Entwicklerinnen ermöglicht, schnell Lösungswege zu finden.

Ebenso verlief die Integration der Web Speech API in JavaScript grundsätzlich bis auf einige Aspekte reibungslos. Die API bietet die Möglichkeit, in einfachen Schritten ein Grundgerüst aufzubauen. In nur wenigen Zeilen Code konnten Spracheingaben erfasst werden, welche in weiteren Funktionen verarbeitet werden konnten.

Jedoch mussten einige Lösungswege entwickelt werden, um spezifischen Anforderungen gerecht zu werden. Unter anderem die Implementierung einer Toggle-Funktion. Zwar konnte hierfür ein Button im PHP-Teil des Projekts erstellt werden, jedoch bietet die API nicht die Funktion, einen Sprachkanal kontinuierlich geöffnet zu lassen. Um dies zu umgehen, wurde die Lösung angewendet, welche den Sprachkanal nach dem Schließen automatisch wieder öffnet, bis der Toggle aufgehoben oder die Seite neu geladen wird.

Weitere Hürden bestanden zu Beginn in der Extrahierung von Sprachbefehlen aus der Spracheingabe. Dies hing damit zusammen, dass es vorkommen kann, dass Nutzende während einer Konversation mit anderen Personen einen Befehl ausführen wollen. Da anfangs exakte Eingaben erwartet wurden, musste dies abgeändert werden. Die Integration von der `includes` Methode wurde in diesem Falle als Lösung eingebunden. Ein beständiges Problem stellt die Navigation und Ausführung von Sprachbefehlen innerhalb der Modalfenster dar. Sie werden dynamisch aufgerufen, ohne, dass die Weboberfläche neu geladen wird, wodurch der JavaScript Code keinen Zugriff auf die entsprechenden Elemente besitzt.

Dennoch konnte eine gute Grundlage aufgebaut werden, in der Sprachbefehle problemlos konfiguriert und hinzugefügt werden können. Ein extensives Testen der Erweiterung könnte

die Zuverlässigkeit erhöhen, indem Randfälle aufgedeckt und in den Sprachbefehlen inkludiert werden könnten.

## 6.4 Ergebnisse Funktionstests

Die Ergebnisse der zuvor definierten Testfälle sollen in diesem Kapitel dargestellt werden. Zunächst werden die Testergebnisse der durchgeführten abgeleiteten Fälle vorgezeigt werden, welche die grundlegenden Funktionen in ihrer Funktionsweise überprüfen. Daraufhin wird ein vereinfacht dargestellter Arbeitsfluss durchgeführt, wodurch eventuelle Lücken der Implementierung aufgedeckt werden sollen.

### 6.4.1 Grundlegende Funktionen

Die durchgeführten Funktionstests basieren auf den vier abgeleiteten Testfällen und zielen darauf ab, die grundlegende Funktionsfähigkeit der Sprachsteuerung zu überprüfen.

Der erste Testfall überprüfte, ob das System auf einen gültigen Sprachbefehl zur Erstellung eines neuen Inhaltselements korrekt reagiert. Der Sprachbefehl „Erstelle ein neues Inhaltselement“ wurde erfolgreich erkannt und das System verarbeitete es entsprechend. Die Web-Oberfläche zeigte wie erwartet das Modalfenster zur Selektion der Art des Inhalts an. Dieses Ergebnis bestätigt, dass Funktionen des JavaScript Codes wie gewünscht ausgeführt werden können.

Ein weiterer Testfall prüfte, wie das System auf einen ungültigen Sprachbefehl reagiert. So wurde bei dem Befehl „Lösche alle Inhalte“ die Eingabe vom System erfasst, jedoch wurde keine Aktion ausgeführt. Dieses Verhalten entspricht den Erwartungen und zeigt, dass nicht unterstützte Befehle keine unerwünschten Änderungen vornehmen.

Zusätzlich wurde ein Randfall getestet, bei dem ein komplexerer, aneinandergereihter Sprachbefehl als Eingabe diente. Der Befehl „Select Header mit der Eingabe meine Überschrift“ wurde nur teilweise verarbeitet. Das System erkannte die Anweisung, die Kopfzeile zu selektieren, ignorierte jedoch den zweiten Teil der Eingabe. Dies hängt damit zusammen, dass im `If`-Statement des JavaScript Codes, welche die Spracheingabe einer Funktion zuweist, den Befehl „eingabe“ oder „gebe ein“ erst zum Schluss prüft. Also wird die Bedingung vorab durch Anweisung der Selektion des spezifischen Feldes erfüllt. Dieser Testfall verdeutlicht, dass einfache Befehle zuverlässig umgesetzt werden können, jedoch bei komplexeren Anweisungen, wenn dessen Implementierung gewünscht ist, noch Optimierungsbedarf besteht.

Schließlich wurde im vierten Testfall untersucht, wie das System auf fehlerhafte Eingaben reagiert. Bei Hintergrundgeräuschen wie Tastatureinschläge oder Berührungen des Mikrofons zeigte das System keine Reaktion. Es wurden also keine gültigen Sprachbefehle erkannt. Die-

ses Verhalten entsprach den Erwartungen und zeigt, dass durch solch Störungen keine unnötigen Aktionen durchgeführt werden.

Die Ergebnisse der grundlegenden Funktionstests zeigen also, dass die TYPO3 Erweiterung in der Lage ist, Sprachbefehle in einfachen Szenarien zu verarbeiten und umzusetzen. Während die Implementierung für gültige und ungültige Eingaben gut funktionierte, erfordern aneinandergereihte Befehle in einer einzelnen Spracheingabe noch Optimierung. Es zeigt, dass eine solide Basis für einen weiteren Ausbau zur Verfügung steht.

#### **6.4.2 Sprachsteuerung im Arbeitsfluss**

Um zu bewerten, ob die Sprachsteuerung für typische Redakteurstätigkeiten geeignet sein könnte, wurde ein vollständiger Workflow getestet, der die Erstellung, Bearbeitung und Ergänzung von Inhaltselementen beinhaltet. Ziel war es, festzustellen, wie nahtlos die Sprachsteuerung einzelne Schritte unterstützt und welche Herausforderungen dabei vorkommen.

Da es sich bei der Erweiterung bewusst um einen Prototypen handelt, wurden spezifische Sprachbefehle hinzugefügt, um einen vereinfachten Workflow darzustellen. Diese Befehle ermöglichen eine Navigation innerhalb der Weboberfläche, sowie grundlegende Eingaben in Textfeldern.

Der erste Schritt, das Erstellen eines neuen Inhaltselements verlief reibungslos. Der Sprachbefehl wurde korrekt erkannt und das System öffnete wie erwartet das Modalfenster zur Auswahl des Elementtypen. Jedoch stellte sich hier heraus, dass die Auswahl dessen nicht erfolgreich per Spracheingabe durchgeführt werden konnte. Es scheint, dass der JavaScript Code in der aktuellen Verfassung keinen Zugriff auf die dynamisch geladenen Fenster erlangen kann. Demnach musste die Selektion manuell erfolgen.

Ein neues Element wurde erstellt und es wurde versucht, Textfelder auszuwählen, zwischen den Menüpunkten zu wechseln und sowohl Textfelder, als auch Datumswähler zu befüllen. Dies erwies sich als erfolgreich. Eingaben in Form von Texten konnten getätigt werden, wobei diese lediglich in Kleinbuchstaben erfasst werden. Diverse Felder konnten selektiert werden und Menüpunkte konnten ausgewählt werden, um die Ansichten zu wechseln. Weiterhin war es möglich, das Datum beispielsweise mit der Eingabe „heute“ zu befüllen, wodurch dies in das aktuelle Datum konvertiert wurde.

Ein Problem trat jedoch beim Bearbeiten bereits vorhandener Texte auf. Statt bestehende Texte zu ersetzen oder gezielt zu bearbeiten, wurde der eingegebene Text stets vollständig neu eingefügt, was den Arbeitsfluss in der Inhaltsbearbeitung unterbrechen würde. Dieses Verhalten deutet daraufhin, dass in der Verarbeitung von Textbearbeitungsbefehlen eine differenzierte Funktionsweise implementiert werden muss.

Die Integration von Medien stellte eine weitere Hürde dar. Während das Hochladen einer neuen Datei per Sprachsteuerung eventuell möglich erschien, da auf einen Button geklickt werden könnte, wurde schnell ersichtlich, dass es vermutlich keine Option gäbe. Geöffnet wird die Suche von Dateien des Betriebssystems, worauf das TYPO3 System keinen Zugriff erlangen kann. Ebenso stellte sich das Hinzufügen von Medien basierend auf bestehenden Dateien innerhalb des TYPO3 Backends als Hürde dar. Dies hängt erneut mit Modalfenstern zusammen. Da es dem Prototypen nicht gelangt, Aktionen innerhalb eines Modalfensters auszuführen, fällt diese Funktion vergeblich aus.

Die Ergebnisse der Tests zum Arbeitsfluss zeigen, dass die Sprachsteuerung grundlegende Aufgaben wie das Erstellen und Befüllen von Inhaltselementen sowie die Navigation innerhalb der Weboberfläche gut unterstützt. Jedoch bleiben die Modalfenster und externe Dialoge wie Fenster zur Dateiauswahl ein Hindernis. Würde dies durch weitere Ausarbeitung überwunden werden, so könnte der vereinfachte Workflow ununterbrochen durchgeführt werden. Die Textbearbeitung innerhalb von Editoren bedarf ebenfalls einer durchdachten Überarbeitung. Der Prototyp bietet allerdings eine solide Grundlage, um die Machbarkeit einer sprachgesteuerten Interaktion in TYPO3 vorzuzeigen und könnte als Basis für zukünftige Integration dienen.

## **6.5 Ergebnisse Nutzbarkeitstests**

Obwohl der zentrale Fokus der Arbeit auf der Frage nach der Machbarkeit der Integration einer Sprachsteuerung in das TYPO3-System liegt, wurden auch Tests bezüglich der Nutzbarkeit durchgeführt. Die gewählten Aspekte wurden insbesondere berücksichtigt, da beispielsweise eine lange Latenzzeit oder ungenaue Erkennung grundlegender Sprachbefehle die Einsatzfähigkeit der entwickelten Erweiterung beeinträchtigen könnten.

### **6.5.1 Überprüfung der Genauigkeit**

Bereits im Rahmen der Funktionstests zeigte sich, dass die Genauigkeit der Spracherkennung durch diverse Faktoren variiert. Um diese Gründe genauer zu evaluieren, wurden gezielt verschiedene Mikrofone getestet, darunter ein Webcam-Mikrofon und ein Standmikrofon. Es wurde festgestellt, dass die Erkennungsgenauigkeit trotz unterschiedlicher Mikrofonqualität keine nennenswerten Differenzen aufweisen. Zudem wurde die Aussprache der Befehle bewusst in Variationen durchgeführt. Von deutlich und klar bis hin zu alltäglicher oder sogar ungenauer Aussprache. Bewusst sehr undeutliche Artikulationen führten zu wesentlich häufigeren Fehlern. Es wurden nur halbe Worte erkannt oder es wurden Teile von Worten in einem zusammengeführt. Selbst in bewusst deutlicher Aussprache kam es in einigen Ausführungen zu unerwünschter Spracheingabe. So kam es beispielsweise im Falle des Menüpunktes „General“ gehäuft zu Problemen. Wird die Sprachintegration dem Backend gerecht auf Englisch eingestellt, wurde bei der Eingabe „General Tab“ des Öfteren „generator tab“ erkannt.

Recognized speech: generator tab	<a href="#">SpeechInput.js?bust=...e0728c25f878bb03:57</a>
✖ ▶ generator tab not found.	<a href="#">SpeechInput.js?bust=...0728c25f878bb03:144</a>
Recognized speech: general tab	<a href="#">SpeechInput.js?bust=...e0728c25f878bb03:57</a>
Clicked general tab.	<a href="#">SpeechInput.js?bust=...0728c25f878bb03:139</a>

Abbildung 13: Fehlermeldung Falsche Spracherkennung

Meist wird das gewünschte Ziel durch einen weiteren Versuch mit sehr deutlicher Aussprache erreicht. Dennoch unterbricht dies den Arbeitsfluss.

Ein extensives Testen durch Entwickler:innen oder von Redakteuren und Redakteurinnen würde solch wiederkehrende Hindernisse aufdecken, wodurch diese in die Logik der Erweiterung eingebunden und somit behoben werden könnten.

### 6.5.2 Gemessene Latenzzeit

Die Latenzzeit zwischen der Eingabe eines Sprachbefehls und der Ausführung der entsprechenden Aktion wurde gemessen, indem eine Stoppuhr manuell gestartet wurde, nachdem der gewünschte Befehl ausgesprochen wurde. Die durchschnittliche Dauer von diesem Punkt aus bis hin zur Ausführung der Aktion befand sich in der Regel unter zwei Sekunden.

### 6.5.3 Funktionsweise der Steuerungsmethoden

In dem Prototypen der TYPO3 Erweiterung sind zwei Steuerungsmöglichkeiten gegeben. Zum einen kann ein Button geklickt werden, woraufhin Sprachbefehle angenommen werden, bis für einige Sekunden keine Eingabe erkannt wird. Das automatische Abschalten ist eine Standardfunktion der Web Speech API und beträgt im Falle des verwendeten Google Chrome Browsers mit der Version 131.0.6778.86 in etwa acht Sekunden. Dieser Ansatz wurde gewählt, sodass bei fehlerhaftem Verhalten ein weiterer Befehl unmittelbar ausgesprochen und verarbeitet werden kann. Allerdings kann es als unerwartet empfunden werden, da erwartet werden könnte, dass die Sprachaktivierung nach einem Sprachbefehl beendet wird. Um dies abzuändern, könnte der JavaScript Anteil der Erweiterung angepasst werden, indem der `continuous` Wert auf `false` gesetzt wird.

Der Toggle Button aktivierte die Sprachsteuerung dauerhaft, bis es manuell deaktiviert oder die Seite neu geladen wurde. Diese Methode erwies sich als besonders geeignet für Intensivnutzung und die Navigation der Weboberfläche. Herkömmlich durch Smart Speaker könnten bei konstanter Aktivierung Aktivierungsworte erwartet werden, welche in zukünftigen Iterationen als Implementierungsmöglichkeit festgehalten werden können.

### 6.5.4 Sprachkomplikationen

Im Rahmen der Tests wurde auch das Verhalten der Erweiterung bei unterschiedlichen Sprachkonfigurationen untersucht. Dabei wurde beispielsweise getestet, wie das System rea-

giert, wenn das Backend sowohl von TYPO3, als auch für die Web Speech API auf Englisch eingestellt ist, während die Sprachbefehle auf Deutsch definiert sind. So wurde das Ziel gewählt, das Feld zur Datumeingabe zu selektieren. Hierzu wurde der Befehl „Select Datum“ ausgesprochen, wobei es zahlreiche Anläufe bedarf. In einem vergleichsweise soliden Ablauf musste der Befehl drei Male ausgesprochen werden, bis das gewünschte Ergebnis erreicht wurde.

Recognized speech: select autumn	<a href="#">SpeechInput.js?bust=...e0728c25f878bb03:71</a>
Recognized speech: select home	<a href="#">SpeechInput.js?bust=...e0728c25f878bb03:71</a>
Recognized speech: select datum	<a href="#">SpeechInput.js?bust=...e0728c25f878bb03:71</a>
Focused on date input field.	<a href="#">SpeechInput.js?bust=...0728c25f878bb03:180</a>

**Abbildung 14: Falsche Spracherkennung**

Solche Beobachtungen zeigen, dass für jede unterstützte Sprache spezifische Sprachbefehle definiert werden sollten, um ein möglichst robustes System zu erstellen. Eine mögliche Verbesserung könnte darin bestehen, ein dynamisch ladbares Wörterbuch zu implementieren, welches Befehle für jegliche Sprachen enthält und die korrekte Zuweisung zu entsprechenden Funktionen ermöglicht.

## 7 Schlussfolgerung

Die in dieser Arbeit entwickelte TYPO3 Erweiterung zur Sprachsteuerung zeigt, dass die Frage nach der Machbarkeit mit einem positiven Ausblick beantwortet werden kann. Obwohl die aktuelle Version des Prototyps noch ausbaufähig ist, konnten zentrale Punkte einer Sprachsteuerung erfolgreich umgesetzt und getestet werden.

Mit einem Ausbau der Einbindung von Randfällen in der Konfiguration von Sprachbefehlen kann die Erweiterung zuverlässiger auf Anweisungen reagieren. Bereits jetzt bietet die Erweiterung verschiedene Methoden zur Sprachaktivierung, darunter eine gängige Button Aktivierung, sowie ein Toggle-System. Somit kann die Anwendung nach Bedarf unterschiedlich ausgeführt werden. Ein weiterer Erfolg stellte sich in der Navigation innerhalb der Bearbeitungsansicht von Inhaltselementen heraus. Hierdurch wurde deutlich, dass jegliche Module die Sprachsteuerung einbinden könnten und durch Iterationen des Projekts die Sprache als Steuerungsmöglichkeit weitreichend realisierbar wäre. Zur Vorbereitung dessen wurden die Button zur Sprachaktivierung nicht nur im Seitenmodul, sondern über das Backend verteilt eingebunden, was über gegebene Endpunkte von TYPO3 problemlos möglich war. Damit Redakteurstätigkeiten umfangreich per Sprachanweisungen durchgeführt werden könnten, müssten jedoch weitere Funktionen in die Erweiterung eingebaut werden, so wie die Möglichkeit Tex-

te nicht nur zu ersetzen, sondern auch zu bearbeiten. Ein Hindernis bleibt in der Implementierung von Sprachsteuerung in Modalfenstern bestehen.

## **7.1 Ausblick**

Trotz der erreichten Fortschritte besteht Optimierungspotenzial. Ein zentraler Punkt ist die Integration von Modalfenstern, die aktuell nicht steuerbar sind. Hierzu gibt es jedoch bereits Lösungsansätze aus der TYPO3-Community. [37] So bietet Mogens Fiebrandt in einem Forumsbeitrag Lösungsmöglichkeiten dazu, wie die entsprechenden Modalelemente im TYPO3-Backend angesprochen werden könnten.

Sowohl dieses Forum, als auch die öffentliche Dokumentation von TYPO3 stellen wertvolle Quellen zur Problemlösung schwerwiegender Hindernisse dar. Insbesondere dadurch, dass Experten und Expertinnen sich dazu bereit erklären auf Fragen zu antworten und über diese zu diskutieren. Das bestehende Grundgerüst, vorgezeigte Verbesserungsvorschläge, bereits gegebene Lösungsansätze und die Möglichkeit auf die Implementierung zu iterieren verdeutlichen, dass eine Sprachsteuerung zukünftig als unterstützendes Mittel für Redakteurstätigkeiten eingesetzt werden könnte.

## 8 Danksagung

Zunächst einmal bedanke ich mich Sebastian Kreideweiß, der nicht nur eine große Anzahl an Themenvorschlägen geboten hat, sondern auch diese Arbeit durchweg hervorragend betreut hat.

Ein großes Dankeschön möchte ich auch an die gesamte TYPO3-Community richten, die eine unfassbar extensive Dokumentation des gesamten Systems erstellt hat. An dieser Stelle auch noch einmal einen herzlichen Dank an Mogens Fiebrandt für die zahlreichen Hilfestellungen und Lösungsansätze im t3academy Forum.

Ebenso wäre die Ausarbeitung in dieser Art längst nicht so gut verlaufen, wäre die Bereitstellung einer Hosting-Umgebung von JWeiland nicht gegeben.

Abschließend möchte ich mich bei meiner Freundin bedanken, die mich im gesamten bisherigen Studium stets begleitet hat.

## 9 Literaturverzeichnis

- [1] A. Kochba, „Web Almanac,“ 1 Dezember 2021. [Online]. Available: <https://almanac.httparchive.org/en/2021/cms#what-is-a-cms>. [Zugriff am 8 Oktober 2024].
- [2] TYPO3, „TYPO3,“ [Online]. Available: <https://typo3.org/>. [Zugriff am 10 Oktober 2024].
- [3] TYPO3, „TYPO3 Extension development,“ [Online]. Available: <https://docs.typo3.org/m/typo3/reference-coreapi/main/en-us/ExtensionArchitecture/Index.html#extension-development>. [Zugriff am 10 Oktober 2024].
- [4] TYPO3, „TYPO3 Permissions management,“ [Online]. Available: <https://docs.typo3.org/m/typo3/reference-coreapi/main/en-us/Administration/PermissionsManagement/Index.html>. [Zugriff am 10 Oktober 2024].
- [5] TYPO3, „TYPO3 Backend,“ [Online]. Available: <https://docs.typo3.org/m/typo3/tutorial-getting-started/main/en-us/Concepts/Backend/Index.html#backend>. [Zugriff am 20 10 2024].
- [6] IT-P, „IT-P Sprachassistenten,“ [Online]. Available: <https://www.it-p.de/lexikon/sprachassistenten/>. [Zugriff am 11 Oktober 2024].
- [7] M. Cooper, „The Developer’s Guide to Web Speech API: What It Is, How It Works, and More,“ 7 November 2022. [Online]. Available: <https://dzone.com/articles/the-developers-guide-to-web-speech-api-what-is-it>. [Zugriff am 12 Oktober 2024].
- [8] World Wide Web Consortium (W3C), „W3C,“ [Online]. Available: <https://www.w3.org/>. [Zugriff am 12 Oktober 2024].
- [9] M. Goodwin, „What is an API (application programming interface)?,“ 9 April 2024. [Online]. Available: <https://www.ibm.com/topics/api>. [Zugriff am 12 Oktober 2024].
- [10] Mozilla, „Web Speech API - Speech Recognition,“ 4 Januar 2022. [Online]. Available: [https://wiki.mozilla.org/index.php?title=Web\\_Speech\\_API\\_-\\_Speech\\_Recognition](https://wiki.mozilla.org/index.php?title=Web_Speech_API_-_Speech_Recognition). [Zugriff am 12 Oktober 2024].

- [11] Bundesamt für Justiz, „Gesetze im Internet - Behindertengleichstellungsgesetz,“ 2016. [Online]. Available: <https://www.gesetze-im-internet.de/bgg/>. [Zugriff am 12 Oktober 2024].
- [12] ISO/TC 159/SC 4, „ISO 9241-11:2018(en),“ März 2018. [Online]. Available: <https://www.iso.org/obp/ui/en/#iso:std:iso:9241:-11:ed-2:v1:en>. [Zugriff am 20 Oktober 2024].
- [13] ALEKS & SHANTU, „Modalfenster,“ [Online]. Available: <https://www.aleksundshantu.com/wiki/modalfenster/>. [Zugriff am 18 November 2024].
- [14] Bruns Verlags-GmbH & Co. KG, „TYPO3 Einführung für Redakteure - Grundlagen und Tipps für den Einstieg,“ [Online]. Available: <https://www.p2media.de/news/typo3-einfuehrung-fuer-redakteure-grundlagen-und-tipps-fuer-den-einstieg/>. [Zugriff am 25 Oktober 2024].
- [15] jweiland, „TYPO3 für Redakteure,“ 17 Juli 2024. [Online]. Available: <https://jweiland.net/video-anleitungen/typo3/typo3-fuer-redakteure.html>. [Zugriff am 25 Oktober 2024].
- [16] M. Morell, „Die 7 wichtigsten Infos über Voice User Interfaces,“ 26 September 2022. [Online]. Available: <https://blog.mvst.co/de/post/infos-und-design-von-voice-user-interface/>. [Zugriff am 20 Oktober 2024].
- [17] Amazon, „Alexa Development Tools and Skill Management APIs,“ [Online]. Available: <https://developer.amazon.com/en-US/alexa/alexa-skills-kit/get-deeper/dev-tools-skill-management-api>. [Zugriff am 22 Oktober 2024].
- [18] S. Chauhan, „Was ist TYPO3? Hier ist Was Sie wissen müssen,“ 5 Juni 2023. [Online]. Available: <https://t3planet.de/blog/was-ist-typo3/>. [Zugriff am 22 Oktober 2024].
- [19] L. Brocks und A. Bätjer-Gleitsmann, „The Age of Voice 3.0 Zwischen Routine und Potenzialen für Skills, User Experiences und Voice SEO,“ OMD Germany GmbH, Susanne Grundmann, Düsseldorf, 2021.
- [20] S. Taş, A. Liebe und L. Wiewiorra, „WIK-Kurzstudie: Digitale (Sprach-)Assistenten,“ Bad Honnef, 2022.
- [21] OpenAI, „ChatGPT can now see, hear, and speak,“ 25 September 2023. [Online]. Available: <https://openai.com/index/chatgpt-can-now-see-hear-and-speak/>. [Zugriff am 24 Oktober 2024].

- [22] A. Mosby, „62 Voice Search Statistics For 2024 (Usage & Trends),“ 16 November 2024. [Online]. Available: <https://www.yaguara.co/voice-search-statistics/>. [Zugriff am 20 November 2024].
- [23] D. Kavakli, „Der CMS-Vergleich Teil 1: TYPO3 vs. Wordpress,“ [Online]. Available: <https://www.wus.de/blog/detailseite/der-cms-vergleich-teil-1-TYPO3-vs-wordpress>. [Zugriff am 25 Oktober 2024].
- [24] MUUUH! Next GmbH, „MUUUH! Voice Studie 2023,“ 2023.
- [25] bitkom, „Mehr als die Hälfte nutzt digitale Sprachsteuerung,“ 13 September 2023. [Online]. Available: <https://www.bitkom.org/Presse/Presseinformation/Mehr-Haelfte-nutzt-digitale-Sprachsteuerung>. [Zugriff am 13 Oktober 2024].
- [26] StatCounter, „Desktop Browser Market Share Germany,“ Oktober 2024. [Online]. Available: <https://gs.statcounter.com/browser-market-share/desktop/germany#monthly-202409-202409-bar>. [Zugriff am 1 November 2024].
- [27] Google, „Google Assistant Action Console,“ [Online]. Available: <https://developers.google.com/assistant/console>. [Zugriff am 3 Oktober 2024].
- [28] E. D. Hardt, „The OAuth 2.0 Authorization Framework,“ Oktober 2012. [Online]. Available: <https://datatracker.ietf.org/doc/html/rfc6749>. [Zugriff am 15 Oktober 2024].
- [29] TYPO3, „TYPO3 Extensions,“ [Online]. Available: <https://docs.typo3.org/m/typo3/tutorial-getting-started/main/en-us/Concepts/Extensions/Index.html>. [Zugriff am 14 Oktober 2024].
- [30] TYPO3, „TYPO3 Introduction to Routing,“ [Online]. Available: <https://docs.typo3.org/m/typo3/reference-coreapi/main/en-us/ApiOverview/Routing/Introduction.html>. [Zugriff am 10 Oktober 2024].
- [31] PHP Framework Interop Group, „PSR-7: HTTP message interfaces,“ [Online]. Available: <https://www.php-fig.org/psr/psr-7/>. [Zugriff am 18 Oktober 2024].
- [32] TYPO3, „TYPO3 API A-Z,“ [Online]. Available: <https://docs.typo3.org/m/typo3/reference-coreapi/main/en-us/ApiOverview/Index.html>. [Zugriff am 9 Oktober 2024].
- [33] TYPO3, „TYPO3 ModifyPageLayoutContentEvent,“ [Online]. Available: <https://docs.typo3.org/m/typo3/reference-coreapi/main/en->

us/ApiOverview/Events/Events/Backend/ModifyPageLayoutContentEvent.html. [Zugriff am 14 Oktober 2024].

[34] TYPO3, „TYPO3 ModifyButtonBarEvent,“ [Online]. Available: <https://docs.typo3.org/m/typo3/reference-coreapi/main/en-us/ApiOverview/Events/Events/Backend/ModifyButtonBarEvent.html>. [Zugriff am 11 Oktober 2024].

[35] TYPO3, „TYPO3 ButtonBar,“ [Online]. Available: <https://api.typo3.org/main/classes/TYPO3-CMS-Backend-Template-Components-ButtonBar.html>. [Zugriff am 11 Oktober 2024].

[36] ISO/IEC/IEEE, „ISO/IEC/IEEE 29119-4:2015 - Software and systems engineering — Software testing — Part 4: Test techniques,“ Schweiz, 2015.

[37] TYPO3-Community, „Zugriff auf Modal-Elemente im TYPO3-Backend per JavaScript,“ November 2024. [Online]. Available: <https://forum.t3academy.de/d/702-zugriff-auf-modal-elemente-im-typo3-backend-per-javascript/14>. [Zugriff am 25 November 2024].

## 10 Abbildungsverzeichnis

Abbildung 1: Gründe für Nichtnutzung WIK Studie .....	13
Abbildung 2: Codeausschnitt Grundaufbau JavaScript .....	21
Abbildung 3: Codeausschnitt Sprachkanal .....	22
Abbildung 4: Codeausschnitt Sprachsteuerung.....	22
Abbildung 5: Codeausschnitt Transkript Spracheingabe .....	23
Abbildung 6: TYPO3 Menüband .....	23
Abbildung 7: Codeausschnitt Texteingabe .....	24
Abbildung 8: Codeausschnitt clickTab().....	24
Abbildung 9: Codeausschnitt Button für die Aktivierung .....	26
Abbildung 10: Codeausschnitt Button einfügen .....	26

Abbildung 11: Alexa Developer Console .....	32
Abbildung 12: Codeausschnitt Alexa.....	33
Abbildung 13: Fehlermeldung Falsche Spracherkennung .....	38
Abbildung 14: Falsche Spracherkennung .....	39